tanulmányok 110/1980

MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest

MAGYAR TUDOMÁNYOS AKADÉMIA
SZÁMITÁSTECHNIKAI ÉS AUTOMAITZÁLÁSI KUTATÓ INTÉZETE

TANULMÁNYOK A
A STATISZTIKAI ADATFELDOLGOZÁSRÓL

Irta:

Gaál Anna
Soltész János
Ruda Mihály
Ratkó István

Tanulmányok 110/1980.

A kiadásért felelős:

DR VÁMOS TIBOR

ISBN 963 311 107 2
ISSN 0324-2951

8011534 MTA KESZ Sokszorosító, Budapest. F. v.: dr. Héczey Lászlóné

T a r t a l o m

Jelen tanulmány néhány olyan számitástechnikai /adatfeldolgozási/ és matematikai eredményt mutat be, amelyek a SZTAKI Valószinüségszámitási Osztályán létrehozott SIS77 információs rendszerrel kapcsolatosak, és amelyek általános érdeklődésre tarthatnak számot.

A tanulmány első két dolgozata a VIII. Magyar Operációkutatási Konferencián elhangzott előadás; elméleti kérdésekkel foglalkozik. A következő három programleirás olyan számitástechnikai eszközöket mutat be, amelyek bármely adatfeldolgozási rendszerben alkalmazhatók.

Optimalizálási kérdések a statisztikai adatfeldolgozásban

Ruda Mihály

Az MTA SZTAKI Valószinüségszámitási és Matematikai Statisztikai Osztályán egy csoport már több éve foglalkozik kórházi morbiditási adatok statisztikai feldolgozásának kérdéseivel. A feladat nagy mérete /országos adatokról van szó/ és bonyolultsága miatt szükség volt arra, hogy a kialakitott számitógépes adatfeldolgozó rendszer egyes részeiben egy optimális megoldásmód felé közelitsen. Az előadás ezeket az optimalizálási lehetőségeket tárgyalja. Elsősorban nagy adattömeg bonyolult rendszerben történő statisztikai feldolgozását vizsgáljuk. Bár az ilyen /nagyméretü, összetett statisztikai/ feladatok viszonylag ritkák a számitástechnikai gyakorlatban, mégis érdemes optimalizálásukkal foglalkozni, hiszen esetenként jelentős anyagi kapacitást kötnek le, ahol bizonyos hányadú megtakaritás nagy haszonnal járhat. Ugyanakkor ezek a feladatok központi szerepet játszanak, - hiszen éppen ez indokolja a jelentős költségeket - és igy fontos lehet pl. az átfutási idő csökkentése is.

A statisztikai adatfeldolgozás általános kérdéseivel, és speciálisan a kórházi morbiditási adatok vizsgálatával a szerző és munkatársai már több publikációban foglalkoztak. Ezek a publikációk számos optimalizálási kérdést érintenek, de elszórtan, különböző helyeken és különböző formában jelentek meg /hazai és nemzetközi konferenciák, folyóiratcikkek, tanulmányok, szerződésdokumentációk - ld. [1-6], [10-12], [16], [17], [20-25]/. A jelen előadás célja ezeknek a szétszórt eredményeknek az összefoglalása, egységes keretben történő tárgyalása. Az előbb idézett publikációk a statisztikai adatfeldolgozás kérdéseit nem operációkutatási szempontból közelitik meg, hanem általános

adatfeldolgozási, orvosszakmai, egészségügyi szervezési, rendszerszervezési, matematikai statisztikai problémákat tárgyalnak. Ezért az előadás az eredmények összefoglalásán tul, azoknak elsősorban operációkutatási vonatkozásait emeli ki.

A felvetett problémák mindegyike az előbb emlitett adatfeldolgozási rendszeren belül merült fel, tehát gyakorlati feladatokból erednek. Ugyanakkor a fent emlitett rendszer készitésekor és alkalmazásakor lehetőség nyilt a kialakitott elvek és elképzelések helyességének ellenőrzésére, és a létrehozott eljárások hatékonyságának kipróbálására. Az alkalmazási feladatok a kórházi morbiditási rendszerben évenként mintegy 6oo ezer illetve 15o ezer személy adatainak igen részletes és szerteágazó, de a változó igényekhez maximális mértékben igazodó statisztikai elemzését jelentették.

Az előadás nem egy jól körülhatárolt matematikai kérdés vizsgálatával foglalkozik, hanem különböző kérdéscsoportok felsorolásával. A felsorolásban nem kivánunk sem logikai sem didaktikai szempontokat érvényesiteni. Elsősorban arra kivánjuk felhivni a figyelmet, hogy egy olyan viszonylag szük területen is mint a statisztikai adatfeldolgozás, mennyi fontos és érdekes optimalizálási probléma merül fel.

A számitástechnikai alkalmazások során sokszor pontosan megfogalmazott matematikai modellel dolgoznak a kutatók. Fontos feladat azonban időnként a használt modellek körét kiterjeszteni, és a gyakorlatban felmerült problémák alapján újabb és újabb matematikai feladatokat /modelleket/ megfogalmazni. A statisztikai adatfeldolgozás számitástechnikai problémáin belül számos érdekes matematikai feladat kerül előtérbe /ezek elsősorban matematikai statisztikai és operációkutatási problémák/. Ugyanakkor a statisztikai vizsgálatok esetén sokszor hiányzik a szakmai /pl. orvosi, szociológiai, stb./ kérdések pontos megfogalmazása. Tehát szükség van a matematikai problémák megoldásán túl azok megfogalmazására is. Ez utóbbi jelenség természetesnek mondható,

hiszen a statisztikai adatszolgáltatás éppen a vizsgált populáció megismerését szolgálja; ismeretlen populációval kapcsolatban viszont nehéz előre jól használható kérdéseket /modelleket/ megfogalmazni. Ebből a problémából indulunk ki első kérdéskörünk tárgyalásakor.

1. Statisztikai adatfeldolgozások tervezése, a feldolgozások optimalizálásának szervezése

Mivel a statisztikai vizsgálatok előtt az eredményeket felhasználó szakembereknek sokszor nincs pontos elképzelésük a feldolgozásra kerülő populációról, ezért az ilyen esetekben az adatok /statisztikai tulajdonságainak/ megismerésére egy fokozatos megközelités - "szekvenciális" feldolgozásmód - javasolható /ld[6] /.

Ennek a módszernek az alkalmazásával, a matematikai statisztikából ismert szekvenciális mintavételi módszerhez hasonlóan az információk egyre bővülő körét állitjuk elő, a már rendelkezésre álló ismeretek függvényében. Azonban a szekvenciális mintavételi eljárástól eltérően itt nem azonos jellegü információk növekvő mennyiségéről van szó, hanem az információk minősége is változik. A szakemberek /orvosok, közgazdászok, stb./ először egyszerü, jól áttekinthető adatokat /táblázatok, grafikonok, egyszerübb statisztikai jellemzők, pl. átlag, terjedelem, stb./ kapnak. Ezek alapján teszik fel azután az egyre részletesebb kérdéseket, amelyekre esetleg egy fokozatosan bővülő populáció felhasználásával kapnak választ.

Ilyen módon növelhető a feldolgozás hatékonysága. Felesleges információkat nem kell szolgáltatni, az egyszerübb összefüggésekre rögtön fény derül, és a felhasználó egy jobb áttekintést kap a vizsgált populációról /átfogó és részletesebb információi is vannak/. A rendelkezésre álló adatmennyiségből több értékes információ nyerhető.

Ennek a módszernek a hátránya egy előre megtervezett /és rögzitett/ feldolgozással szemben az, hogy az együttmüködő felek kapcsolatait általában szerződések rögzitik, amelyeket nehéz egy előre nem megadott, változó feldolgozáshoz igazitani. Ezt a szempontot a feladatok szervezésének kezdeti szakaszában már figyelembe kell venni.

A legtöbb optimalizálási feladatnál, igy az adatfeldolgozásban is egy bonyolult sokdimenziós függvényt kell vizsgálni [25]. Információfeldolgozásnál pl. nemcsak a hasznos információk mennyiségét kivánjuk növelni, miközben egy viszonylag kisméretü mintát akarunk feldolgozni, hanem a feldolgozó rendszer előállitásához és müködtetéséhez szükséges erőforrásokkal is takarékoskodni akarunk. Ugyanakkor a feldolgozás átfutási idejét is minimalizálni kivánjuk. Az egyes feladatokat jelentőségük szerint súlyozhatjuk is, igy a ráforditások és az eredmények aránya sem egyértelmü. Külön probléma, hogy a szóbanforgó föggvény dimenziói nem mindig összemérhetők /pl. gépóradij, statisztikai táblázatokba foglalt információ, átfutási idő, emberi munkaerő, stb./.

Itt emlitünk meg egy a feladatok szervezével kapcsolatos általános optimalizálási problémát. Feladatainkat - durván szólva - kétféleképpen közelithetjük meg. Az egyik lehetőség az, hogy nem törődünk a megoldásmód javitásával és a legegyszerübb kivitelezést választjuk, még ha az sok kapacitást köt is le /mérnöki munkáknál jó példa erre a túlbiztositott szerkezetek esete; a tervezési, számitási munkát minimalizáljuk, de a szerkezet előállitásához szükséges munka és anyagmennyiség tetemesen növekszik/. A másik lehetőség az, hogy nagy energiát forditunk a tervezésre /jelen esetben az adatfeldolgozás optimalizálására/, és igy jelentős erőforrások szabadulnak fel a feldolgozás folyamán. Bonyolultabb feladatoknál a tervezési munkák túlsúlyba kerülhetnek és igy éppen ezek köthetnek már le feleslegesen nagy kapacitásokat. Nagy gondot kell tehát forditani az egészséges arányok

kialakitására /ld. pl. [14], [15]/.

2. Azonositó kódok tervezése

Az adatfeldolgozás egyik legfontosabb lépése - mely már az előkészitő munkák során előtérbe kerül - egy jól használható azonositó kód kialakitása. Minden adatfeldolgozási feladatnál, ahol egyedek megkülönböztetésére, megjelölésére szükség van, elő kell állitani egy azonositó kódot. A legegyszerübb esetekben ez a kód egy sorszám. Ha azonban az adatfelvétel nem egy helyen történik, vagy a vizsgálatban szereplő egyedek többször is sorrakerülhetnek, akkor egy olyan azonositóra van szükség, amely éppen a megjelölni kivánt egyed adataiból áll, és igy természetes módon biztositja az azonositást - pl. embereket születési adataik és más állandó jellemzőik /nemük, családi és keresztnév, stb./ jelölhetnek meg. Ezeknél az adatoknál viszont véletlenszerü egybeesések lehetségesek. A megoldandó feladat egy olyan azonositó kód kialakitása, amely viszonylag egyszerü, de a véletlenszerü egybeeséseket minimalizálja. Ennek a problémának a megoldása egy tipikus sztochasztikus optimalizálási feladat. A kérdéskört bonyolitja még az is, hogy az azonositóban szereplő kódértékek /pl. életkor/ nem egyenletes eloszlásúak, együttes eloszlásukról pedig többnyire csak feltételezéseink vannak. A kérdés részletesebb tárgyalása - a kórházi morbiditási vizsgálat konkrét példájával illusztrálva - az [5], [12] és [13] tanulmányokban szerpel.

3. A reprezentativ mintavétel problémái

Az előző kérdéskörhöz hasonló módon problémákat okozhat reprezentativ minták biztositása. A fő probléma az, hogy különböző szempontok alapján kivánunk statisztikákat készitени. A kórházi morbiditási vizsgálatból a következő példát

vehetjük: A felhasznált mintát a kórházi betegek születésnapja alapján választjuk ki /a 4-én, 14-én és 24-én és esetleg még egy negyedik napon született betegek alkotják a vizsgált 10 %-os mintát/. Nyilvánvaló, hogy az igy nyert populációban az életkor, a nem, a különböző betegségek stb. szerinti eloszlások az értékek természetes szóródása miatt nem lesznek arányosak /pontos tizedrész/ a teljes populációbeli eloszlással. Optimalizálási feladatként kitüzhető egy vagy több adott eloszlás, adott pontosságú becsléséhez szükséges minimális minta kiválasztása. Néhány ide vonatkozó klasszikus matematikai statisztikai kérdést vet fel a [12] tanulmány. Emeljünk ki ezek közül egy érdekes és újszerü részfeladatot, a kórházi morbiditási vizsgálat területéről.

A kórházi adatok feldolgozása évenként történik. Egy beteg egy éven belül többször is kórházba kerülhet. A mintakiválasztás kórházi szakmánként bontva történik /egy szakma - pl. belgyógyászat - betegei közül pontosan 10 %-nyi kerül a mintába/. A különböző szakmákban az egyes születésnapokra /melyek az előbb emlitett módon meghatározzák a mintakiválasztást/ jutó betegek számának szóródása miatt gyakran előfordulhat, hogy pl. egy 4-én született beteg bekerül a mintába, amikor mondjuk egy sebészeti osztályon ápolták, de mondjuk egy másik alkalommal, amikor pl. belgyógyászaton feküdt, kimaradhat a mintából, ha pl. a belgyógyászati szakma 10 %-os mintájához nem szükséges minden 4-én született beteg ápolási esete. Ez a jelenség az egy évben többször ápoltak statisztikáit nagymértékben torzithatja. Az itt vázolt probléma egzakt vizsgálata - még sok egyszerüsitő feltételezés mellett is - bonyolult matematikai statisztikai modellhez vezet [12].

4. Adatszervezési kérdések

Az adattárolás kérdéskörét számos különböző oldalról közelithetjük meg. Egyszerü szekvenciális file-ok esetén is

vizsgálhatjuk a felhasznált tárolókapacitás nagyságát a tárolt adatok eléréséhez szükséges időt. Tekintsünk egy egyszerü példát. Nemnemgativ egész számokat kivánunk számológépen tárolni. A szokásos megoldás a decimális vagy a bináris tárolás. Az első esetben /pl. BCD kódban/ számjegyenként 6 bitet foglalunk el, a második esetben szavanként helyezhetünk el egy számértéket - egy szó pl. egy HWB 66-os gépen 36 bit. Az adatfeldolgozásban gyakori, hogy néhány /egy-két/ jegyből álló számokat kell tárolni. Ez decimális formában gazdaságosabb /csak 12 bit kell hozzá/. Decimális értékekkel viszont a számitási eljárások általában lassubbak, mint bináris értékekkel.

Az általunk kidolgozott statisztikai rendszerben [25] a két tárolásmód előnyeit egyesitő megoldást alkalmaztunk; egy-egy adat számára annyi bitet foglalunk le, ahány a szóbanforgó adat maximális értékének bináris tárolására elegendő.

Itt és a következőkben újra és újra felmerül az 1.pontban már felvetett kérdés: javitja-e a számológépfelhasználás hatásfokát az itt vázolt - a szokásosnál valamivel bonyolultabb-adattárolási forma. Megtérül-e a rendszer alkalmazásakor a kialakitására forditott többletmunka?

Egy ugyancsak jelentős számitástechnikai kérdés a kódolási, rendezési eljárások témaköre. Ennek a témakörnek gazdag irodalma van /ld. pl. [9] /. Most csak két részletkérdést emelünk ki.

Adatfeldolgozás egyik első lépéseként ellenőrzési, átkódolási eljárásokat szokás végezni. Ezek sokszor bonyolult logikai kapcsolatok vizsgálatát igénylik. Ilyen célt szolgál pl. az u.n. döntési táblázatok alkalmazása [7] vagy a sok változatban kidolgozott keresési eljárások /pl. bináris keresés/ is. Az utóbbi eljárást logikai szabályok vizsgálatánál úgy alkalmazhatjuk, hogy a szóbanforgó objektumot /pl. egy vagy több adatértéket/ megkeressük a vizsgált szabályt leiró függvény értelmezési tartományában.

A bevezetésben emlitett /kórházi morbiditási/ adatfeldolgozási rendszerben egy hierarchikus gráfstrukturával leirható ellenőrzési, kódolási eljárást alakitottunk ki [24], amely mind a tárolóigény mind a feldolgozási idő szempontjából előnyösnek mutatkozott.

Egy másik - adatok rendezésével kapcsolatos - kérdés ugyancsak a kórházi morbiditási vizsgálatok kapcsán merült fel [5]. Hogyan lehet mágnesszalagokon lévő szekvenciális file rendezésekor az input és output szalagok mozgatását minimalizálni, és mekkora ez a minimum, ha a rendelkezésre álló munkaterület /mágneslemezen/ adott nagyságú? Az [5] tanulmány meghatározza a minimum pontos értékét és a rendezési feladat optimális szervezésének módját is.

5. Logikai kifejezések számitógépes vizsgálata

Speciálisan /konjunktiv vagy diszjuktiv/ normálformára hozott logikai kifejezések kiszámitásának egy gyors és egyszerü módjával foglalkozik [16], [17] és [18]. Az utóbbi két publikáció sztochasztikus optimalizálási kérdéseket is felvet a logikai kifejezések optimális összeállitásával kapcsolatban, ha a kifejezésben szereplő egyes változók statisztikai viselkedését /eloszlásukat, az eloszlások paramétereit/ ismerjük.

6. Szervezési, programozási és alkalmazási feladatok

A következőkben azzal foglalkozunk, hogy milyen stratégiát érdemes követni számológépes rendszerek kialakitásakor és alkalmazásakor - különös tekintettel a statisztikai adatfeldolgozás kérdéseire. Ezen a ponton - mint általában a tervezési feladatoknál - nagy szerep juttatható a hálózati folyamatok [8] elméletének. Tervezéskor - mivel most bonyolult rendszerekről van szó - sok nehezen vagy pontatlanul meghatározható tényezővel kell számolni. Ezért célszerü elsősorban becslésekre szoritkozni, pl. a hálózati folyamat

"alaki" jellemzőit vagy sztochasztikus viselkedését vizsgálni [19] .

A kórházi morbiditási adatokat feldolgozó rendszer kialakitásának és alkalmazásának tapasztalatai alapján a következőket mondhatjuk el.

Nagyobb rendszereknél mind a felhasználás, mind a rendszerkészités szempontjából feltétlenül egy modulszerkezet kialakitása javasolható. Itt a következő szempontokat kell figyelembevenni: az egyes modulok egymástól függetlenül felépithetők /a párhuzamosan folyó tevékenységek rövidebb átfutási időt biztosithatnak/; növekszik a rendszer felhasználhatósága, mivel az egyes modulok egymástól függetlenül akár különböző helyeken, különböző célokra is felhasználhatók, egy feladaton belül az egyes modulok sokféleképpen kombinálhatók, egyes modulok alkalmazására egy folyamaton belül többször vissza lehet térni. A kórházi morbiditási adatfeldolgozó rendszerünkben /SIS77 - ld [25]/ egy szétágazó feldolgozási folyamatot alakitottunk ki. Igy lehetővé vált, hogy a rendszer müködésekor jelentős gépakapacitást takaritsunk meg. Megfelelő előkészités után - a vizsgált statisztikai minta nagyságától függetlenül /akár több százezres vagy milliós mintára/ - néhány másodperc gépidő szükséges a statisztikai táblázatok előállitásához /ez az időadat egy HWB 66/60-as gép processor idejére vonatkozik, ha legfeljebb pár ezer soros táblázatot készitünk/.

Itt hivjuk fel a figyelmet a következő lehetőségre is. Statisztikai adatfeldolgozásnál a vizsgált adatok értékkészlete nagymértékben befolyásolja a feldolgozást. Nagy intervallumban változó értékek kezelése nehezebb. A gyakorlati esetek nagy részében azonban az adatok értékhatárai között nem egyenletes eloszlásban szerepelnek a lehetséges adatértékek. Jó példa erre a kórházi morbiditási vizsgálatban szereplő betegségkódok esete, amelyeknél a kódértékek döntő többsége a lehetséges értékek kis hányadán /mintegy 10 %-án/

belül található. Ezt a feldolgozás optimalizálásánál ki lehet használni - ld [3], [12].

7. Egy programozástechnikai lehetőség

Adatfeldolgozó rendszerünkben /SIS77/ a legtöbb részrendszernél alkalmaztuk a következő programozástechnikai fogást [4],[22],[23] . A rendszer programjait nem készitjük el egy előre rögzitett formában, hanem olyan programszerkesztő /generáló/ programokat hozunk létre, amelyek mindig az aktuális helyzetnek legjobban megfelelő feldolgozó programot hozzák létre. Ilyen módon a felhasznált gépkapacitás jelentős mértékben csökkenthető. Ugyanakkor a rendszer rugalmassága nagymértékben növekszik.

8. Adatbázisok és stat. adatfeldolgozás, a felhasznált software eszközök

Befejezésként röviden foglalkozunk a stat. adatfeldolgozás software környezetének kérdéseivel. Statisztikai adatokat általában egyenrangú egyedek halmazából kell képezni. Ennek megfelelően a számitógépes megvalósitásban is lehetőség van a legegyszerübb adattárolási módok alkalmazására /fix rekordok szekvenciális file-jait lehet használni/. Ennek ellenére általános az a jelenség, hogy a statisztikai értékeléseket nem választják külön a) bonyolultabb szerkezetü adatbázisoktól, és az adatbázisok szerkesztési, keresési, stb. feladataira kidolgozott eljárásokat használják az egyszerübb szerkezetü statisztikai sokaságok kezelésére is. Ha egy adatfeldolgozási feladatban szükség is van adatszerkesztési müveletekre, akkor is célszerü ezeket különválasztva kezelni az egyidejüleg nagytömegü adattal dolgozó statisztikai feldolgozásoktól, ld. [21],[25].

Hivatkozások

[1] Csukás A.-né, Greff L., Krámli A., Ruda M., A kórházi morbiditási vizsgálat számitógépes feldolgozásának tapasztalatai és továbbfejlesztése, Számitástechnikai és kibernetikai módszerek alkalmazása az orvostudományban és a biológiában, 5. Kollokvium, Szeged, 1974.

[2] Csukás A.-né, Greff L., Krámli A., Ruda M., An approach to the hospital morbidity data system development in Hungary, Symposium on Medical Data Processing, Toulouse, 1975.

[3] Csukás A.-né, Greff L., Krámli A., Ruda M., Lekérdező rendszer kórházi morbiditási vizsgálat anyagára, Számitástechnikai és kibernetikai módszerek alkalmazása az orvostudományban és a biológiában, 6. Kollokvium, Szeged, 1975.

[4] Gál A., Ruda M., Egy lehetőség Honeywell FORTRAN programok konverziós müveleteinek gyorsitására, SZÁMKI Tanulmányok, 1978/2.

[5] Garádi J., Krámli A., Ratkó I., Ruda M., Statisztikai és számitástechnikai módszerek alkalmazása kórházi morbiditási vizsgálatokban, MTA SZTAKI Tanulmányok, 35/1975.

[6] Greff L., Krámli A., Ruda M., Kórházi morbiditási vizsgálattal kapcsolatos statisztikai és számitástechnikai meggondolások, Számitástechnikai és kibernetikai módszerek alkalmazása az orvostudományban és a biológiában, 4. Kollokvium, Szeged, 1973.

[7] Halassy B., Döntési táblázatok számitógépes feldolgozása, A számitástechnika legújabb eredményei, 4., Statisztikai Kiadó Vállalat, Budapest, 1977.

[8] Klafszky E., Hálózati folyamatok, Bólyai János Matematikai Társulat kiadványa, Budapest, 1969.

[9] Knuth D.E., The art of computer programing, Sorting and searching /3. kötet/, Addison Wesley, London, California, 1973.

[10] Krámli A., Ruda M., The computer realisation and first experiences of the hospital morbidity study, WHO Statisztikai Vándorszeminárium, Budapest, 1974.

[11] Krámli A., Ruda M., Izpravocsno-informcionnaja szisztyema zaproszov bolnyicsnovo morbigyityizma, Sztruktura i organyizacija paketov programm, Nemzetközi Konferencia, Tbiliszi, 1976.

[12] Krámli A., Ratkó I., Ruda M., Soltész J., A statisztikai adatfeldolgozás matematikai és számitástechnikai problémái, MTA SZTAKI Tanulmányok, 70/1977.

[13] MTA SZTAKI dokumentáció, Az 1972-73. évi kórházi morbiditási vizsgálat számológépes feldolgozása /1. és 2. kötet/, Budapest, 1974.

[14] Pogány Cs., Néhány időszerü kérdés számológépekkel kapcsolatban, I., MTA III. O.K., 19., 1969.

[15] Pogány Cs., Az operációkutatás /VII./, Számitástechnika, VIII. évf. 10.sz., 1977.

[16] Ratkó I., Egy számitástechnikai eszköz bonyolult logikai kifejezések leirására orvosstatisztikai alkalmazásokban, Számitástechnikai és kibernetikai módszerek alkalmazása az orvostudományban és a biológiában, 8. Kollokvium, Szeged, 1977.

[17] Ratkó I., Bonyolult logikai kifejezések kiértékelésének számitástechnikai és optimalizálási problémái, MTA SZTAKI Közlemények, 20/1978.

[18] Ratkó I., On optimization problems of logical expressions in programming languages, Matematikai logika a programozáselméletben kollokvium, Salgótarján, 1978.

[19] Ruda M., Some estimates in connection with the critical path method, Project planning by network analysis, Proceedings of the second international congress, Amsterdam, 1969, North-Holland Publ.

[20] Ruda M., Egy általános információs rendszer kórházi morbiditási adatok feldolgozására, Számitástechnikai és kibernetikai módszerek alkalmazása az orvostudományban és a biológiában, 8. Kollokvium, Szeged, 1977.

[21] Ruda M., Statistical Information System with Health Service Application, 4. Winterschool of Visegrád on the Theory of Operating System, Szentendre, 1978.

[22] Ruda M., Egy széles körben alkalmazható program-optimalizálási módszer, MTA SZTAKI Közlemények, 20/1978.

[23] Ruda M., Módszer a programkészités egyszerüsitésére, Számitástechnika, IX. évf., 7.-8. sz., 1978.

[24] Ruda M., Egy számitástechnikai módszer függvénytáblázatok tömör tárolására, egy adatfeldolgozási alkalmazással /kézirat/.

[25] Ruda M., A SIS 77 statisztikai információs rendszer kialakitásának szempontjai, alkalmazásának és továbbfejlesztésének lehetőségei, MTA SZTAKI Tanulmányok /megjelenőben/.

Döntések sztochasztikus optimalizációja adatfeldolgozásnál

Ratkó István

1. A probléma felvetése

Tekintsünk egy olyan L diszjunktiv normálformát, melyben a konjunkciók tagjainak értéke a véletlentől függ.

L kiértékelését a következőképpen hajtsuk végre: a diszjunkciókat és azokon belül a konjunkciókat balról jobbra haladva értékeljük; egy adott diszjunkción belül a konjunkció tagjai közül, illetve a diszjunkciók közül csak annyit értékelünk ki, amennyi szükséges a diszjunkció illetve L értékének megállapitásához.

Definició: L kiértékelési számának nevezzük és $\varrho(L)$-lel jelöljük az előbbi módon meghatározott kiértékelésben megvizsgált konjunkció tagok számát.

Könnyen látható, hogy ha az első diszjunkció nem igaz, a $(k+1)$-edik diszjunkció igaz, akkor

$$\varrho(L) = h_1 + h_2 + \ldots + h_k + a_{k+1}$$

ahol a/ a_i az i-edik diszjunkció konjunkció tagjainak száma b/ a j-edik diszjunkcióban a h_j-edik konjunkció tag hamis, az előtte lévő konjunkció tagok igazak.

Ha az első diszjunkció igaz, akkor $\varrho(L) = a_1$.

/Megjegyzés: Konjunkció tagok ill. diszjunkció tagok helyett a rövidség kedvéért gyakran csak konjunkciót ill. diszjunkciót mondunk./

Nyilvánvaló, hogy $\varrho(L)$ valószinüségi változó. Ha L-ben változtatjuk a diszjunkciók vagy valamelyik diszjunkcióban a konjunkciók sorrendjét, $\varrho(L)$ értéke változik /adott értékü konjunkciók esetén is/.

A következő kérdést vizsgáljuk: a diszjunkciók ill. azokon belül a konjunkciók mely sorrendjénél lesz minimális $\varrho(L)$ várható értéke: $E\varrho(L)$.

2. Eredmények

Legyenek először

$L = \ldots \vee (\ldots \wedge L_1 \wedge L_2 \wedge \ldots) \vee \ldots$ és $\tilde{L} = \ldots \vee (\ldots \wedge L_2 \wedge L_1 \wedge \ldots) \vee \ldots$

a vizsgált kifejezések. $E\varrho(L)$ és $E\varrho(\tilde{L})$ összehasonlitására törekszünk.

Vezessük be a következő jelöléseket:

$L' = \ldots \wedge L_1 \wedge L_2 \wedge \ldots$

$L'' = \ldots \wedge L_2 \wedge L_1 \wedge \ldots$

$p = P(L_1 = i)$ (azaz, hogy L_1 igaz)

$q = P(L_2 = i)$

$A' = \{L'$ -ben minden L_1 előtti konjunkció igaz és minden L' előtti diszjunkció hamis, továbbá $L_1 = i, L_2 = h\}$

$A'' = \{L''$ -ben minden L_2 előtti konjunkció igaz és minden L'' előtti diszjunkció hamis, továbbá $L_2 = i, L_1 = h\}$

$p_1 = P(A') \qquad p_2 = P(A'')$

<u>1. tétel:</u> $E\rho(L) \leqq E\rho(\tilde{L})$ akkor és csak akkor, ha

$$p(1-q)p_1 \leqq (1-p)qp_2 \qquad (1)$$

<u>bizonyitás:</u> Legyen ξ_1 és η_1 (ξ_2 és η_2) L ($\tilde{L}$) kiértékelési száma L' előtt és után (L'' előtt és után) azon feltevés mellett, hogy $L_1 = i, L_2 = h$ és L' -ben minden L_1 előtti konjunkció igaz ($L_1 = h$, $L_2 = i$ és L'' -ben minden L_2 előtti konjunkció igaz). Jelölje k az L_1 előtti konjunkciók számát /L'- ben, L'' -ben/, tovább p' annak valószinüségét, hogy L' -ben minden L_1 előtti konjunkció igaz. Nyilván p' annak valószinüségét is megadja, hogy L'' -ben minden L_2 előtti konjunkció igaz.

Elemi meggondolásokból következik, hogy $E\rho(L) \leqq E\rho(\tilde{L})$ akkor és csak akkor, ha (χ a karakterisztikus változót jelöli)

$$E(\xi_1' + (k+2)\chi_{A'} + \eta_1\chi_{A'})p'p(1-q) + E(\xi_2' + (k+1)\chi_{A''} + \eta_2\chi_{A''})p'(1-p)q \leqq$$

$$\leqq E(\xi_2' + (k+2)\chi_{A''} + \eta_2\chi_{A''})p'(1-p)q$$

Ez pedig ekvivalens (1)-gyel, ami állitásunkat bizonyitja. Megjegyezzük, hogy ha sem L_1 sem L_2 nem szerepel az L' előtti diszjunkciókban, akkor (1) a

$p \leqq q$ egyenlőtlenséggel ekvivalens. ▨

Azt vizsgáljuk ezek után, hogy az $L = \ldots \vee L^{*} \vee L^{**} \vee \ldots$ és az $\tilde{L} = \ldots \vee L^{**} \vee L^{*} \vee \ldots$ kifejezésekre mit mondhatunk $E\varrho(L)$-ről és $E\varrho(\tilde{L})$-ról.

Jelölje a ill. b L^{*}-ben ill. L^{**} -ben a konjunkciók számát. A következő jelöléseket használjuk:

p_1 : annak valószinüsége, hogy L^{*} minden tagja igaz, L^{**} és az L^{*} előtti tagok mindegyike hamis

η : L^{**} -ben az első hamis tag sorszáma, feltéve, hogy $L^{*} = i$

p_2 : annak valószinüsége, hogy L^{*}, L^{**} minden tagja igaz, L^{*} előtti tagok mindegyike hamis

p_3 : annak valószinüsége, hogy L^{**} minden tagja igaz, L^{*} és az L^{**} előtti tagok mindegyike hamis

ξ : L^{*} -ben az első hamis tag sorszáma, feltéve, hogy $L^{**} = i$

<u>2. tétel:</u> $E\varrho(L) \leqq E\varrho(\tilde{L})$ akkor és csak akkor, ha

$$a p_2 + p_3 E\xi \leqq p_1 E\eta + b p_2 \qquad (2)$$

<u>bizonyitás</u> Legyen

ξ_1 : L esetén L^{*} -ig /$\tilde{L}$ esetén L^{**} -ig/ a kiértékelt tagok száma azon feltétel mellett, hogy $L^{*} = i$

: L ($\tilde{L}$) esetén L^{*} -ig /L^{**} -ig/ a kiértékelt tagok száma azon feltétel mellett, hogy $L^{**} = i$, $L^{*} = i$

: L /$\tilde{L}$/ esetén L^{*} -ig /L^{**} -ig/ a kiértékelt tagok száma azon feltétel mellett, hogy $L^{**} = i$

Nyilván $E\varrho(L) \leqq E\varrho(\tilde{L})$ akkor és csak akkor igaz, ha

$$E(\xi_1+a)p_1 + E(\xi_2+a)p_2 + E(\xi_3+\xi'+b)p_3 \leqq$$

$$\leqq E(\xi_1+\eta+a)p_1 + E(\xi_2+b)p_2 + E(\xi_3+b)p_3$$

Ennek átrendezésével adódik (2). ▨

Valószinü, újabb eredmények is elérhetők a feladattal kapcsolatban, melyek az optimalizálást elegánsabbá teszik.

3. Gyakorlati alkalmazás

A MTA SZTAKI Valószinüségszámitási és Matematikai Statisztikai Osztály munkatársai által létrehozott SIS77 általános statisztikai adatfeldolgozó rendszerben használtuk és használhatók az elmondott eredmények. [1]

Adott egy fix hosszuságú rekordokból álló adatfile. A szakembereket igen gyakran csak speciális feltételeknek elegettevő adatok érdeklik. Ezek a feltételek logikai kifejezésekkel adhatók meg. Számuk azonban többezres nagyságrendü is lehet. Hogyan épitsük be a programba ezeket a feltételeket?

A rekord álljon N adatelemből, az i-edik adatelem értékét jelölje ξ_i, ξ_i lehetséges értékeinek halmazát pedig H_i. / H_i lehet pl. egy intervallum, de lehet bonyolultabb halmaz is./ Valamely konkrét adatfeldolgozásnál az adatfile bizonyos feltételeknek elegettevő rekordjaira van csak szükségünk. Hogyan irjunk fel egy az ezen rekordokat kiválasztó logikai kifejezést? A kifejezés akkor és csak akkor legyen igaz, ha a rekordra szükségünk lesz. Tegyük fel, hogy a logikai kifejezésben az $i_1, i_2, \ldots, i_M$ sorszámú adatelemekre vonatkozó feltételek szerepelnek. A k-adik adatelemmel kapcsolatos feltételek igy néznek ki:

$$(\xi_k \in A) \qquad \text{ahol} \qquad A \subset H_k$$

A $(\xi_k \in A)$ itélet tulajdonképpen $|A|$ számú diszjunkcióból áll itt $|A|$ az A halmaz elemszáma.

Jelölje $A_{i_k,j}$ H_{i_k} azon részhalmazait, amelyek $(\xi_{i_k} \in A_{i_k,j})$ formában szerepelnek a rekordokat kiválasztó logikai kifejezésben $(j=1,2,\ldots,r_k;\ k=1,2,\ldots,M)$ Nyilvánvaló, hogy logikai kifejezésünknek legalább

$$\sum_{k=1}^{M} \sum_{j=1}^{r_k} |A_{i_k,j}| \quad \text{tagja /konjunkció,}$$

diszjunkció vegyesen/ van.

Ezt hagyományos módon beépiteni a programba reménytelen, sőt legtöbbször lehetetlen. A fenti tagszám ugyanis - mint már emlitettük - többezres nagyságrendü is lehet.

Mit tehetünk? Redukáljuk a kifejezésben szereplő változók számát. Ezt a következő egyszerü ötlettel érhetjük el.

Definiáljuk a $Z_{i_k,j}(s)$ függvényt a következőképpen:

$$Z_{i_k,j}(s) = \begin{cases} 0, \text{ ha } s \in A_{i_k,j} \\ 1, \text{ ha } s \notin A_{i_k,j} \end{cases} \qquad (s \in H_k)$$

Ezzel elértük, hogy a $(\xi_k \in A_{i_k,j})$ $|A_{i_k,j}|$ számú diszjunkció helyett az egyetlen $Z_{i_k,j}(\xi_k) = 0$ tag áll, ugyanis nyilvánvaló, hogy $(\xi_k \in A_{i_k,j})$ akkor és csak akkor igaz, ha $Z_{i_k,j}(\xi_k) = 0$ igaz. Megjegyezzük, hogy kifejezésünk most legalább $\sum_{k=1}^{M} r_k$ tagból áll.

Bizonyos mértékü egyszerüsitést már elértünk, ha azonban a logikai kifejezés kiértékelési idejét, s igy a futási időt csökkenteni akarjuk, további meggondolásra van szükségünk.

A rekordokat kiválasztó L logikai kifejezésünket hozzuk diszjuktiv normálalakra:

$L = L_1 \vee L_2 \vee \ldots \vee L_d$, ahol tehát L_i konjunkciókból áll.

Nyilvánvaló, hogy az

```
IF(L.EQ.FALSE) GO TO 2 /a rekord kihagyandó/
```

utasitás végrehajtása több időt vesz igénybe, mint az

```
IF (L1.EQ.TRUE)  GO TO 1
IF (L2.EQ.TRUE)  GO TO 1
.                .
.                .
.                .
IF(Ld.EQ.TRUE )  GO TO 1
GO TO 2
```

1 szükség van a rekordra utasitáscsoporté.

Mig ugyanis az első esetben L értékének eldöntéséhez az összes konjunkciót és diszjunkciót ki kell értékelni, a második esetben a kiértékelés hamarabb befejeződik.

Az `IF(Li.EQ.TRUE) GO TO 1` tovább bontható.

Legyen evégből: $L_i = L_{i1} \wedge L_{i2} \wedge \ldots \wedge L_{iq_i}$.

Az `IF(Li1.EQ.FALSE) GO TO Ci+1`

```
IF(Li2 .EQ.FALSE) GO TO Ci+1
      .                .
      .                .
      .                .
```

1 IF(L_{iq_i} .EQ.FALSE) GO TO C_{i+1}
GO TO 1

C_{i+1} { Mint fent, csak az L_{i+1} -et bontva

utasitáscsoport végrehajtása hamarabb fejeződik be, mint az IF(L_i.EQ.TRUE) GO TO 1 utasitásé.

A futási idő ujabb csökkentését érhetjük el a események relativ gyakoriságának /v.valószinüségének/ ismeretében. Ehhez a következő minimum feladatot kell megoldani. Az $1,2,\dots,d$ számok egy permutációja legyen $m_1, m_2, \dots, m_d$ az $1, 2, \dots, q_i$ számoké pedig $n_1, n_2, \dots, n_{q_i}$ $(i=1,2,\dots,d)$. Ha L -ben az L_i diszjunkciók sorrendje $L_{m_1}, L_{m_2}, \dots, L_{m_d}$ továbbá az L_i szétbontásában L_{in_1}, $L_{in_2}, \dots, L_{in_{q_i}}$, a kiértékelések várható száma meghatározható. A kérdés: milyen sorrend esetén lesz ez a várható érték minimális?

Igy eljutottunk az 1. pontban felvetett problémához. A SIS77 rendszerben a usernek - megadott formátum szerint, melyre itt nem óhajtunk kitérni - a H halmazokat, továbbá a $Z_{i_k,j}(\xi_k^d).EQ.0$ logikai feltételeket kell megadnia. A sorrendet ezen feltételekre vonatkozó eddigi morbiditásvizsgálati tapasztalataink valamint a két tétel alapján adtuk meg. [2.]

4. <u>Megjegyzések</u>

a/ A forditóprogramok Boole kifejezések optimalizálására való törekvésekor a két tételt eredményesen lehetne használni. Ehhez természetesen a minimalizálás gépi kódú programmal történő megoldására lenne szükség.

b/ Ugy véljük, az elmondottak döntési táblázatok használatánál is alkalmazhatók. Ugyanis, mint az pl. [3]-ból kiderül, ilyen optimalizálási meggondolásra csak utalnak, de jól használható utasitást nem adnak.

Irodalom

[1] Ruda Mihály, A SIS77 statisztikai információs rendszer kialakitásának szempontjai, alkalmazásának és továbbfejlesztésének lehetőségei /MTA SZTAKI, Tanulmányok, megjelenőben/

[2] A SIS77 statisztikai információs rendszer programjainak leirása, használati utasitásai /MTA SZTAKI, Témadokumentáció, 1978/

[3] Halassy Béla - Zentai Tamás, Döntési táblázatok /NSZÁMOK, 1973/

Egy általánosan használható eljárás táblázatoknak a központi memóriában történő kitöltésére

Soltész János

1. A program célja: Egy olyan eljárás biztositása, amelynek segitségével kényelmesen tölthetünk ki táblázatokat /kódszótárakat, ugrótáblákat, függvény értéktáblázatokat/ a központi memóriában.

2. A program formája, a felhasznált gép:
FORTRAN nyelvü szubrutin, neve: ZSAK.
Honeywell 66-os szintü gépeken a GCOS operációs rendszerben használható.

3. A feladat részletes leirása, a felhasznált módszerek:

3.1. A feladat: A szubrutin a központi memóriában értéktáblázatokat tölt ki. Alkalmazásának több előnye van:

a/ a felhasználó az értéktáblázatok leirását a lehető legkényelmesebb, tömör és jól áttekinthető formában adhatja meg /ld. 6. pont/.

b/ az értéktáblázatok leirásait a szubrutin szintaktikai vizsgálatnak veti alá és részletes hibaüzeneteket ad /ld. 7. pont/ nagy biztonságot nyújtva a felhasználónak

c/ a szubrutin bizonyos többváltozós függvények leképezésének gyors és kényelmes megvalósitását teszi lehetővé.

Az értéktáblázatokat a szubrutin folyamatosan egyetlen Z tömbben helyezi el. Az egyes értéktáblázatokat a következőkben "zsákoknak" nevezzük. A "zsákoknak" a Z tömbben való elhelyezkedését a ZSP tömb megfelelő elemei határozzák meg /ld. 3.5. pont/.

A szubrutin jelenlegi változatában csak nem negativ egész számokon értelmezett függvények szerepelhetnek. Az értékkészlet is csak egész számokból állhat, de itt már lehetnek negativ értékek is. Az értelmezési tartomány csak összefüggő intervallum lehet.

Az értelmezési tartományba tartozó értékek a "zsákok" egymásutáni elemeihez /az egymásutáni memóriarekeszekhez/ tartozó memóriacimek /relativ címek - ld. a ZSP tömb leirását a 3.5. pontban/. Az értelmezési tartomány egy x eleméhez rendelt értéket a szubrutin az x-nek megfelelő memóriacim alatt helyezi el.

Egy "zsák" egy összefüggő memóriatömb /ezért kell, hogy összefüggő legyen az értelmezési tartomány/.

Az értékkészlet elemei legfeljebb négyjegyü egész számok lehetnek /tehát -999 és 9999 közötti értékek/.

3.2. Példák a felhasználásra: leképezhetjük pl. Magyarország megyéit /a megyekódokat/ a Budapest, Dunántúl, Észak-Magyarország, stb. csoportokra, ugyanigy évenkénti finomságban adott életkorokat mondjuk 5-éves korcsoportokra. Természetesen a szubrutin alkalmazásának hasznossága nagyobb és bonyolultabb kódrendszer esetén mutatkozik meg igazán. Például orvosi alkalmazásoknál kb. 3000 diagnózist kell leképezni /nem monoton függvénnyel/ egy 300 tételt tartalmazó jegyzékre.

3.3 Többváltozós függvényeknél akkor lehetséges a szubrutin használata, ha a leképezés

$$f(x_1,x_2,\ldots,x_n)= h\left(g_1(x_1),g_2(x_2),\ldots,g_n(x_n)\right)$$

alakú. Ha h az f-nél egyszerübb vagy jobban kezelhető függvény, akkor a szubrutin használata előnyös is.

Ilyen esettel állunk szemben akkor is, amikor többdimenziós statisztikai táblázatokat készitünk, vagy egyik táblázatot a másikra képezzük le. Ebben az esetben a $g_j(x_j)$ értékeket a szubrutin a felhasználó kérésére automatikusan kiszámitja.

3.4. Memóriacimszámitás /beszorzás/

Többdimenziós statisztikai táblázatok készitésekor egy x_j $(j=1,2,\ldots,n)$ sorozatot képezünk le egy y_j $(j=1,2,..,n)$ sorozatra $(n\geq 2)$, és az y_j értékekből, mint indexekből kiszámitjuk a létrehozandó táblázat megfelelő $/y_1,y_2,\ldots,y_n/$ indexü elemének memóriacimét /ld. még később/.

Az $/x_1,x_2,\ldots,x_n/$ értékek lehetnek egy beolvasott rekord kijelölt elemei, vagy egy n dimenziós táblázat futó indexei. Mindkét esetben az x_j értékekből kell egy memóriacimet kiszámitani a következő módon:

Jelölje $UKM(j)$ az y_j értékek egy felső korlátját. Kikötjük, hogy y_j csak pozitiv lehet, tehát

$$1 \leq y_j \leq UKM(j) \qquad j=1,2,\ldots,n$$

Az $UKM(j)$ érték lesz a készitendő táblázat j-edik dimenziójának mérete. A számitandó M memóriacimet n dimenziós táblázat esetén a következőképpen állithatjuk elő:

$$M = y_n + \sum_{j=1}^{n-1}\left[(y_j-1) * \prod_{k=j+1}^{n} UKM(k)\right] \qquad (1)$$

Ha a felhasználó kér memóriacimszámitást /beszorzást/ akkor a szubrutin kiszámitja a

$$g_j^*(x_j) = (y_j - 1) \prod_{k=j+1}^{n} UKM(k) \qquad j = 1, 2, \ldots, n-1$$

$$g_n^*(x_n) = y_n \qquad (2)$$

értékeket az x_j-k minden lehetséges értékére. A (2) képlet jobboldalán álló szorzási müveletekről kapta a szubrutin ilyen müködési módja a "beszorzás" nevet.

A (2) formulák kiszámitása két lépésben történik:

A felhasználó a szokásos módon /ld. 6. pont/ megadja az $x_j \rightarrow y_j$ leképezést. /Egy leképezés egy zsák./ Ezután a szubrutin a zsákokban levő y_j értékek helyére a (2)-ben leirt $g_j^*(y_j)$ számokat teszi.

A felhasználó ezután (1) és (2) alapján az

$$M = g_1^*(x_1) + g_2^*(x_2) + \ldots + g_n^*(x_n)$$

formában előállithatja a kivánt memóriacimet. Látható, hogy a táblázat dimenzióinak felsorolása /az (1)-ben és (2)-ben szereplő j index/ a legkülső dimenziótól indul; az n-edik dimenzió a legbelső.

A fenti esetben a 3. pontban szereplő h függvény alakja

$$h\left(g_1^*(x_1), g_2^*(x_2), \ldots, g_n^*(x_n)\right) = g_1^*(x_1) + g_2^*(x_2) + \ldots + g_n^*(x_n)$$

A szubrutin eredeti müködési módjában /ha nem kérünk beszorzást/ csak az $x_j \rightarrow y_j$ leképezés /osztályokba sorolás/ történik meg.

3.5. A szubrutin müködése: A szubrutint hivó főprogram által biztositott területen belül tetszőleges számú "zsák" /leképezés/ hozható létre. A zsákok megkülönböztetésére /megjelölésére/ zsáknév szolgál /ld. a ZSN paramétervektor leirását az 5. pontban/. A zsáknevet minden zsákra meg kell adni. A létrehozott zsákok némelyikére kérhetünk beszorzást /ld. a 3.4. pontot/. Ezeknek a zsákoknak a kijelölése és sorrendjük megadása a GY paramétertömbben történik. Ha vannak beszorzandó zsákok, akkor ezek midegyikére külön-külön meg kell adni értékkészletük felső korlátját /ld. az UKM paramétervektor leirását/.

Azért, hogy az egyes zsákok méreteit és elhelyezkedésüket meghatározhassuk, meg kell adni az értelmezési tartományok alsó és felső korlátját /ld. az RKK és az RKV paramétervektorok leirását az 5. pontban/.

A j-edik "zsák" kezdőcimét meghatározó ZSP(j) vektor olyan, hogy az x_j értékhez tartozó y_j értéket /vagy a megfelelő $g_j^{*}(x_j)$ beszorzott értéket/ a zsákokat tartalmazó Z tömb ZSP(j)+ x_j cimén találjuk, azaz

$$y_j \left(\text{vagy } g_j^{*}(x_j)\right) = Z\left(ZSP(j) + x_j\right)$$

A felhasználó kérheti az elkészült "zsákok" /értéktáblázatok/ tartalmának kinyomtatását /ld. a KIIRZS paraméter leirását/.

3.6. Megjegyzések

a/ A 2. pontban emlitett példák alapján a "zsákok" /értéktáblázatok/ értelmezési tartományainak elemeit /az x_j értékeket/ a továbbiakban régi kódoknak, a "zsákok" értékkészleteinek az elemeit /az y_j értékeket/ pedig új kódoknak fogjuk hivni.

b/ A ZSAK szubrutin egy KULONB nevü szubrutint tartalmaz - ez a felhasználót nem érinti.

4. A szubrutin inputja és outputja

4.1. Az input

a/ Bemenő paraméterek /ld. 5. pont/

b/ A zsákleiró kártyákat tartalmazó 18-as filekódú file. A zsákleiró kártyákat /ld. 6. pont/ beolvashatjuk kártyáról, munkafile-ról, stb.

4.2. Az output

a/ Kimenő paraméterek/ld. 5. pont/

b/ Lista

Ha nem kérünk beszorzást, akkor az "OSZTÁLYOZÁS", ha kérünk, akkor a "TÁBLÁZATOK ÖSSZEVONÁSA" felirat jelenik meg a printerlapon. Ezt a paraméterek kinyomtatása követi. A szubrutin kiirja a zsákok együttes hosszát is, igy a felhasználó ellenőrizheti, hogy a Z tömb méretet elég nagynak adta-e meg a ZSÁK szubrutint behivó programban.

A szubrutin kinyomtatja a beolvasott zsákleiró kártyákat is.

Ha fatális hiba nem történik a zsákleiró kártyák feldolgozása során /a fatális hibákat a 7. pontban soroljuk fel/, és a felhasználó kéri, akkor a szubrutin kinyomtatja az elkészitett zsákok tartalmát. Ez a beszorzás elvégzése után is megtörténik /persze csak akkor, ha kér beszorzást a felhasználó/.

A zsákok tartalma a következő módon jelenik meg a printerlapon: A szubrutin kinyomtatja a ZSN(1) nevü zsák nevét, valamint régi kódjának alsó és felső korlátjait /azaz RKK(1)-et és RKV(1)-et/. Ezek után kiirja a ZSN(1) zsáknak az RKK(1), RKK(1)+1, RKK(1)+2,..., RKV(1) régi kódjaihoz rendelt új kódjait 10(20X, 10I6/) formátum szerint. Igy egy sorban 10 új kód található, és minden tizedik sor után egy sor üresen van hagyva. A ZSN(1) zsák tartalmának kinyomtatása után a szubrutin hasonló módon kiirja a többi zsák tartalmát.

Hiba esetén a szubrutin hibaüzenetet ir ki. Ezek leirása a 7. pontban található.

5. A paraméterek

A szubrutin hivása:

CALL ZSAK (ZSSZ,NBE,ZSN,RKK,RKV,GY,UKM,Z,ZSP,KIIRZS)

A paraméterek egész tipusúak.

Z és ZSP kimenő adat, a többi bemenő adat.

A fenti paraméterek legtöbbjének jelentését a 3.5. pontban már leirtuk. Most kiegészitjük az ott mondottakat, és megadjuk azokat a feltételeket, amelyeket a paramétereknek teljesiteniük kell.

ZSSZ – "zsákszám"; a készitendő zsákok száma; skalár; $1 \leqslant ZSSZ \leqslant 999$

NBE – a beszorzandó zsákok száma; skalár. Ha nem kérünk beszorzást, akkor NBE=0; ha kérünk, akkor $1 \leqslant NBE \leqslant \min(ZSSZ, 20)$

ZSN - "zsáknév"; ZSSZ elemü vektor, mely egymástól különböző, I4 formátummal beolvasható egész számokat tartalmaz

RKK,RKV - "a régi kódok kezdő- és végértékei", pontosabban mondva RKK(i) illetve RKV(i) az i-edik, ZSN(i) nevü zsák /leképezés/ értelmezési tartományának alsó ill. felső korlátja. RKK ill. RKV ZSSZ elemü vektor

$$0 \leqslant RKK(i) \leqslant RKV(i) \leqslant 99\ 999$$

GY - a beszorzandó zsákok nevei.
Ha NBE>0 /azaz kérünk beszorzást/, akkor NBE elemü vektor, melyre $GY(j) \neq GY(k)$ $j \neq k$ esetén;
$\{GY(j): j=1,2,\ldots,NBE\} \subseteq \{ZSN(i): i=1,2,\ldots,ZSSZ\}$.
A GY(j) zsákneveket a készitendő táblázat legkülső dimenziójától kiindulva és onnét befelé haladva kell megadni.

Ha NBE=0 /azaz nem kérünk beszorzást/, akkor GY egy vagy több elemü vektor. Nem kell értéket adni neki.

UKM - az "új kód maximális értéke"
NBE>0 esetén NBE elemü vektor. UKM(j) a GY(j) nevü zsák értékkészletének egy felső korlátja.
$UKM(j) \geqslant 1$
NBE=0 esetén UKM egy vagy több elemü vektor. Nem kell értéket adni neki.

Z - a zsákokat tartalmazó tömb; IZSOH elemü vektor, ahol

$$IZSOH = \sum_{i=1}^{ZSSZ} \left[RKV(i) + 1 - RKK(i) \right]$$

ZSP - "zsákplusz"; ZSSZ elemü vektor

KIIRZS - ha KIIRZS=1, akkor a szubrutin kiirja az elkészitett zsákokat. Ha kérünk beszorzást, akkor a beszorzás előtt is és a beszorzás után is megtörténik a zsákok kinyomtatása.
KIIRZS=0 esetén a szubrutin nem irja ki az elkészitett zsákokat.

Megjegyzés: ZSN(i), RKK(i), RKV(i) illetve GY(j) és UKM(j) tartozik ugyanahhoz a zsákhoz /ha ZSN(i)=GY(j)/.

6. A zsákleiró kártyák

Ha hivjuk a ZSÁK szubrutint, akkor az feldolgozza a paramétereit, majd kinullázza a Z zsákot. Ezután elkezdi a zsákleiró rekordok /kártyák/ beolvasását egy 18-as filekóddal jelzett területről. A beolvasás "I1,I4,75A1" formátummal történik. A zsákleiró kártyák 76 -80. karaktereit a szubrutin nem dolgozza fel. Ide tetszőleges commentet /pl. sorszámot/ irhatunk. A kártyák 6. karakterét mindig üresen kell hagyni.

6.1. Zsákkezdő kártyák

Alakjuk: |2| zsáknév |␣|comment

1. 2-5. 6. 7-80. karakter
I1 I4 75A1

Pl |2|␣␣18|␣| BARANYA MEGYE
|2|␣␣-3|␣|FERTŐZŐ BETEGSÉGEK

6.2. "Zsákleirás vége"- kártya

Alakja: |3| | |comment
1. 2-5. 6-75. 76-80. karakter

A zsákleiró kártyák igy követik egymást:

```
|2␣␣␣7|   DUNANTUL
 a 7 nevü zsák leirása
|2␣␣␣3|   FERTOZO BETEGSEGEK
 a 3 nevü zsák leirása
|2␣␣␣5|
 az 5 nevü zsák leirása
 ...
 ...
|2␣␣␣4|   REUMA
 a 4 nevü zsák leirása
|3|
```

Egy zsákot többször is elkezdhetünk /tehát pl. a "2␣␣␣7" kártya többször is szerepelhet/.

6.3. Egy zsák leirása

A "|2| ZSN(i)" kártya után kezdjük a ZSN(i) nevü zsák /azaz függvényértéktáblázat/ leirását.

A tényleges zsákleiró kártyák általános alakja: az első kártyapozición egy vezérlő paraméter áll, a másodiktól ötödiken egy függvényérték /új kód/, a hatodik üres, a hetediktől a hetvenötödik karakteren pedig az értelmezési tartomány bizonyos elemei /bizonyos régi kódok/ állnak. A 76-80. helyre commentet irhatunk:

|V| függvényérték|ᵥ|$r_1, r_2, r_3, \ldots, r_k$ |comment

1. 2-5. 6. 7-75. 76-80. karakter

I1 I4 75A1 beolvasási formátum

ahol r_i egy szám /pl. r_i=17/ vagy egy intervallum /r_i=m-n, pl. r_i=7-13/.

A jobboldalon lehet üres helyeket hagyni, akár számok beiséjében is.

Példák:

|0|ᵥᵥᵥ1|ᵥ|2-10, 16, 2 3,12 |9/1
|ᵥ|ᵥᵥᵥ1|ᵥ|19,15 |9/2
|1|ᵥᵥᵥ3|ᵥ|28-31,47 |9/3
|4|ᵥᵥᵥ5|ᵥ|1-50,53-60 |9/4

1. 6.
2-5. 7-75. 76-80. karakter

6.3.1. "Közönséges" zsákleiró kártya

Az első helyen blank vagy 0 áll

Alakja:

|ᵥ| új kód|ᵥ|régi kódok| comment vagy
|0| új kód|ᵥ|régi kódok| comment

1. 2-5. 6. 7-75. 76-80. karakter

I1 I4 75A1 beolvasási formátum

Pl. |ᵥ|ᵥᵥ13|ᵥ| 7, 19-25 | ABC

Hatása: A szubrutin a 7-75. karakteren felsorolt régi kód értékekhez a 2-5. helyen álló, I4 formátummal beolvasott új kód értéket rendeli. A példában levő esetben a 7,19,20,21,22,23,24,25 régi kódok új kódja 13 lesz.

6.3.2. "Egyesével"-kártya

Alakja: |1| új kód|ᵥ|régi kódok |comment

Pl. |1| 4|ᵥ|11-15,26,5-7,31,39 |EGYES

Hatása: Ha az új kód értékét u-val jelöljük, akkor a szubrutin a 7-75. karakteren felsorolt régi kód értékekhez az u,u+1,u+2,u+3,... új kód értékeket rendeli. A példában emlitett esetben a hatás:

régi kód érték	11	12	13	14	15	26	5	6	7	31	39
új kód érték	4	5	6	7	8	9	10	11	12	13	14

6.3.3. "Tölts ki az üreset" -kártya

Alakja:

	4	y	$r_1,r_2,\ldots,r_k$
Pl.	4	8	0-79,83-101

comment

Hatása: a példában szereplő esetben az y=8 új kód értéket adja a jobboldalon felsorolt mindazon régi kódoknak, amelyeknek még nem volt 0-tól különböző új kód érték adva.

6.3.4. A 0 új kód veszélyei

Ha egy régi kódnak 0 új kód értéket adunk, akkor azt a program hibajelzés nélkül felülirja, ha tévedésből egy újabb értéket adunk ugyanannak a régi kódnak.

7. Hibaüzenetek

A szubrutin azonnal megáll, ha a 7.1. vagy a 7.3.2-7.3.6. pontban felsorolt hibák valamelyikével találkozik. Ekkor a szubrutin a "*** FATÁLIS HIBA ***" üzeneteket irja ki a printerlap jobboldalára. Ha egy, a 7.2.2-7.2.8. vagy a 7.3.1. pontban szereplő hiba áll fenn, akkor a szubrutin

hibaüzenetet ad ki és a printerlap jobboldalán megjelenik a "*** HIBA ***" felirat. A szubrutin azonban tovább fut, kéri a következő zsákleiró kártyát, és csak akkor áll le, ha már elfogytak a zsákleiró kártyák. Leállás előtt a szubrutin még kinyomtatja a zsákok tartalmát /ha KIIRZS értéke 1 volt/ és a következő hibaüzenetet irja ki ötvenötször: A SZUBRUTIN TALÁL SZINTAKTIKUS HIBÁT A ZSÁKKÁRTYÁK FELDOLGOZÁSA KÖZBEN. STOP.
A fentiek alapján elmondhatjuk, hogy a program többékevésbé képes a zsákleiró kártyák szintaktikázására.

A 7.2.1. pontban szereplő hiba zsákkezdő kártyában fatális hiba, különben nem fatális hiba.

Ha a program hibás zsákleiró kártyát talál, akkor a hibaüzenet után a hibás kártyát újra kiprinteli. "Egyesével" - kártya esetén előfordul, hogy a 2-5. karakterre nem az oda lyukasztott számot, hanem annak valamennyivel megnövelt értékét nyomtatja ki ilyen esetben a szubrutin.

A továbbiakban felsoroljuk és szükség esetén megmagyarázzuk a szubrutin hibaüzeneteit.

7.1. Rossz szubrutin paraméter

Ilyen hiba esetén a következő tipusú szöveg iródik ki:

ROSSZ SUBROUTINE-PARAMÉTER p

esetleg egyéb szöveg

$p_1, p_2, p_3, \ldots, p_m$

ahol $1 \leq p \leq 4$ egész szám; arra utal, melyik paraméterre nem teljesülnek az 5. pontban felsorolt feltételek /1-ZSSZ,NBE; 2-ZSN; 3-RKK,RKV; 4-GY/; $p_1 p_2, \ldots, p_m$ pedig a hibás paraméter.

Pl.

ROSSZ SUBROUTINE-PARAMÉTER 2
AZ 1. ÉS A 4. ZSÁKNÉV AZONOS

7 3 4 7

/7,3,4,7 és 9 az öt zsáknév/

7.2. Hibák a régi kódok megadásában

7.2.1. "6. KARAKTER NEM ÜRES"

7.2.2. "MEG NEM ENGEDETT KARAKTER A JOBBOLDALON"

Pl. 45,4L
A 7.2. pontban adott példákban a hibás zsákleiró kártyának csak a comment nélküli jobboldalát /a 7-75. karaktert/ irjuk le.

7.2.3. "TUL KICSI RÉGI KÓD"

Azaz a régi kód kisebb, mint a megfelelő RKK(i)

7.2.4. "TUL NAGY RÉGI KÓD"

Azaz a régi kód nagyobb, mint a megfelelő RKV(i)

7.2.5. "- VAGY, ELŐTT NINCS SZÁM VAGY A KÁRTYA NEM SZÁMMAL VÉGZŐDIK"

Példák: 79,-60
53,,55
782,89,

7.2.6. "X. HELYRE KÉTSZER AKAR BEÍRNI.
KORÁBBI UJ KÓD: Y"

Ilyen hibajelzést akkor kapunk, ha egy x régi kód értéknek már adtunk egy y új kódot /és y≠0/ és most egy újabb új kód értéket akarunk adni. A program akkor is hibát jelez, ha egy régi kód értéknek kétszer akarjuk ugyanazt a 0-tól különböző új kód értéket adni.

7.2.7. "HIBA: X-Y X>Y"

Pl. HIBA: 19-12 19>12
/tehát egy intervallum kezdőpontja nagyobb mint a végpont/

7.2.8. "HIBA: -SZÁM-"
Pl. 93-95-97

7.3. Egyéb hibák

7.3.1. "3 UTÁN NEM ÜRES A VÉGE-KÁRTYA"

Rossz "zsákleirás vége" - kártya
Pl. |3| 1

7.3.2. "AZ ELSŐ ZSÁKLEIRÓ KÁRTYA NEM ZSÁKKEZDŐ KÁRTYA"

7.3.3. "ISMERETLEN ZSÁKNÉV: k"

A "|2| k|" zsákkezdő kártyában levő k nem szerepel a zsáknevek közt

7.3.4. "MEG NEM ENGEDETT ÉRTÉK AZ 1. HELYEN"

Az 1. helyen 4-nél nagyobb egész szám áll.
Pl. |5|vv23|47-51

7.3.5. "NINCS VÉGE-KÁRTYA (3) *** FATÁLIS HIBA ***"

Ilyen hibaüzenetet akkor kap a felhasználó, ha nem tesz "zsákleirás vége"-kártyát a zsákokat leiró kártyák után /ld. 6.2. pont/.

7.3.6. "k NEVÜ ZSÁK x ÉRTÉKÜ RÉGI KÓDJÁHOZ RENDELT UJ KÓD ÉRTÉK (=y) NEM JÓ: y < 1 VAGY y > m"

Pl.
13 NEVÜ ZSÁK 9 ÉRTÉKÜ RÉGI KÓDJÁHOZ RENDELT UJ KÓD ÉRTÉK (=22) NEM JÓ: 22 < 1 VAGY 22 > 10

Ilyen hiba akkor lép fel, ha kérünk beszorzást (NBE > 0) és a k nevü beszorzandó zsákban /amelyre az "új kód maximális értéke" m/ az x régi kód értékhez rendelt új kód értéke (=y) egynél kisebb vagy az m maximális értéknél nagyobb. A példában a 13 nevü beszorzandó zsák új kódjainak maximális értékére 10-et adtunk meg, és a 13 nevü zsákban a 9 régi kód értékhez a 22 új kód értéket rendeltük. A fenti hibaüzenet után a szubrutin negyvenszer kiirja azt, hogy "FÉLBESZAKITJUK A BESZORZÁST". Ilyen hiba esetén a szubrutin félbeszakitja a beszorzást, kiirja a zsákok tartalmát /ha a felhasználó ezt kérte, azaz KIIRZS értéke 1 volt/, majd STOP-pal megáll.

8. A felhasználás tapasztalatai

A ZSÁK szubrutin a SIS77 statisztikai információs rendszer eleme /ld. [3]/, de attól függetlenül önállóan is használható. A SIS77 rendszert 1977 óta sikeresen alkalmazzák az országos hospitalizált morbiditási vizsgálatokban, azaz a magyarországi kórházakban ápolt betegekről felvett adatok

feldolgozásában /ld. [2] /. Ezen vizsgálatokban a ZSÁK szubrutin segitségével különböző feladatokat oldottunk meg /ld. [3] 3.10 pontja és [4] /, igy ezt a szubrutint használtuk átkódolási feladatoknál, gyakoriságszámlálás során, többdimenziós táblázatok összevonásakor, bonyolult logikai kifejezésekben szereplő logikai változók értékeit megadó táblázatok kitöltésénél /ld. [1]/. A szubrutin segitségével valósitottuk meg többváltozós függvények értéktáblázatainak kitöltését is oly módon, hogy a táblázatokat egy hierarchikus gráfban helyeztük el /ld. [3] 3.1. pontja/. Hierarchikus gráfokat és a ZSÁK szubrutint használtuk adatcsoportok összeférhetőségének ellenőrzésekor is, pl. amikor a kórházi morbiditási vizsgálatok során azt ellenőriztük, hogy összefér-e a beteg diagnózisa az életkorával és a nemével /ld. [3] 3.1 pontja/.

A futtatási tapasztalatok alapján a ZSÁK szubrutin kényelmes, jól használható, biztonságos, megbizható. A szubrutin alkalmazása különösen nagyméretü táblázatok kitöltésénél nyújtott óriási segitséget a felhasználóknak.

9. Két mintapélda

9.1. Az első mintafutás feladatának leirása és az output lista

A feladat

Kérünk beszorzást.

4 értéktáblázatot kell készitenünk. Ezek nevei, valamint értelmezési tartományaik alsó és felső határai:

1: 86-95
2: 1-20
3: 1-10
4: 0-14

3 zsákra kérünk beszerzást. Ezek nevei, valamint értékkészleteik felső korlátai /a legkülső dimenziótól a legbelső felé haladva/:

4: 4
2: 7
3: 3

Kérjük a zsákok tartalmának kinyomtatását.

Ehhez a feladathoz a <u>bemenő paramétereket</u> a következő módon kell megadnunk:

```
ZSSZ=4
NBE=3
ZSN(1)=1     RKK(1)=86     RKV(1)=95
ZSN(2)=2     RKK(2)=1      RKV(2)=20
ZSN(3)=3     RKK(3)=1      RKV(3)=10
ZSN(4)=4     RKK(4)=0      RKV(4)=14
GY(1)= 4     UKM(1)=4      KIIRZS=1
GY(2)= 2     UKM(2)=7
GY(3) =3     UKM(3)=3
```

```
**********************************************************************

 MTA SZTAKI SIS77     MTA SZTAKI SIS77     MTA SZTAKI SIS77

**********************************************************************

                        ********************************
                        *                              *
                        *   A ZSAK SZUBRUTIN FUTASA    *
                        *                              *
                        ********************************


                         ************************
                         TABLAZATOK OSSZEVONASA
                         ************************


EREDETI DIMENZIOSZAM:  4
A DIMENZIOK NEVEI (SORSZAMAI),
VALAMINT AZ EREDETI KOIOK ALSO ES FELSO HATARAI:
      1 :    86-    95
      2 :     1-    20
      3 :     1-    10
      4 :     0-    14


UJ DIMENZIOSZAM:  3
AZ UJ DIMENZIOK NEVEI (SORSZAMAI), VALAMINT MERETEI
(A DIMENZIOK MEGADASI SORRENDJE KIVULROL BEFELE TORTENIK):
      4 :     4
      2 :     7
      3 :     3


KEREM A ZSAKOK TARTALMANAK KINYOMTATASAT


A ZSAKOK EGYUTTES HOSSZA (A Z VEKTOR MERETE):      55
```

```
**********************************************************************

 MTA SZTAKI SIS77     MTA SZTAKI SIS77     MTA SZTAKI SIS77

**********************************************************************

                          **********************************
                          A BEOLVASOTT ZSAKLEIRO KARTYAK
                          **********************************


          UJ ZSAK KEZDODIK.ZSAKNEV:      4

2    4
1    1 0-3
0    1 7,14,10
0    4 11-13
4    2 0-14


          UJ ZSAK KEZDODIK.ZSAKNEV:      3

2    3           SZOVEG   EEEEE
0    1 1-4
0    2 5-7
0    3 8-10


          UJ ZSAK KEZDODIK.ZSAKNEV:      2

2    2
1    1 1-5
4    7 1-20


          UJ ZSAK KEZDODIK.ZSAKNEV:      1

2    1                                                    1
1    0    86-95                                           2

3    0                                                VEGE

ZSAKLEIRAS VEGE

**********************************************************************
```

****

MTA SZTAKI SIS77 MTA SZTAKI SIS77 MTA SZTAKI SIS77 MTA SZTAKI SIS77

****

****
A ZSAKOK TARTALMA
****

ZSAKNEV: 1 REGI KOD HATARAI: 86- 95

0 1 2 3 4 5 6 7 8 9

ZSAKNEV: 2 REGI KOD HATARAI: 1- 20

1 2 3 4 5 7 7 7 7 7
7 7 7 7 7 7 7 7 7 7

ZSAKNEV: 3 REGI KOD HATARAI: 1- 10

1 1 1 1 2 2 2 3 3 3

ZSAKNEV: 4 REGI KOD HATARAI: 0- 14

1 2 3 4 2 2 2 1 2 2
1 4 4 4 1

****
A ZSAKOK TARTALMA BESZORZAS UTAN
****

ZSAKNEV: 1 REGI KOD HATARAI: 86- 95

0 1 2 3 4 5 6 7 8 9

ZSAKNEV: 2 REGI KOD HATARAI: 1- 20

0 3 6 9 12 18 18 18 18 18
18 18 18 18 18 18 18 18 18 18

ZSAKNEV: 3 REGI KOD HATARAI: 1- 10

1 1 1 1 2 2 2 3 3 3

ZSAKNEV: 4 REGI KOD HATARAI: 0- 14

0 21 42 63 21 21 21 0 21 21
0 63 63 63 0

9.2. A második mintafutás feladatának leirása és az output lista

A feladat:

Nem kérünk beszorzást. 4 értéktáblázatot készitünk. Ezek nevei, valamint értelmezési tartományaik alsó és felső határai:

11:	1-60
-22:	1-10
33:	0-9
44:	1-10

Kérjük a zsákok tartalmának kinyomtatását.

Ehhez a feladathoz a bemenő paramétereket a következő módon kell megadnunk:

```
ZSSZ=4
NBE=0
ZSN(1)=11     RKK(1)=1     RKV(1)=60
ZSN(2)=-22    RKK(2)=1     RKV(2)=10
ZSN(3)=33     RKK(3)=0     RKV(3)=9
ZSN(4)=44     RKK(4)=1     RKV(4)=10
KIIRZS=1
```

A GY és az UKM vektornak nem kell értéket adnunk.

A szubrutin több szintaktikusan hibás zsákleiró kártyát talált. Ezeket hibaüzenettel kinyomtatta, majd kiirta a zsákok tartalmát, és végül ötvenötször kinyomtatta az "A SZUBRUTIN TALÁLT SZINTAKTIKUS HIBÁT A ZSÁK-KÁRTYÁK FELDOLGOZÁSA KÖZBEN. STOP" szöveget. Ezek után a szubrutin egy STOP utasitásra ugrott és megállt.

Megjegyzések:

1. A mintafutáshoz tartozó zsákleiró kártyák közt található

"0␣␣␣3␣34, 13-15" kártya "13-15" része a 76 -80.

pozicióba került. Mivel a szubrutin ezeket a pozíciókat comment-nek tekinti, ez a kártya hibás /olyan, mintha a "34" régi kód utáni vesszővel végződött volna/.

2. A "0␣␣␣8␣91-93-95" zsákleiró kártyát a szubrutin azért találta hibásnak, mert a 91 régi kód érték nagyobb a 11 nevü zsákra "régi kód végértéknél", 60-nál /RKK(1)=60/. A ZSAK szubrutin akkor is csak egy hibát ir ki, ha egy zsákleiró kártyán több hiba van. A fenti kártyánál igy a szubrutin nem adott "HIBA: -SZÁM-" hibaüzenetet.

3. A szubrutin a zsákleiró kártyákat balról jobbra haladva dolgozza fel. Igy ha pl. egy "közönséges" zsákleiró kártyának 1 -6. pozicióját, majd néhány régi kódját jól adjuk meg, azután hibát csinálunk, majd a további régi kód értékeket jól adjuk meg, akkor a szubrutin a hiba előtti régi kódokhoz hozzárendeli a megadott új kód értéket, mig a hibás, és a hiba utáni régi kód értékekhez nem rendel új kód értéket.

Példák:

a/ A "0␣␣12␣50,51" kártyán az 50 régi kód érték hibás, mert ahhoz már korábban hozzárendeltük a 10 új kód értéket. Igy az 50 régi kódhoz rendelt új kód érték 10 marad, a hiba utáni 51 régi kód értékhez rendelt új kód érték pedig 0 marad, mint az a zsákok kinyomtatott tartalmából is látszik.

b/ A "0␣␣␣9␣56,,57" kártya hibás, mivel az 56 régi kód érték után két vessző áll. A szubrutin a hiba előtti 56 régi kód értékhez hozzárendeli a 9 új kód értéket, mig a hiba utáni 57 régi kód értékhez rendelt új kód érték 0 marad.

```
**********************************************************************

 MTA SZTAKI SIS77     MTA SZTAKI SIS77     MTA SZTAKI SIS77

**********************************************************************

                  ********************************
                  *                              *
                  *  A ZSAK SZUBRUTIN FUTASA     *
                  *                              *
                  ********************************


                        ***********
                        OSZTALYOZAS
                        ***********

AZ OSZTALYOZASI SZEMPONTOK SZAMA:  4
AZ OSZTALYOZASI SZEMPONTOK NEVEI (SORSZAMAI),
VALAMINT AZ EREDETI KODOK ALSO ES FELSO HATARAI:
     11 :     1-   60
    -22 :     1-   10
     33 :     0-    9
     44 :     1-   10


KEREM A ZSAKOK TARTALMANAK KINYOMTATASAT


A ZSAKOK EGYUTTES HOSSZA (A Z VEKTOR MERETE):      90
```

```
**********************************************************************************************************
MTA SZTAKI SIS77    MTA SZTAKI SIS77    MTA SZTAKI SIS77    MTA SZTAKI SIS77    MTA SZTAKI SIS77
**********************************************************************************************************

                    **************************
                    A BEOLVASOTT ZSAKLEIRO KARTYAK
                    **************************


          UJ ZSAK KEZDO IIK.ZSAKNEV:    11

2  11       SZOVEG   BBBBBBB
0   1 2-10, 16, 12, 23
0   1 25, 29-32
0   2 11,17-20,28
0   3 34  ,                                                          13-15

- VAGY , ELOTT NINCS SZAM, VAGY A KARTYA NEM SZAMMAL VEGZODIK
0   3 34  ,                                                          13-15   ***HIBA***

0   4 4  3, 3 9 -  41
0  10 37,50
0  12 50,51

   50.HELYRE KETSZER AKAR BEIRNI.KORABBI UJ KOD:   10
0  12 50,51                                                                  ***HIBA***

0   7 52,54,

- VAGY , ELOTT NINCS SZAM, VAGY A KARTYA NEM SZAMMAL VEGZODIK
0   7 52,54)                                                                 ***HIBA***

0   8 91-93-95

TUL NAGY REGI KOD
0   8 91-93-95                                                               ***HIBA***

0   9 56,,57

- VAGY , ELOTT NINCS SZAM, VAGY A KARTYA NEM SZAMMAL VEGZODIK
0   9 56,,57                                                                 ***HIBA***

0  13  45,4L

MEG NEM ENGEDETT KARAKTER A JOBB OLDALON
0  13  45,4L                                                                 ***HIBA***
```

```
          UJ ZSAK KEZDODIK.ZSAKNEV:    -22


2 -22
1 -29  1-8


          UJ ZSAK KEZDODIK.ZSAKNEV:     33

2  33
0  33 0,4,5,9
4   3 0-9


          UJ ZSAK KEZDODIK.ZSAKNEV:     44

2  44
0  -1 1-3,0

TUL KICSI REGI KOD
0  -1 1-3,0                                          ***HIBA***

1   1 5-8,10
4  44 2-10
0  20 11

TUL NAGY REGI KOD
0  20 11                                             ***HIBA***


31000

3 UTAN NEM URES A VEGE-KARTYA
31000                                                ***HIBA***


3   0

ZSAKLEIRAS VEGE

*****************************************************************
```

```
****************************************************************************************************
MTA SZTAKI SIS77    MTA SZTAKI SIS77    MTA SZTAKI SIS77    MTA SZTAKI SIS77    MTA SZTAKI SIS77
****************************************************************************************************

                               *****************
                               A ZSAKOK TARTALMA
                               *****************

ZSAKNEV:   11      REGI KOD HATARAI:    1-   50

                     0    1    1    1    1    1    1    1    1    1
                     2    1    0    0    0    1    2    2    2    2
                     0    0    1    0    1    0    0    2    1    1
                     1    1    0    3    0    0   10    0    4    4
                     4    0    4    0   13    0    0    0    0   10
                     0    7    0    7    0    9    0    0    0    0

ZSAKNEV:  -22      REGI KOD HATARAI:    1-   10

                   -29  -23  -27  -26  -25  -24  -23  -22    0    0

ZSAKNEV:   33      REGI KOD HATARAI:    0-    9

                    33    3    3    3   33   33    3    3    3   33

ZSAKNEV:   44      REGI KOD HATARAI:    1-   10

                    -1   -1   -1   44    1    2    3    4   44    5

************************** A SZUBRUTIN TALALT SZINTAKTIKUS HIBAT A ZSAK-KARTYAK FELDOLGOZASA KOZBEN. STOP
************************** A SZUBRUTIN TALALT SZINTAKTIKUS HIBAT A ZSAK-KARTYAK FELDOLGOZASA KOZBEN. STOP
************************** A SZUBRUTIN TALALT SZINTAKTIKUS HIBAT A ZSAK-KARTYAK FELDOLGOZASA KOZBEN. STOP
************************** A SZUBRUTIN TALALT SZINTAKTIKUS HIBAT A ZSAK-KARTYAK FELDOLGOZASA KOZBEN. STOP
************************** A SZUBRUTIN TALALT SZINTAKTIKUS HIBAT A ZSAK-KARTYAK FELDOLGOZASA KOZBEN. STOP
************************** A SZUBRUTIN TALALT SZINTAKTIKUS HIBAT A ZSAK-KARTYAK FELDOLGOZASA KOZBEN. STOP
************************** A SZUBRUTIN TALALT SZINTAKTIKUS HIBAT A ZSAK-KARTYAK FELDOLGOZASA KOZBEN. STOP
************************** A SZUBRUTIN TALALT SZINTAKTIKUS HIBAT A ZSAK-KARTYAK FELDOLGOZASA KOZBEN. STOP
************************** A SZUBRUTIN TALALT SZINTAKTIKUS HIBAT A ZSAK-KARTYAK FELDOLGOZASA KOZBEN. STOP
************************** A SZUBRUTIN TALALT SZINTAKTIKUS HIBAT A ZSAK-KARTYAK FELDOLGOZASA KOZBEN. STOP
************************** A SZUBRUTIN TALALT SZINTAKTIKUS HIBAT A ZSAK-KARTYAK FELDOLGOZASA KOZBEN. STOP
************************** A SZUBRUTIN TALALT SZINTAKTIKUS HIBAT A ZSAK-KARTYAK FELDOLGOZASA KOZBEN. STOP
************************** A SZUBRUTIN TALALT SZINTAKTIKUS HIBAT A ZSAK-KARTYAK FELDOLGOZASA KOZBEN. STOP
************************** A SZUBRUTIN TALALT SZINTAKTIKUS HIBAT A ZSAK-KARTYAK FELDOLGOZASA KOZBEN. STOP
************************** A SZUBRUTIN TALALT SZINTAKTIKUS HIBAT A ZSAK-KARTYAK FELDOLGOZASA KOZBEN. STOP
************************** A SZUBRUTIN TALALT SZINTAKTIKUS HIBAT A ZSAK-KARTYAK FELDOLGOZASA KOZBEN. STOP
************************** A SZUBRUTIN TALALT SZINTAKTIKUS HIBAT A ZSAK-KARTYAK FELDOLGOZASA KOZBEN. STOP
************************** A SZUBRUTIN TALALT SZINTAKTIKUS HIBAT A ZSAK-KARTYAK FELDOLGOZASA KOZBEN. STOP
************************** A SZUBRUTIN TALALT SZINTAKTIKUS HIBAT A ZSAK-KARTYAK FELDOLGOZASA KOZBEN. STOP
************************** A SZUBRUTIN TALALT SZINTAKTIKUS HIBAT A ZSAK-KARTYAK FELDOLGOZASA KOZBEN. STOP
************************** A SZUBRUTIN TALALT SZINTAKTIKUS HIBAT A ZSAK-KARTYAK FELDOLGOZASA KOZBEN. STOP
************************** A SZUBRUTIN TALALT SZINTAKTIKUS HIBAT A ZSAK-KARTYAK FELDOLGOZASA KOZBEN. STOP
************************** A SZUBRUTIN TALALT SZINTAKTIKUS HIBAT A ZSAK-KARTYAK FELDOLGOZASA KOZBEN. STOP
************************** A SZUBRUTIN TALALT SZINTAKTIKUS HIBAT A ZSAK-KARTYAK FELDOLGOZASA KOZBEN. STOP
```

********** A SZUBRUTIN TALALT SZINTAKTIKUS HIBAT A ZSAK-KARTYAK FELDOLGOZASA KOZBEN. STOP

I R O D A L O M

[1] Ratkó I.: Bonyolult logikai kifejezések kiértékelésének számitástechnikai és optimalizálási problémái, MTA SZTAKI Közlemények, 2o/1978.

[2] Ruda M.: A SIS77 statisztikai információs rendszer kialakitásának szempontjai, alkalmazásának és továbbfejlesztésének lehetőségei, MTA SZTAKI Tanulmányok, 86/1978.

[3] Ruda M.: A SIS77 statisztikai információs rendszer, MTA SZTAKI Tanulmányok, 89/1978.

[4] Soltész J.: Egy általánosan használható kódolási eljárás és alkalmazása a hospitalizált morbiditási vizsgálatokban, Számitástechnikai és kibernetikai módszerek alkalmazása az orvostudományban és a biológiában, Neumann J.Sz.T. 9. Kollokviuma, Szeged, 1978.

S U M M A R Y

A general purpose procedure to fill up central memory tables /Program description/

A subroutine is described that facilitates filling up tables /code-dictionaries, jump-tables, function tables/ located in the central memory. The advantages in using this subroutine are

a/ the tables can be filled up in the possibly most compact and comfortable way

b/ appropriate safety is provided by the syntactical analysis of the table descriptions and the detailed error messages

This subroutine provides means for a quick and easy calculation of some multivariate functions.

Programleirás

Gyors olvasó eljárások létrehozása FORTRAN programokban

Gál Anna - Ruda Mihály

1. A program célja

FORTRAN nyelvü programokban a szokásosnál lényegesen gyorsabb olvasási és konvertálási eljárás biztositása /ld. [1] /.

2. A program formája, a felhasznált gép

FORTRAN szubrutin, hivása:

CALL LECTOR (JSZ, JFC, JEND, JERR, JHIBA)

A felhasznált gép: Honeywell 66/60

3. A feladat részletes leirása, a felhasznált módszerek

3.1 A szubrutint "szerkesztő" programok hivhatják.
Ezek a "szerkesztő" /generáló/ programok gazdaságos müködésü programok előállitását végzik el. /ld. [1] / A szerkesztő program a "szerkesztett" /generált/ programot egy kijelölt file-on helyezi el, ahonnan az forditható és futtatható.

Hivásakor a szubrutin a szerkesztett programba egy néhány utasitásból álló részt ir. Ez a rész megfelel egy olyan olvasási utasitásnak, amely a K_1, K_2 ... változókba olvas be karakter formájú rekordot, megadott formátum szerint. /A változók száma legfeljebb 999 lehet./ A formátum I, A vagy X specifikációkat tartalmazhat.

A szubrutin által szerkesztett rész első utasitása egy olvasási utasitás, amely a rekordot binárisan olvassa be az ISZ1, ISZ2 ... változókba. /A változók száma itt is legfeljebb 999 lehet./ Az egyes szavakba igy 6-6 karakter kerül.

A K1, K2 ... változóknak ezután a következő módon ad értéket a program:

3.2 Az I specifikációnál csak nem negativ, legfeljebb 9 jegyü értékek szerepelhetnek. A szubrutin a formátum alapján kiszámitja, hogy az egyes számjegyeknek megfelelő karakterek hányadik szóban találhatók, és a szó hányadik bitjén kezdődnek. Ezután olyan utasitásokat szerkeszt, amelyek ezeket a karaktereket egy-egy külön változóban helyezik el. Ezt az FLD függvény /Honeywell FORTRAN eljárás/ használata teszi lehetővé. Pl.: az

$$LN = FLD(I,6,ISZK) \qquad (1)$$

utasitás hatására az LN változó utolsó 6 bitjére az ISZK szó I-edik bitjétől I+5-ödik bitjéig tartó bitsorozat kerül. A szerkesztett program igy külön változókban(L1,L2,...) helyezi el a konvertálandó egész szám egyes számjegyeit. A számjegyekből a számot a

$$KM = I1(L1) + I2(L2) + I3(L3) + \ldots + IN(LN) \qquad (2)$$

alakú utasitás állitja elő, ahol

$$\begin{aligned} I1(L1) &= L1 \\ I2(L2) &= L2 * 10 \\ I3(L3) &= L3 * 10^2 \\ &\cdot \\ &\cdot \\ &\cdot \\ IN(LN) &= LN * 10^{N-1} \end{aligned} \qquad (3)$$

és L1, L2,...,LN az egyes számjegyek értékei /az egyes helyiértéktől kezdve/. A hibás karakterekhez a szerkesztett konvertáló eljárás 0-t rendel.

Az I1(I), I2(I),... ,I9(I) vektoroknak értéket adó utasitásokat a szubrutin beszerkeszti a szerkesztett programba. Ehhez a szerkesztő programnak a tényleges hivások előtt egyszer speciális paraméterekkel kell hivnia a szubrutint. /ld. a paraméterek leirását, 4.3 pont./

3.3 Az A specifikációnál A6 a legnagyobb megengedett formátumelem. A szubrutin olyan utasitásokat szerkeszt, amelyek a szükséges karaktereket átteszik a K1,K2 ... változókba. Az

FLD(I,J,KM)= FLD(L,J,ISZK) (4)

utasitás hatására a KM változó I-edik bitjétől I+J-1-edik bitjéig tartó részére az ISZK szó L-edik bitjétől L+J-1-edik bitjéig tartó bitsorozat kerül. A K1,K2 ... változók betöltése a szavak elején kezdődik.

3.4 A szerkesztett program többi részében az olvasó programrészletben lefoglalt cimkék: 8888, 9999 és tömbök I1(64), I2(64),...,I9(64), NU1(2), NU2(2),... ,NU9(2)nem használhatók.

4. A szubrutin paraméterei

4.1 Bemenő paraméterek:

JSZ A szerkesztett programot tartalmazó file kódja /1 és 99 között lehet/.

JFC Az input file kódja /1 és 99 között lehet/

JEND A cimke értéke, ahová file-vége jelzésnél ugrani kell /1 és 99999 között lehet/

JERR A cimke értéke, ahová olvasási hiba esetén ugrani kell /1 és 99999 között lehet/

4.2 Kimenő paraméter

JHIBA Hibajelző paraméter
Ha a szubrutin valamilyen hibát talál, akkor nem kezdi el a szerkesztést, a sornyomtatón hibajelzést ir, és visszatér a hivó programba JHIBA = 1 értékkel. Hibátlan futás esetén JHIBA = 0.

4.3 A szerkesztő programnak az olvasó utasitásokat szerkesztő hivások előtt egyszer speciális paraméterekkel kell hivnia a szubrutint. Ennél az első hivásnál a szubrutin megszerkeszti az I specifikációnál szükséges I1,...,I9, NU1, ...,NU9 tömböket deklaráló és azoknak értéket adó (3) utasitásokat. Az első hivásnál meg kell adni az első paraméter /a szerkesztett programot tartalmazó file kódja/ értékét, a második paraméter értékének pedig 0-nak kell lennie. A többi paraméter értéke ekkor tetszőleges lehet.

5. A bemenő adatok: Az input formátum

A szubrutin kártyaolvasóról olvassa be az input formátumot. A formátum tartalmazhat I, A és X specifikációt. Az I specifikációnál 9, az A specifikációnál 6 a legnagyobb megengedett mezőszélesség. A beolvasandó adatok száma nem lehet nagyobb, mint 999, és a rekord beolvasandó részének el kell férnie 999 szóban. Lehet használni ismétlési tényezőket /pl. 20A6, 5I4/, de zárójeles ismétlési csoportokat nem. Nem használható "/" jel sem.

A formátumot egy kártyán nyitó- és záró jellel együtt kell megadni. O helyett nem szabad szóközt használni. A formátum szabályos FORTRAN formátum kell hogy legyen.

6. A kimenő adatok

6.1 A szubrutin a sornyomtatón kinyomtatja az input formátumot.

6.2 Az ellenőrzések során talált hibákról a következő hibajelzéseket nyomtatja:

- A bemenő paraméterek ellenőrzése

HIÁNYZIK A DEKLARÁLÓ ÉS ÉRTÉKADÓ UTASITÁSOK SZERKESZTÉSÉNEK HIVÁSA
/ha az első hivásnál a második paraméter - az input file kódja - értéke nem 0/.
AZ INPUT FILE CODE NEM 1 ÉS 99 KÖZÉ ESIK
A SZERKESZTETT PROGRAMOT TARTALMAZÓ FILE CODE NEM 1 ÉS 99 KÖZÉ ESIK
A HARMADIK PARAMÉTER /JEND/ ÉRTÉKE NEM 1 ÉS 99999 KÖZÉ ESIK
A NEGYEDIK PARAMÉTER /JERR/ ÉRTÉKE NEM 1 ÉS 99999 KÖZÉ ESIK

- A formatum ellenőrzése

A FORMATUM HELYE ÜRES
A FORMATUM NEM NYITÓZÁRÓJELLEL KEZDŐDIK
NYITÓZÁRÓJEL VAGY VESSZŐ UTÁN HIBÁS KARAKTER KÖVETKEZIK
A SZÜKSÉGES HELYEN NEM SZEREPEL X, I VAGY A
AZ ISMÉTLÉSI TÉNYEZŐ HELYÉN 0 ÁLL
0 MEZŐSZÉLESSÉG SZEREPEL
I VAGY A UTÁN A MEZŐSZÉLESSÉG HELYETT HIBÁS KARAKTER KÖVETKEZIK
"A" FORMATUMBAN 6-NÁL NAGYOBB MEZŐSZÉLESSÉG SZEREPEL
A SZÜKSÉGES HELYEN NEM SZEREPEL VESSZŐ VAGY ZÁRÓJEL
/Ez a hibaüzenet jelenik meg akkor is, ha "1" formátumban 9-nél nagyobb mezőszélesség szerepel/
A FORMATUM NEM ZÁRÓJELLEL VÉGZŐDIK
A BEOLVASANDÓ SZAVAK SZÁMA NAGYOBB MINT 999
A BEOLVASANDÓ ADATOK SZÁMA NAGYOBB MINT 999
A BEOLVASANDÓ ADATOK SZÁMA 0

6.3 A szubrutin az első paraméterben /JSZ/ megadott kódú file-ra programrészletet ir. Az első hivásnál szükséges deklaráló és értékadó utasitásokat irja a file-ra. A további hivásoknál irt programrészletek egy-egy olvasási utasitást helyettesitenek.

7. A felhasználás tapasztalatai

Az alábbi táblázat olyan programok futásidőit mutatja, amelyek 153426 rekordot olvastak be hagyományos módon illetve a LECTOR szubrutin segitségével / az időértékek a központi egység időértékei, a csatornaidők minden esetben 0,08 - 0,09 óra közti értékek /.

FORMÁTUM	FUTÁSI IDŐ /ÓRA/		IDŐARÁNY /KÖZELITŐLEG/
	1.	2.	
(48X,A1,36X,A2,69X,I1)	0.2525	0.0470	5.37
(A6,29A1,I3,34I1)	0.5351	0.0695	7.70
(20A1)	0.2196	0.0421	5.22
(I9,I8,4A6,I2,3A2,4I1,2A1,2I2, A5,3I3,7X,7A1,A6,I8,2A4,2I3, 2I5,4X,2A4,6X,2I5,4A1)	0.6352	0.0907	7.00
(2I9,48X,I5,79X,5A2)	0.3091	0.0578	5.35
(28A1,3X,19A1,4X,21A1,8I1,2X, 8A1,1X,23A1,2X,12A1,3X,18A1, 1X,A1)	1.2414	0.1130	10.99
(19A1,4X,4I5,11I1,A3)	0.3605	0.0596	6.05

A táblázat utolsó oszlopából látható , hogy az adott formátumtól függően változó , de mindenképpen jelentős arányu gyorsulásról van szó. Az olvasás az adatok bemozgatásából és konvertálásából áll. Tisztán a konverziós időket tekintve / pl. DECODE utasitásnál / még nagyobb sebességnövekedés érhető el / ld. [1] / .

/ 1.: hagyományos olvasás
2.: olvasás a LECTOR szubrutint használva /

8. Egy felhasználási példa

Illusztráció képpen bemutatunk egy programot, amely mint szerkesztő /generáló/ program felhasználja a LECTOR szubrutint.

A most következő első oldalon a szerkesztő program látható. A bemutatott program két különböző formátummal müködő olvasási eljárást állit elő / kétszer hivja a LECTOR szubrutint /.

A második oldalon a felhasználó által megadott két formátum olvasható. Ezeket a LECTOR szubrutin irta ki az egyes hivásokkor.

A harmadik és negyedik oldalon a szerkesztett program látható a szerkesztő program és azon belül a LECTOR szubrutin által létrehozott utasitásokkal.

```
E3304 01  07-14-8-    21.708                SZERKESZTO PROGRAM                                LABEL ......  PAGE      1

                      C                      SZERKESZTO PROGRAM
                      C
                            WRITE(12,1)
                          1 FORMAT("C",20X,"SZERKESZTETT PROGRAM")
                            WRITE(12,6)
                            WRITE(12,5)
                          5 FORMAT("C",5X,"DEKLARALO ÉS ERTEKADO UTASITASOK"/"C")
                            JSZ=12
                      C
                      C     AZ ELSO HIVAS SPECIALIS PARAMETEREKKEL
                      C
                            JFC=3
                            CALL LECTOR(JSZ,JFC,JEND,JERR,JHIBA)
                            IF(JHIBA.EQ.1) STOP
                            WRITE(12,6)
                            WRITE(12,2)
                          2 FORMAT(6X,"JR=1"/"C"/"C",5X,"OLVASAS AZ ELSO FORMATUMMAL"/"C"/4X,
                           *"1 CONTINUE")
                            JFC=1
                            JEND=3
                            JERR=4
                      C
                      C     HIVAS AZ ELSO FORMATUMMAL
                      C
                            CALL LECTOR(JSZ,JFC,JEND,JERR,JHIBA)
                            IF(JHIBA.EQ.1) STOP
                            WRITE(12,6)
                            WRITE(12,3)
                          3 FORMAT(6X,"JR=JR+1"/6X,"IF(JR-100)1,1,2"/"C"/"C",5X,"OLVASAS A MAS
                           *ODIK FORMATUMMAL"/"C"/4X,"2 CONTINUE")
                      C
                      C     HIVAS A MASODIK FORMATUMMAL
                      C
                            CALL LECTOR(JSZ,JFC,JEND,JERR,JHIBA)
                            IF(JHIBA.EQ.1) STOP
                            WRITE(12,6)
                            WRITE(12,4)
                          4 FORMAT(6X,"JR=JR+1"/6X,"GO TO 2"/4X,"4 PRINT 7,JR"/4X,"7 FORMAT(/2
                           *X,I1),24H,REKORDNAL OLVASASI HIBA)"/6X,"JR=JR+1"/6X,"GO TO 2"/4X,
                           *"3 STOP"/6X,"END")
                          6 FORMAT("C"/"C",5X,"A SZERKESZTETT PROGRAM TETSZOLEGES UTASITASAI"/
                           *"C")
                            STOP
                            END


   EDIT DATE   12 31-79  **$R4J**

ELAPSED TIME (SEC        .76          LINES/MINUTE  3449

  THERE WERE  NO   DIAGNOSTICS IN ABOVE COMPILATION
   26K  WORDS WER  USED FOR THIS COMPILATION


SNUMB = E3304,  CTIVITY # = 02, , REPORT CODE = 52, RECORD COUNT = 000004

A FORMATUM: ( I, 41X, I4, 15X, A3 )

A FORMATUM:  (  54X , 2 I2, 87X, A 6 )
```

```
E3304 03  07-14-8    21.715                SZERKESZTETT PROGRAM                    LABEL ......  PAGE    1
        C                      SZERKESZTETT PROGRAM
        C
      . C      A SZERKESZTETT PROGRAM TETSZOLEGES UTASITASAI
        C
        C      DEKLARALO ES ERTEKADO UTASITASOK
        C
               DIMENSION NU1(2),I1(63),NU2(2),I2(63),NU3(2),I3(63),NU4(2),I4(63)
              *,NU5(2),I5(63),NU6(2),I6(63),NU7(2),I7(63),NU8(2),I8(63)
              *,NU9(2),I9(63)
               EQUIVALENCE (NU1(2),I1(1)),(NU2(2),I2(1))
              *,(NU3(2),I3(1)),(NU4(2),I4(1)),(NU5(2),I5(1))
              *,(NU5(2),I6(1)),(NU7(2),I7(1)),(NU8(2),I8(1)),(NU9(2),I9(1))
               DATA NU1(1),NU2(1),NU3(1),NU4(1),NU5(1),NU6(1),NU7(1),NU8(1),NU9(1)
              */9*0/
               DO 8888 I=1,9
               I1(I)=I
               I2(I)=I*        10
               I3(I)=I*       100
               I4(I)=I*      1000
               I5(I)=I*     10000
               I6(I)=I*    100000
               I7(I)=I*   1000000
               I8(I)=I*  10000000
               I9(I)=I*100000000
          8888 CONTINUE
               DO 9999 I=10,63
               I1(I)=0
               I2(I)=0
               I3(I)=0
               I4(I)=0
               I5(I)=0
               I6(I)=0
               I7(I)=0
               I8(I)=0
               I9(I)=0
          9999 CONTINUE
        C
        C      A SZERKESZTETT PROGRAM TETSZOLEGES UTASITASAI
        C
               JR=1
        C
        C      OLVASAS AZ ELSO FORMATUMMAL
        C
             1 CONTINUE
               READ( 10,END=    3,ERR=    4)ISZ1,ISZ2,ISZ3,ISZ4,ISZ5,ISZ6,ISZ7
              *,ISZ  8,ISZ  9,ISZ 10,ISZ 11
               L1=FLD( 0,6,ISZ  1)
               K  1=I1(L1)
               L1=FLD(18,6,ISZ  8)
               L2=FLD(12,6,ISZ  8)
               L3=FLD( 6,6,ISZ  8)
               L4=FLD( 0,6,ISZ  8)
               K  2=I1(L1)+I2(L2)+I3(L3)+I4(L4)
               FLD( 0,18,K  3)=FLD( 6,18,ISZ 11)
        C
        C      A SZERKESZTETT PROGRAM TETSZOLEGES UTASITASAI
        C
               JR=JR+1
               IF(JR-100)1,1,2
        C
        C      OLVASAS A MASODIK FORMATUMMAL
        C
             2 CONTINUE

E3304 03  07-14-8    21.715                SZERKESZTETT PROGRAM                    LABEL ......  PAGE    2
               READ( 10,END=    3,ERR=    4)ISZ1,ISZ2,ISZ3,ISZ4,ISZ5,ISZ6,ISZ7
              *,ISZ  8,ISZ  9,ISZ 10,ISZ 11,ISZ 12,ISZ 13,ISZ 14,ISZ 15,ISZ 16
              *,ISZ 17,ISZ 18,ISZ 19,ISZ 20,ISZ 21,ISZ 22,ISZ 23,ISZ 24,ISZ 25
              *,ISZ 26
               L1=FLD( 6,6,ISZ 10)
               L2=FLD( 0,6,ISZ 10)
               K  1=I1(L1)+I2(L2)
               L1=FLD(18,6,ISZ 10)
               L2=FLD(12,6,ISZ 10)
               K  2=I1(L1)+I2(L2)
               FLD( 0,30,K  3)=FLD( 6,30,ISZ 25)
               FLD(30, 6,K  3)=FLD( 0, 6,ISZ 26)
        C
        C      A SZERKESZTETT PROGRAM TETSZOLEGES UTASITASAI
        C
               JR=JR+1
               GO TO 2
             4 PRINT 7,JR
             7 FORMAT(/2X,I10,24H.REKORDNAL OLVASASI HIBA)
               JR=JR+1
               GO TO 2
             3 STOP
               END

   EDIT DATE   12.31-79  **SR4J**

ELAPSED TIME (SEC       1.15          LINES/MINUTE  4455

  THERE WERE  NO   DIAGNOSTICS IN ABOVE COMPILATION
   27K  WORDS WER. USED FOR THIS COMPILATION
```

9. A szubrutin listája

```
      SUBROUTINE LECTOR(JSZ,JFC,JEND,JERR,JHIBA)
C                              ************************************
C                              * GYORS OLVASO ELJARAS SZERKESZTESE *
C                              ************************************
C
C     JSZ   A SZERKESZTETT PROGRAMOT TARTALMAZO FILE KODJA
C     JFC   AZ INPUT FILE KODJA
C     JEND   A CIMKE ERTEKE, AHOVA FILE VEGE JELZESNEL UGRANI KELL
C     JERR   A CIMKE ERTEKE, AHOVA OLVASASI HIBA ESETEN UGRANI KELL
C     JHIBA   HIBAJELZO PARAMETER
C
C     CF   A FORMATUM HELYE
C
      CHARACTER CF(80)
      CHARACTER C1*5,C2*4,C3*7,C4*2,C5*2,C6*1
      DIMENSION L1(9),L2(9)
      DIMENSION ICF(80)
      EQUIVALENCE (ICF(1),CF(1))
      C1=")ISZ1"
      C2=",ISZ"
      C3="=I1(L1)"
      C4="+I"
      C5="(L"
      C6=")"
      JHIBA=0
      NF=80
      MC=10**10
      MM=0
      MI=1
      IF(JFC) 66,67,66
   66 IF(IFUT-8745177718) 68,21,68
   67 IF(IFJT-8745177718) 70,71,70
C
C           AZ "ELSO" HIVAS PROGRAMJA: A DEKLARALO ES
C           ERTEKADO UTASITASOK SZERKESZTESE
C
   70 IFUT=8745177718
      WRITE(JSZ,51) (I,I,I=2,8)
   51 FORMAT(6X,'DIMENSION NU1(2),I1(63)',3('(NU',I1,'(2),I',I1,'(63)')
     */5X,'*',4('(NU',I1,'(2)',',I',I1,'(63)')/5X,'*,NU9(2),I9(63)')
      WRITE(JSZ,52) (I,I,I=3,9)
   52 FORMAT(6X,"EQUIVALENCE (NU1(2),I1(1)),(NU2(2),I2(1))"/5X,"*",
     *3("(NU",I1,"(2),I",I1,"(1))")/5X,"*",4("(NU",I1,"(2),I",I1,
     *'(1))'))
      WRITE(JSZ,53) (I,I=2,9)
   53 FORMAT(6X,"DATA NL1(1)",8("(NU",I1,"(1)")/5X,"*/9*0/")
      WRITE(JSZ,54)(I,10**(I-1),I=2,9)
   54 FORMAT(6X,"DO 8888 I=1,9"/6X,"I1(I)=I",8(/6X,"I",I1,"(I)=I*",I9))
      WRITE(JSZ,55)(I,I=1,9)
   55 FORMAT(" 8888 CONTINUE"/6X,"DO 9999 I=10,63",9(/6X,"I",I1,"(I)=0")
     *)
      WRITE(JSZ,56)
   56 FORMAT(" 9999 CONTINUE")
      RETURN
   21 CONTINUE
C
C           A PARAMETEREK ELLENORZESE
C
      IF(JFC.GT.99) GO TO 71
      IF(JSZ.LT.1.OR.JSZ.GT.99) GO TO 100
      IF(JEND.LT.1.OR.JEND.GT.99999) GO TO 102
      IF(JERR.LT.1.OR.JERR.GT.99999) GO TO 104
C
```

```
      C              A FORMATUM BEOLVASASA, KIIRASA
      C
            READ 22,CF
      22    FORMAT(80A1)
            PRINT 65,CF
         65 FORMAT(/2X,"A FORMATUM: ",80A1/)
      C
      C              A FORMATUM ELLENORZESE,
      C              A BEOLVASANDO SZAVAK SZAMANAK KISZAMITASA
      C
            NO=0
            NK=0
            I=1
      C
      C     A NYITOZAROJEL MEGKERESESE
      C
      1     IF(CF(I).EQ.1H ) GO TO 23
            IF(CF(I).NE.1H() GO TO 26
            IF(I.EQ.NF) GO TO 28
            I=I+1
            GO TO 2
      23    IF(I.EQ.NF) GO TO 24
            I=I+1
            GO TO 1
      C
      C     FORMATUMELEM KEZDETENEK VIZSGALATA
      C
      2     IF(CF(I).EQ.1H ) GO TO 30
            IF(0.LE.ICF(I).AND.ICF(I).LE.MC) GO TO 3
            IF(CF(I).EQ.1HI) GO TO 4
            IF(CF(I).EQ.1HA) GO TO 74
            GO TO 94
      30    IF(I.EQ.NF) GO TO 24
            I=I+1
            GO TO 2
      C
      C     SZAMMAL KEZDODO FORMATUMELEM
      C
          3 IF(I.EQ.NF) GO TO 28
            DECODE(CF(I),33) M
            MM=10*MM+M
            I=I+1
         19 IF(0.LE.ICF(I).AND.ICF(I).LE.MC) GO TO 3
            IF(CF(I).EQ.1H ) GO TO 20
            IF(CF(I).EQ.1HX) GO TO 15
            IF(CF(I).EQ.1HI) GO TO 16
            IF(CF(I).EQ.1HA) GO TO 73
            GO TO 96
         20 IF(I.EQ.NF) GO TO 28
            I=I+1
            GO TO 19
      C
      C     X SPECIFIKACIO
      C
         15 IF(MM.EQ.0) GO TO 92
            IF(I.EQ.NF) GO TO 28
            I=I+1
            NK=NK+MM
            MM=0
            GO TO 6
      C
      C     I SPECIFIKACIO
      C
```

```
127         16 IF(MM.EQ.0) GO TO 77
12              MI=MM
12              MM=0
13          4   IF(I.EQ.NF) GO TO 28
13              I=I+1
13              IF(CF(I).EQ.1H ) GO TO 4
13              IF(0.LE.ICF(I).AND.ICF(I).LE.MC) GO TO 5
13              GO TO 31
13          5   IF(I.EQ.NF)GO TO 28
13              DECODE(CF(I),33) M
13              IF(M.EQ.0) GO TO 92
13              I=I+1
13              NO=NO+MI
14              NK=NK+MI*M
14              MI=1
14              GO TO 6
14          C
14          C       A SPECIFIKACIO
14          C
14             73 IF(MM.EQ.0) GO TO 77
14              MI=MM
14              MM=0
14             74 IF(I.EQ.NF) GO TO 28
15              I=I+1
15              IF(CF(I).EQ.1H ) GO TO 74
15              IF(0.LE.ICF(I).AND.ICF(I).LE.MC) GO TO 75
15              GO TO 31
15             75 IF(I.EQ.NF) GO TO 28
15              DECODE(CF(I),33) M
156             IF(M.GT.6) GO TO 76
15              IF(M.EQ.0) GO TO 92
15              I=I+1
15              NO=NO+MI
16              NK=NK+MI*M
16              MI=1
16          C
162         C       FORMATUMELEM VEGE UTAN A VESSZO VAGY A ZAROZAROJEL MEGKERESESE
16          C
16          6   IF(CF(I).EQ.1H ) GO TO 34
16              IF(CF(I).EQ.1H,) GO TO 35
16              IF(CF(I).EQ.1H)) GO TO 36
16              GO TO 98
16          34  IF(I.EQ.NF) GO TO 28
17              I=I+1
17              GO TO 6
17             35 IF(I.EQ.NF) GO TO 28
17              I=I+1
17              GO TO 2
17          36  CONTINUE
17              IF(MOD(NK,6))37,38,37
17          37  NSZ=NK/6+1
17              GO TO 39
17          38  NSZ=NK/6
18          39  IF(NSZ.GT.999) GO TO 57
18              IF(NO.GT.999) GO TO 59
18              IF(NO.EQ.0) GO TO 61
18          C
18          C               A READ UTASITAS SZERKESZTESE
18          C
18              IF(NSZ.EQ.1) GO TO 41
18              WRITE(JSZ,40)JFC,JEND,JERR,C1,(C2,I,I=2,NSZ)
18              GO TO 42
18          41  WRITE(JSZ,40)JFC,JEND,JERR,C1
```

```
190          40 FORMAT(6X,"READ(",I3,",END=",I5,",ERR=",I5,A5,6(A4,I1)/(5X,"*",
19             *9(A4,I3)))
19           42    CONTINUE
19           C
19           C              A FORMATUM FELDOLGOZASA
19           C              A VALTOZOK BETOLTESENEK SZERKESZTESE
19           C
19                 IBIT=0
19                 ISZ0=1
19                 J=1
20                 I=1
20                 MM=0
20                 MI=1
20           7     IF(CF(I).EQ.1H ) GO TO 43
20                 I=I+1
20                 GO TO 8
20              43 I=I+1
20                 GO TO 7
20               8 IF(CF(I).EQ.1H ) GO TO 44
20                 IF(CF(I).EQ.1HI) GO TO 10
21                 IF(CF(I).EQ.1HA) GO TO 82
21                 GO TO 9
21              44 I=I+1
21                 GO TO 8
21               9 CONTINUE
21           C
21           C     AZ X SPECIFIKACIO
21           C
21                 DECODE(CF(I),33) M
21                 MM=10*MM+M
22              91 I=I+1
22                 IF(CF(I).EQ.1H ) GO TO 91
22                 IF(0.LE.ICF(I).AND.ICF(I).LE.MC) GO TO 9
22                 IF(CF(I).EQ.1HI) GO TO 17
22                 IF(CF(I).EQ.1HA) GO TO 81
22                 I=I+1
22                 IBIT=IBIT+MM*6
22                 ISZ0=ISZ0+IBIT/36
22                 IBIT=MOD(IBIT,36)
22                 MM=0
23                 GO TO 12
23           C
23           C     AZ I SPECIFIKACIO
23           C
23              17 MI=MM
23                 MM=0
23              10 I=I+1
23                 IF(CF(I).EQ.1H ) GO TO 10
23                 DECODE(CF(I),33) M
23                 I=I+1
24                 DO 13 KK=1,MI
24                 DO 13 K=1,M
24                 KI=M-K+1
24                 L1(KI)=IBIT
24                 L2(KI)=ISZ0
24                 IBIT=IBIT+6
24                 IF(IBIT.EQ.36) ISZ0=ISZ0+1
24                 IF(IBIT.EQ.36) IBIT=0
24              13 CONTINUE
24                 DO 14 K=1,M
25                 WRITE(JSZ,45) K,L1(K),L2(K)
25              45 FORMAT(6X,"L",I1,"=FLD(",I2,",6,ISZ",I3,")")
25              14 CONTINUE
```

```
                      IF(M.EQ.1) GO TO 47
                      WRITE(JSZ,46) J,C3,(C4,K,C5,K,C6,K=2,M)
                      GO TO 11
                   47 WRITE(JSZ,46) J,C3
                   46 FORMAT(6X,"K",I3,A7,7(A2,I1,A2,I1,A1)/5X,"*",A2,I1,A2,I1,A1)
                   11 CONTINUE
                      J=J+1
                   18 CONTINUE
                      MI=1
                      GO TO 12
                C
                C     AZ A SPECIFIKACIO
                C
                   81 MI=MM
                      MM=0
                   82 I=I+1
                      IF(CF(I).EQ.1H ) GO TO 82
                      DECODE(CF(I),33) M
                      MK=M*6
                      I=I+1
                      DO 85 KA=1,MI
                      NU=0
                      IS=ISZO
                      IB=IBIT
                      IF(IBIT+MK-36) 84,85,86
                   84 IBIT=IBIT+MK
                   90 WRITE(JSZ,88) NU,MK,J,IB,MK,IS
                      J=J+1
                      GO TO 83
                   85 IBIT=0
                      ISZO=ISZO+1
                      IF(I3.NE.0) GO TO 90
                      WRITE(JSZ,89) J,IS
                   89 FORMAT(6X,"K",I3,"=ISZ",I3)
                      J=J+1
                      GO TO 83
                   86 MK1=36-IB
                      MK2=MK-MK1
                      WRITE(JSZ,88) NU,MK1,J,IB,MK1,IS
                      IS=IS+1
                      WRITE(JSZ,88) MK1,MK2,J,NU,MK2,IS
                      IBIT=MK2
                      ISZO=ISZO+1
                      J=J+1
                   88 FORMAT(6X,"FLD(",I2,",",I2,",K",I3,")=FLD(",I2,",",I2,",ISZ",I3,")
                     *")
                   83 CONTINUE
                      MI=1
                   12 IF(CF(I).EQ.1H ) GO TO 48
                      IF(CF(I).EQ.1H,) GO TO 49
                      GO TO 50
                   48 I=I+1
                      GO TO 12
                   49 I=I+1
                      GO TO 8
                C
                C              HIBAJELZESEK
                C
                   68 PRINT 69
                   69 FORMAT(/2X,"HIANYZIK A DEKLARALO ES ERTEKADO UTASITASOK SZER
                     *KESZTESENEK HIVASA")
                      JHIBA=1
                      RETURN
```

```
E3304 01  07-14-81    21.708                                                                LABEL LECTOR  PAGE     6

         71 PRINT 72
         72 FORMAT(/2X,"AZ INFUT FILE CODE NEM 1 ES 99 KOZE ESIK")
            JHIBA=1
            RETURN
        100 PRINT 101
        101 FORMAT(/2X,"A SZERKESZTETT PROGRAMOT TARTALMAZO FILE CODE ERTEKE
           *NEM 1 ES 99 KOZE ESIK")
            JHIBA=1
            RETURN
        102 PRINT 103
        103 FORMAT(/2X,"A HARMADIK PARAMETER (JEND) ERTEKE NEM 1 ES 99999 KOZE
           * ESIK")
            JHIBA=1
            RETURN
        104 PRINT 105
        105 FORMAT(/2X,"A NEGYEDIK PARAMETER (JERR) ERTEKE NEM 1 ES 99999 KOZE
           * ESIK")
            JHIBA=1
            RETURN
         24 PRINT 25
         25 FORMAT(/2X,"A FORMATUM HELYE URES")
            JHIBA=1
            RETURN
         26 PRINT 27
         27 FORMAT(/2X,"A FORMATUM NEM NYITOZAROJELLEL KEZDODIK")
            JHIBA=1
            RETURN
         94 PRINT 95
         95 FORMAT(/2X,"NYITOZAROJEL VAGY VESSZO UTAN HIBAS KARAKTER KOVETKEZI
           *K")
            JHIBA=1
            RETURN
         96 PRINT 97
         97 FORMAT(/2X,"A SZUKSEGES HELYEN NEM SZEREPEL X , I VAGY A")
            JHIBA=1
            RETURN
         77 PRINT 79
         79 FORMAT(/2X,"AZ ISMETLESI TENYEZO HELYEN 0 ALL")
            JHIBA=1
            RETURN
         92 PRINT 93
         93 FORMAT(/2X,"0 MEZOSZELESSEG SZEREPEL")
            JHIBA=1
            RETURN
         31 PRINT 32
         32 FORMAT(/2X,"I VAGY A UTAN A MEZOSZELESSEG HELYETT HIBAS KARAKTER
           *KOVETKEZIK")
            JHIBA=1
            RETURN
         76 PRINT 80
         80 FORMAT(/2X,52H"A" FORMATUMBAN 6-NAL NAGYOBB MEZOSZELESSEG SZEREPEL
           *  )
            JHIBA=1
            RETURN
         98 PRINT 99
         99 FORMAT(/2X,"A SZUKSEGES HELYEN NEM SZEREPEL VESSZO VAGY ZAROZAROJE
           *L")
            JHIBA=1
            RETURN
         28 PRINT 29
         29 FORMAT(/2X,"A FORMATUM NEM ZAROZAROJELLEL VEGZODIK")
            JHIBA=1
            RETURN
```

```
E3304 01  07-14-81    21.708                                                                LABEL LECTOR  PAGE     7

         57 PRINT 58
         58 FORMAT(/2X,"A BEOLVASANDO SZAVAK SZAMA NAGYOBB MINT 999")
            JHIBA=1
            RETURN
         59 PRINT 60
         60 FORMAT(/1X,"A BEOLVASANDO ADATOK SZAMA NAGYOBB MINT 999")
            JHIBA=1
            RETURN
         61 PRINT 62
         62 FORMAT(/2X,"A BEOLVASANDO ADATOK SZAMA 0")
            JHIBA=1
            RETURN
         50 CONTINUE
         33 FORMAT(I1)
            IFUT=3745177718
            RETURN
            END
******W      7 MEMO Y EXPANDED. USE $LIMITS OR CORE= OPTION FOR NEXT RUN


  EDIT DATE   12 31-79  **SR4J**

ELAPSED TIME (SEC        3.96          LINES/MINUTE  5974

  THERE WERE     1' DIAGNOSTICS IN ABOVE COMPILATION
   29K  WORDS WER  USED FOR THIS COMPILATION
```

IRODALOM

[1] Ruda M. , Egy széles körben alkalmazható programoptimalizálási módszer , MTA SZTAKI Közlemények , 1978 .

SUMMARY

Creating quick reading procedures in FORTRAN programs / Program description /

The authors present a quick reader - converter procedure, which is applicable in FORTRAN language. The table of the section 7 shows the velocity relation between the usual reader - converter directive / READ / and the quick procedure.

РЕЗЮМЕ

Ускоренная процедура для чтения данных в программах на языке FORTRAN / Описание программы /

Авторы дают быстродействующую процедуру для чтения и преобразования данных , которая может быть использована в языке FORTRAN. Отношение скоростей традиционной и ускоренной процедур показано в таблице седьмого пункта.

A SCRIPTOR szubrutin

Gyors iró eljárások létrehozása FORTRAN programokban

Gál Anna - Ruda Mihály

1. A program célja

A "Gyors olvasó eljárások létrehozása FORTRAN programokban" cimü programleirásban bemutatott eljáráshoz hasonlóan a szokásosnál lényegesen gyorsabb irási és konvertálási eljárás biztositása FORTRAN nyelvü programokban /ld.[1]/.

2. A program formája, a felhasznált gép

FORTRAN szubrutin, hivása:

```
CALL SCRIPTOR (JSZ,JFC,JERR,JHIBA)
```

A felhasznált gép: Honeywell 66/60

3. A feladat részletes leirása, a felhasznált módszerek

3.1 A fent emlitett programleirásban bemutatott LECTOR szubrutinhoz hasonlóan ezt a szubrutint is "szerkesztő" programok hivhatják. Ezek a "szerkesztő" /generáló/ programok gazdaságos müködésü programok előállitását végzik el /ld.[1]/. A szerkesztő program a "szerkesztett" /generált/ programot egy kijelölt file-on helyezi el, ahonnan az forditható és futtatható.

Hivásakor a szubrutin a szerkesztett programba egy néhány utasitásból álló részt ir. Ez a rész megfelel egy olyan irási utasitásnak, amely a $K_1, K_2, \ldots$ változókat megadott formátum szerint karakter formájú rekordba irja ki. /A változók száma legfeljebb 999 lehet./ A formátum I, A vagy X specifikációkat tartalmazhat.

A szubrutin által szerkesztett rész a $K_1, K_2 \ldots$ változókat karaktereikre bontja, majd az egyes karaktereket a megadott formátum szerint helyezi el az ISZ1, ISZ2 ... változókba. /A változók száma itt is legfeljebb 999 lehet./ Ezután az ISZ1, ISZ2 ... változókból álló rekordot binárisan irja ki az output file-ra.

A változók karakterekre bontása és az output rekord összeállitása a következő módon történik:

3.2 Az I specifikációnál csak nem negativ, legfeljebb 9 jegyü értékek szerepelhetnek. A szubrutin a formátum alapján kiszámitja, hogy az egyes számjegyeknek megfelelő karakterek a rekord hányadik szavába kell hogy kerüljenek, és a szó hányadik bitjén kell kezdődniük. Ezután olyan utasitássorozatot szerkeszt, amely a KM változót számjegyeire bontja és az egyes számjegyeknek megfelelő karaktereket elhelyezi az ISZ1, ISZ2 ... változók megfelelő bitjeire. Ezt az FLD függvény /Heneywell FORTRAN/ használata teszi lehetővé. Például az

```
FLD (I,6,ISZK)=J
```

utasitás hatására az ISZK szó I-edik bitjétől I+5-ik bitjéig tartó részére a J kifejezés értékének megfelelő bináris szám utolsó 6 bitje kerül.

Példaként tekintsük azt az esetet, amikor már 23 kiirandó karakter tárolásra került az ISZ1, ISZ2, ISZ3 szóban, és az ISZ4 szó első 30 bitjén. Ha ezek után a soronlévő KM változót I4 formátummal akarjuk kiirni, akkor a következő utasitás sorozatot generálja a szubrutin:

```
L=MOD (KM,10000)
FLD(30,6,ISZ4)=L/1000
L=MOD(L,1000)
FLD(0,6,ISZ5)=L/100
L=MOD(L,100)
FLD(6,6,ISZ5)=L/10
FLD(12,6,ISZ5)=MOD(L,10).
```

3.3 Az A specifikációnál A6 a legnagyobb megengedett formátumelem. A szubrutin a formátum alapján kiszámitja, hogy a KN változó első hány bitje kerül kiirásra, és az output rekord mely szavainak mely bitjeire kell ezeket áthelyezni. Ezután olyan utasitásokat szerkeszt, melyek a KN változó szükséges bitjeit az ISZ1, ISZ2... szavak megfelelő bitjeire helyezi át. Az

```
FLD(I,J,ISZK)=FLD(L,J,KN)
```

utasitás hatására az ISZK szó I-edik bitjétől I+J-1-edik bitjéig tartó részére a KN változó L-edik bitjétől L+J-1-edik bitjéig tartó bitsorozat kerül.

Példaként tekintsük azt az esetet, amikor már 27 kiirandó karakter tárolásra került az ISZ1, ISZ2, ISZ3, ISZ4 szóban és az ISZ5 szó első 18 bitjén. Ebben az esetben a soronlévő KN változó A5 formátum szerinti kiirásához a következő utasitásokat szerkeszti a szubrutin:

```
FLD(18,18,ISZ5)=FLD(0,18,KN)
FLD(0,12,ISZ6)=FLD(18,12,KN)
```

3.4 Az X specifikációhoz szükséges szóközök egy külön változóban /ISP/ vannak elhelyezve, és alkalmanként innen kerül át az output területre /ISZ1,ISZ2,.../.

4. A szubrutin paraméterei

4.1 Bemenő paraméterek

JSZ — A szerkesztett programot tartalmazó file kódja /1 és 99 között lehet/

JFC — Az output file kódja /1 és 99 között lehet/

JERR — A cimke értéke, ahova irási hiba esetén ugrani kell /1 és 99999 között lehet/

4.2 Kimenő paraméter

JHIBA — Hibajelző paraméter
Ha a szubrutin valamilyen hibát talál, akkor nem kezdi el a szerkesztést, a sornyomtatón hibajelzést ir, és visszatér a hivó programba JHIBA=1 értékkel. Hibátlan futás esetén JHIBA=0.

5. A bemenő adatok: Az output formátum

A szubrutin kártyaolvasóról olvassa be az output formátumot.

A formátum tartalmazhat I, A és X specifikációt. Az I specifikációnál 9, az A specifikációnál 6 a legnagyobb megengedett mezőszélesség. A kiirandó adatok száma nem lehet nagyobb mint 999, és az output rekordnak el kell férnie 999 szóban. Lehet használni ismétlési tényezőket /pl. 20A6, 5I4/, de zárójeles ismétlési csoportokat nem. Nem használható "/" jel sem.

A formátumot egy kártyán nyitó- és zárójellel együtt kell megadni. O helyett nem szabad szóközt használni. A formátum szabályos FORTRAN formátum kell hogy legyen.

6. A kimenő adatok

6.1 A szubrutin a sornyomtatón kinyomtatja az output formátumot.

6.2 Az ellenőrzések során talált hibákról a következő hibajelzéseket nyomtatja:

- A bemenő paraméterek ellenőrzése

AZ OUTPUT FILE CODE ÉRTÉKE NEM 1 ÉS 99 KÖZÉ ESIK

A SZERKESZTETT PROGRAMOT TARTALMAZÓ FILE CODE ÉRTÉKE NEM 1 ÉS 99 KÖZÉ ESIK

A HARMADIK PARAMÉTER /JERR/ ÉRTÉKE NEM 1 ÉS 99999 KÖZÉ ESIK

- A formátum ellenőrzése

A FORMATUM HELYE ÜRES

A FORMATUM NEM NYITÓZÁRÓJELLEL KEZDŐDIK

NYITÓZÁRÓJEL VAGY VESSZŐ UTÁN HIBÁS KARAKTER KÖVETKEZIK

A SZÜKSÉGES HELYEN NEM SZEREPEL X, I VAGY A

AZ ISMÉTLÉSI TÉNYEZŐ HELYÉN O ÁLL

O MEZŐ SZÉLESSÉG SZEREPEL

I VAGY A UTÁN A MEZŐSZÉLESSÉG HELYETT HIBÁS KARAKTER KÖVETKEZIK

"A" FORMATUMBAN 6-NÁL NAGYOBB MEZŐSZÉLESSÉG SZEREPEL

A SZÜKSÉGES HELYEN NEM SZEREPEL VESSZŐ VAGY ZÁRÓZÁRÓJEL
/Ez a hibaüzenet jelenik meg akkor is, ha "I" formatumban 9-nél nagyobb mezőszélesség szerepel/

A FORMATUM NEM ZÁRÓZÁRÓJELLEL VÉGZŐDIK

A KIIRANDÓ SZAVAK SZÁMA NAGYOBB MINT 999

A KIIRANDÓ ADATOK SZÁMA NAGYOBB MINT 999

6.3 A szubrutin az első paraméterben /JSZ/ megadott kódú file-ra programrészletet ir. Az egyes hivásoknál irt programrészletek egy-egy irási utasitást helyettesitenek.

7. A felhasználás tapasztalatai

Az alábbi táblázatban bemutatjuk, hogy a hagyományos módon történő irás /és konvertálás/ milyen viszonyban van a SCRIPTOR szubrutin segitségével előállitott iró-konvertáló eljárással. A próbafutásoknál mintegy 150 ezer rekord kiirására került sor, a táblázatban szereplő formátumok szerint. A táblázat utolsó oszlopából látható, hogy az aktuális formátumtól függően változó, de mindenképpen jelentős arányú futásidő csökkenés tapasztalható a SCRIPTOR szubrutinnal létrehozott eljárás javára. Ha az első pontban idézett "Gyors olvasó eljárások létrehozása FORTRAN programokban" cimü leirás táblázatával hasonlitjuk össze az itteni eredményeket, akkor látható, hogy az irási müveletek gyorsulása valamivel kisebb.

F O R M Á T U M	futási idő /óra/		közelitő időarány
	1.	2.	
/i9,i8,4A6,i2,3A2,4i1, 2A1,2i2,A5,3i3,7X,7A1, A6,i8,2A4,2i3,2i5,4X, 2A4,6x,2i5,4A1/	0.4432	0.1471	3.01
/A6,29A1,i3,34i1/	0.4136	0.0762	5.43
/48X,A1,36X,A2,69X,i1/	0.1503	0.0596	2.52
1. hagyományos irás /WRITE/, 2.a SCRIPTOR szubrutin által szerkesztett iró-konvertáló eljárás			

A táblázatban szereplő adatok a központi egység által felhasznált idők. A csatornaidők függetlenek a használt eljárástól.

8. Egy felhasználási példa

Illusztrációképpen bemutatunk egy mintaprogramot, amely mint szerkesztő /generáló/ program felhasználja a SCRIPTOR szubrutint.

Az első lapon a szerkesztő program látható, amely két különböző formátum szerinti iró eljárást állit elő. A második oldalon éppen ez a két formátum látható /ezeket a SCRIPTOR szubrutin irja ki/. Végül a következő oldaltól kezdve maga a szerkesztett program látható, a szerkesztő program, és azon belül a SCRIPTOR szubrutin által létrehozott utasitásokkal.

```
A2637 01  06-23-78   10.806                                                    LABEL ......  PAGE      1

                1       C
                2       C     MINTAPROGRAM SZERKESZTESE
                3     * C
                4             WRITE (10,1)
                5           1 FORMAT("C"/"C    MINTAPROGRAM"/"C"/"C  A MINTAPROGRAM KEZDOSORAI"/
                6            F "C"/6X,"READ 1,K1,K2,K3,K4,K5"/4X,"1 FORMAT(I2,A6,I5,A3,I9)"/
                7            G "C"/"C   IRAS AZ ELSO FORMATUM SZERINT"/"C")
                8             CALL SCRIPTOR (10,10,4,JHIBA)
                9             IF (JHIBA.EQ.1) GO TO 5
               10             WRITE (10,2)
               11           2 FORMAT("C"/"C  IRAS A MASODIK FORMATUM SZERINT"/"C")
               12             CALL SCRIPTOR (10,11,4,JHIBA)
               13             IF (JHIBA.EQ.1) GO TO 5
               14             WRITE(10,7)
               15           7 FORMAT(6X,"STOP"/4X,"4 PRINT 2"/4X,"2 FORMAT(9H IRASHIBA)")
               16             WRITE (10,3)
               17           3 FORMAT(6X,"STOP"/6X,"END")
               18             STOP
               19           5 PRINT 6
               20           6 FORMAT("   HIBA ,   STOP")
               21             STOP
               22             END


SNUMB = A2637, ACTIVITY # = 02, , REPORT CODE = 52, RECORD COUNT = 000004

A FORMATUM:   (I2,A6,I5,A3,I9)

A FORMATUM:   (1X,I2 , 2X , A 6 ,I7,2X ,A3, 2X   ,I  9)


A2637 03  06-23-78   10.817                                                    LABEL ......  PAGE      1

                1       C
                2       C   MINTAPROGRAM
                3       C
                4       C  A MINTAPROGRAM KEZDOSORAI
                5       C
                6             READ 1,K1,K2,K3,K4,K5
                7           1 FORMAT(I2,A6,I5,A3,I9)
                8       C
                9       C   IRAS AZ ELSO FORMATUM SZERINT
               10       C
               11             ISP=17452565520
               12             L=MOD(K  1,       100)
               13             FLD( 0,6,ISZ  1)=L/       10
               14             FLD( 6,6,ISZ  1)=MOD(L,10)
               15             FLD(12,24,ISZ  1)=FLD( 0,24,K  2)
               16             FLD( 0,12,ISZ  2)=FLD(24,12,K  2)
               17             L=MOD(K  3,    100000)
               18             FLD(12,6,ISZ  2)=L/    10000
               19             L=MOD(L,    10000)
               20             FLD(18,6,ISZ  2)=L/     1000
               21             L=MOD(L,     1000)
               22             FLD(24,6,ISZ  2)=L/      100
               23             L=MOD(L,      100)
               24             FLD(30,6,ISZ  2)=L/       10
               25             FLD( 0,6,ISZ  3)=MOD(L,10)
               26             FLD( 6,18,ISZ  3)=FLD( 0,18,K  4)
               27             L=MOD(K  5,10000000000)
               28             FLD(24,6,ISZ  3)=L/100000000
               29             L=MOD(L,100000000)
               30             FLD(30,6,ISZ  3)=L/ 10000000
               31             L=MOD(L, 10000000)
               32             FLD( 0,6,ISZ  4)=L/  1000000
               33             L=MOD(L,  1000000)
               34             FLD( 6,6,ISZ  4)=L/   100000
               35             L=MOD(L,   100000)
               36             FLD(12,6,ISZ  4)=L/    10000
               37             L=MOD(L,    10000)
               38             FLD(18,6,ISZ  4)=L/     1000
               39             L=MOD(L,     1000)
               40             FLD(24,6,ISZ  4)=L/      100
               41             L=MOD(L,      100)
               42             FLD(30,6,ISZ  4)=L/       10
               43             FLD( 0,6,ISZ  5)=MOD(L,10)
               44             WRITE(10,ERR=    4)ISZ1,ISZ2,ISZ3,ISZ4,ISZ5
               45       C
               46       C  IRAS A MASODIK FORMATUM SZERINT
               47       C
               48             ISP=17452565520
               49             FLD( 0, 6,ISZ  1)=ISP
               50             L=MOD(K  1,       100)
               51             FLD( 6,6,ISZ  1)=L/       10
               52             FLD(12,6,ISZ  1)=MOD(L,10)
```

```
      53            FLD(18,12,ISZ  1)=ISP
      54            FLD(30, 6,ISZ  1)=FLD( 0, 6,K  2)
      55            FLD( 0,30,ISZ  2)=FLD( 6,30,K  2)
      56            L=MOD(K  3,  10000000)
      57            FLD(30,6,ISZ  2)=L/  1000000
      58            L=MOD(L,  1000000)
      59            FLD( 0,6,ISZ  3)=L/   100000
      60            L=MOD(L,   100000)
      61            FLD( 6,6,ISZ  3)=L/    10000
      62            L=MOD(L,    10000)
      63            FLD(12,6,ISZ  3)=L/     1000
      64            L=MOD(L,     1000)
      65            FLD(18,6,ISZ  3)=L/      100
      66            L=MOD(L,      100)
      67            FLD(24,6,ISZ  3)=L/       10
      68            FLD(30,6,ISZ  3)=MOD(L,10)
      69            FLD( 0,12,ISZ  4)=ISP
      70            FLD(12,18,ISZ  4)=FLD( 0,18,K  4)
      71            FLD(30, 6,ISZ  4)=ISP
      72            FLD( 0, 6,ISZ  5)=ISP
      73            L=MOD(K  5,1000000000)
      74            FLD( 6,6,ISZ  5)=L/100000000
      75            L=MOD(L,100000000)
      76            FLD(12,6,ISZ  5)=L/ 10000000
      77            L=MOD(L, 10000000)
      78            FLD(18,6,ISZ  5)=L/  1000000
      79            L=MOD(L,  1000000)
      80            FLD(24,6,ISZ  5)=L/   100000
      81            L=MOD(L,   100000)
      82            FLD(30,6,ISZ  5)=L/    10000
      83            L=MOD(L,    10000)
      84            FLD( 0,6,ISZ  6)=L/     1000
      85            L=MOD(L,     1000)
      86            FLD( 6,6,ISZ  6)=L/      100
      87            L=MOD(L,      100)
      88            FLD(12,6,ISZ  6)=L/       10
      89            FLD(18,6,ISZ  6)=MOD(L,10)
      90            WRITE(11,ERR=    4)ISZ1,ISZ2,ISZ3,ISZ4,ISZ5,ISZ6
      91            STOP
      92          4 PRINT 2
      93          2 FORMAT(9H IRASHIBA)
      94            STOP
      95            END
```

9. A szubrutin listája

```
A1622 01  06-22-78   10.778

                      SUBROUTINE SCRIPTOR(JSZ,JFC,JERR,JHIBA)
                C
                C     JSZ    A SZERKESZTETT PROGRAMOT TARTALMAZO FILE KODJA
                C     JFC    AZ OUTPUT FILE KODJA
                C     JERR    A CIMKE ERTEKE, AHOVA IRASI HIBA ESETEN UGRANI KELL
                C     JHIBA    HIBAJELZO PARAMETER
                C
                C     CF    A FORMATUM HELYE
                C
                      CHARACTER CF(80)
                      CHARACTER C1*4
                      DIMENSION ICF(80)
                      EQUIVALENCE (ICF(1),CF(1))
                      C1="ISZ"
                      JHIBA=0
                      NF=80
                      MC=10**10
                      MM=0
                      MI=1
                C
                C              A PARAMETEREK ELLENORZESE
                C
                      IF(JFC.LT.1.OR.JFC.GT.99) GO TO 71
                      IF(JSZ.LT.1.OR.JSZ.GT.99) GO TO 100
                      IF(JERR.LT.1.OR.JERR.GT.99999) GO TO 104
                C
                C              A FORMATUM BEOLVASASA, KIIRASA
                C
                      READ 22,CF
                   22 FORMAT(80A1)
                      PRINT 65,CF
                   65 FORMAT(/2X,"A FORMATUM: ",80A1/)
                C
                C              A FORMATUM ELLENORZESE,
                C              A KIIRANDO SZAVAK SZAMANAK KISZAMITASA
                C
                      NO=0
                      NK=0
                      I=1
                C
                C     A NYITOZAROJEL MEGKERESESE
                C
                    1 IF(CF(I).EQ.1H ) GO TO 23
                      IF(CF(I).NE.1H( ) GO TO 26
                      IF(I.EQ.NF) GO TO 28
                      I=I+1
                      GO TO 2
                   23 IF(I.EQ.NF) GO TO 24
                      I=I+1
                      GO TO 1
                C
                C     FORMATUMELEM KEZDETENEK VIZSGALATA
                C
                    2 IF(CF(I).EQ.1H ) GO TO 30
                      IF(0.LE.ICF(I).AND.ICF(I).LE.MC) GO TO 3
                      IF(CF(I).EQ.1HI) GO TO 4
                      IF(CF(I).EQ.1HA) GO TO 74
                      GO TO 94
                   30 IF(I.EQ.NF)GO TO 24
                      I=I+1
                      GO TO 2
                C
                C     SZAMMAL KEZDODO FORMATUMELEM
                C
                    3 IF(I.EQ.NF) GO TO 28
                      DECODE(CF(I),33) M
                      MM=10*MM+M
                      I=I+1
                   19 IF(0.LE.ICF(I).AND.ICF(I).LE.MC) GO TO 3
                      IF(CF(I).EQ.1H ) GO TO 20
                      IF(CF(I).EQ.1HX) GO TO 15
                      IF(CF(I).EQ.1HI) GO TO 16
                      IF(CF(I).EQ.1HA) GO TO 73
                      GO TO 96
                   20 IF(I.EQ.NF) GO TO 28
                      I=I+1
                      GO TO 19
                C
                C     X SPECIFIKACIO
                C
                   15 IF(MM.EQ.0) GO TO 92
                      IF(I.EQ.NF) GO TO 28
                      I=I+1
                      NK=NK+MM
                      MM=0
                      GO TO 6
                C
                C     I SPECIFIKACIO
                C
                   16 IF(MM.EQ.0) GO TO 77
                      MI=MM
                      MM=0
                    4 IF(I.EQ.NF) GO TO 28
                      I=I+1
                      IF(CF(I).EQ.1H ) GO TO 4
                      IF(0.LE.ICF(I).AND.ICF(I).LE.MC) GO TO 5
                      GO TO 31
                    5 IF(I.EQ.NF) GO TO 28
                      DECODE(CF(I),33) M
                      IF(M.EQ.0) GO TO 92
                      I=I+1
                      NO=NO+MI
                      NK=NK+MI*M
                      MI=1
```

A1622 01   06-22-78   10.778

```
            GO TO 6
      C
    • C     A SPECIFIKACIO
      C
         73 IF(MM.EQ.0) GO TO 77
            MI=MM
            MM=0
         74 IF(I.EQ.NF) GO TO 28
            I=I+1
            IF(CF(I).EQ.1H ) GO TO 74
            IF(0.LE.ICF(I).AND.ICF(I).LE.MC) GO TO 75
            GO TO 31
         75 IF(I.EQ.NF) GO TO 28
            DECODE(CF(I),33) M
            IF(M.GT.6) GO TO 76
            IF(M.EQ.0) GO TO 92
            I=I+1
            NO=NO+MI
            NK=NK+MI*M
            MI=1
      C
      C     FORMATUMELEM VEGE UTAN A VESSZO VAGY A ZAROZAROJEL MEGKERESESE
      C
          6 IF(CF(I).EQ.1H ) GO TO 34
            IF(CF(I).EQ.1H,) GO TO 35
            IF(CF(I).EQ.1H)) GO TO 36
            GO TO 98
         34 IF(I.EQ.NF) GO TO 28
            I=I+1
            GO TO 6
         35 IF(I.EQ.NF) GO TO 28
            I=I+1
            GO TO 2
         36 CONTINUE
            IF(MOD(NK,6))37,38,37
         37 NSZ=NK/6+1
            GO TO 39
         38 NSZ=NK/6
         39 IF(NSZ.GT.999) GO TO 57
            IF(NO.GT.999) GO TO 59
      C
      C             A FORMATUM FELDOLGOZASA
      C
            WRITE(JSZ,40)
         40 FORMAT(6X,"ISP=17452565520")
            IBIT=0
            ISZO=1
            J=1
            I=1
            MM=0
            MI=1
          7 IF(CF(I).EQ.1H ) GO TO 43
            I=I+1
            GO TO 8
         43 I=I+1
            GO TO 7
          8 IF(CF(I).EQ.1H ) GO TO 44
            IF(CF(I).EQ.1HI) GO TO 10
            IF(CF(I).EQ.1HA) GO TO 82
            GO TO 9
         44 I=I+1
            GO TO 8
          9 CONTINUE
      C
      C     AZ X SPECIFIKACIO
      C
            DECODE(CF(I),33)M
            MM=10*MM+M
         91 I=I+1
            IF(CF(I).EQ.1H ) GO TO 91
            IF(0.LE.ICF(I).AND.ICF(I).LE.MC)GO TO 9
            IF(CF(I).EQ.1HI) GO TO 17
            IF(CF(I).EQ.1HA) GO TO 81
            I=I+1
            MKM=MM*6
            MM=0
        112 IF(MKM.EQ.0) GO TO 12
            MK=MKM
            IB=IBIT
            IS=ISZO
            IF(IBIT+MK-36) 106,107,108
        106 IBIT=IBIT+MK
        109 WRITE(JSZ,110) IB,MK,IS
        110 FORMAT(6X,"FLD(",I2,",",I2,",ISZ",I3,")=ISP")
            MKM=MKM-MK
            GO TO 112
        107 IBIT=0
            ISZO=ISZO+1
            IF(IB.NE.0) GO TO 109
            WRITE(JSZ,111) IS
        111 FORMAT(6X,"ISZ",I3,"=ISP")
            MKM=MKM-MK
            GO TO 112
        108 MK=36-IBIT
            GO TO 107
      C
      C     AZ I SPECIFIKACIO
      C
         17 MI=MM
            MM=0
         10 I=I+1
            IF(CF(I).EQ.1H ) GO TO 10
            DECODE(CF(I),33) M
            I=I+1
```

```
A1622 01  06-22-78   10.778

                  IF(M.EQ.1) GO TO 113
                  KIT=10**M
                  DO 18 KK=1,MI
                  WRITE(JSZ,45) J,KIT
               45 FORMAT(6X,"L=MOD(K",I3,",",I10,")")
                  WRITE(JSZ,46) IBIT,ISZO,(KIT/10)
                  IBIT=IBIT+6
                  IF(IBIT.EQ.36) ISZO=ISZO+1
                  IF(IBIT.EQ.36) IBIT=0
                  IF(M.EQ.2) GO TO 11
                  DO 13 K=1,(M-2)
                  WRITE(JSZ,47) 10**(M-K)
               47 FORMAT(6X,"L=MOD(L,",I9,")")
                  WRITE(JSZ,46) IBIT,ISZO,10**(M-K-1)
                  IBIT=IBIT+6
                  IF(IBIT.EQ.36) ISZO=ISZO+1
                  IF(IBIT.EQ.36) IBIT=0
               13 CONTINUE
               11 WRITE(JSZ,119) IBIT,ISZO
              119 FORMAT(6X,"FLD(",I2,",6,ISZ",I3,")=MOD(L,10)")
                  IBIT=IBIT+6
                  IF(IBIT.EQ.36) ISZO=ISZO+1
                  IF(IBIT.EQ.36) IBIT=0
                  J=J+1
               46 FORMAT(6X,"FLD(",I2,",6,ISZ",I3,")=L/",I9)
               18 CONTINUE
                  MI=1
                  GO TO 12
              113 CONTINUE
                  DO 14 KK=1,MI
                  WRITE(JSZ,114) IBIT,ISZO,J
              114 FORMAT(6X,"FLD(",I2,",6,ISZ",I3,")=MOD(K",I3,",10)")
                  IBIT=IBIT+6
                  IF(IBIT.EQ.36) ISZO=ISZO+1
                  IF(IBIT.EQ.36) IBIT=0
                  J=J+1
               14 CONTINUE
                  MI=1
                  GO TO 12
C
C     AZ A SPECIFIKACIO
C
               81 MI=MM
                  MM=0
               82 I=I+1
                  IF(CF(I).EQ.1H ) GO TO 82
                  DECODE(CF(I),33) M
                  MK=M*6
                  I=I+1
                  NU=0
                  DO 83 KA=1,MI
                  IS=ISZO
                  IB=IBIT
                  IF(IBIT+MK-36) 84,85,86
               84 IBIT=IBIT+MK
               90 WRITE(JSZ,88) IB,MK,IS,NU,MK,J
                  J=J+1
                  GO TO 83
               85 IBIT=0
                  ISZO=ISZO+1
                  IF(IB.NE.0) GO TO 90
                  WRITE(JSZ,89) IS,J
               89 FORMAT(6X,"ISZ",I3,"=K",I3)
                  J=J+1
                  GO TO 83
               86 MK1=36-IB
                  MK2=MK-MK1
                  WRITE(JSZ,88) IB,MK1,IS,NU,MK1,J
                  IS=IS+1
                  WRITE(JSZ,88) NU,MK2,IS,MK1,MK2,J
                  IBIT=MK2
                  ISZO=ISZO+1
                  J=J+1
               88 FORMAT(6X,"FLD(",I2,",",I2,",ISZ",I3,")=FLD(",I2,",",I2,",K",I3,")
                 *")
               83 CONTINUE
                  MI=1
               12 IF(CF(I).EQ.1H ) GO TO 48
                  IF(CF(I).EQ.1H,) GO TO 49
C
C              A WRITE UTASITAS SZERKESZTESE
C
                  IF(NSZ.EQ.1) GO TO 116
                  WRITE(JSZ,115) JFC,JERR,(C1,I,I=2,NSZ)
                  GO TO 50
              116 WRITE(JSZ,115) JFC,JERR
              115 FORMAT(6X,"WRITE(",I2,",ERR=",I5,")ISZ1",6(A4,I1)/(5X,"*",9(A4,I3)
                 *))
                  GO TO 50
               48 I=I+1
                  GO TO 12
               49 I=I+1
                  GO TO 3
C
C              HIBAJELZESEK
C
               71 PRINT 72
               72 FORMAT(/2X,"AZ OUTPUT FILE CODE ERTEKE NEM 1 ES 99 KOZE ESIK")
                  JHIBA=1
                  RETURN
              100 PRINT 101
              101 FORMAT(/2X,"A SZERKESZTETT PROGRAMOT TARTALMAZO FILE CODE ERTEKE
                 *NEM 1 ES 99 KOZE ESIK")
                  JHIBA=1
```

```
A1608 01  06-21-78   10.616

                      RETURN
                  104 PRINT 105
                  105 FORMAT(/2X,"A HARMADIK PARAMETER (JERR) ERTEKE NEM 1 ES 99999 KOZE
                     * ESIK")
                      JHIBA=1
                      RETURN
                   24 PRINT 25
                   25 FORMAT(/2X,"A FORMATUM HELYE URES")
                      JHIBA=1
                      RETURN
                   26 PRINT 27
                   27 FORMAT(/2X,"A FORMATUM NEM NYITOZAROJELLEL KEZDODIK")
                      JHIBA=1
                      RETURN
                   94 PRINT 95
                   95 FORMAT(/2X,"NYITOZAROJEL VAGY VESSZO UTAN HIBAS KARAKTER KOVETKEZI
                     *K")
                      JHIBA=1
                      RETURN
                   96 PRINT 97
                   97 FORMAT(/2X,"A SZUKSEGES HELYEN NEM SZEREPEL X , I VAGY A")
                      JHIBA=1
                      RETURN
                   77 PRINT 79
                   79 FORMAT(/2X,"AZ ISMETLESI TENYEZO HELYEN 0 ALL")
                      JHIBA=1
                      RETURN
                   92 PRINT 93
                   93 FORMAT(/2X,"0 MEZOSZELESSEG SZEREPEL")
                      JHIBA=1
                      RETURN
                   31 PRINT 32
                   32 FORMAT(/2X,"I VAGY A UTAN A MEZOSZELESSEG HELYETT HIBAS KARAKTER
                     *KOVETKEZIK")
                      JHIBA=1
                      RETURN
                   76 PRINT 80
                   80 FORMAT(/2X,52H"A" FORMATUMBAN 6-NAL NAGYOBB MEZOSZELESSEG SZEREPEL
                     *  )
                      JHIBA=1
                      RETURN
                   98 PRINT 99
                   99 FORMAT(/2X,"A SZUKSEGES HELYEN NEM SZEREPEL VESSZO VAGY ZAROZAROJE
                     *L")
                      JHIBA=1
                      RETURN
                   28 PRINT 29
                   29 FORMAT(/2X,"A FORMATUM NEM ZAROZAROJELLEL VEGZODIK")
                      JHIBA=1
                      RETURN
                   57 PRINT 58
                   58 FORMAT(/2X,"A KIIRANDO SZAVAK SZAMA NAGYOBB MINT 999")
                      JHIBA=1
                      RETURN
                   59 PRINT 60
                   60 FORMAT(/2X,"A KIIRANDO ADATOK SZAMA NAGYOBB MINT 999")
                      JHIBA=1
                      RETURN
                   50 CONTINUE
                   33 FORMAT(I1)
                      RETURN
                      END
```

I R O D A L O M

[1] Ruda M., Egy széles körben alkalmazható programoptimalizálási módszer,
MTA SZTAKI Közlemények, 1978.

S U M M A R Y

Creating quick writing procedures in
FORTRAN programs
/Program description/

The authors present a quick converter-writer procedure which is applicaple in FORTRAN language. The table of the section 7 shows the velocity relation between the usual converter-writer directive /WRITE/ and the quick procedure.

Р Е З Ю М Е

Ускоренния процедура для записи данных в программах
на языке FORTRAN
/Описание программы/

Авторы дают быстродействующую процедуру для записи преобразования данных, которая может быть использована в языке FORTRAN . Отношение скоростей традиционной и ускоренной процедур показано в таблице седьмого пункта.

A **TANULMÁNYOK** sorozatban 1979-ben megjelentek:

88/1979 Renner G. - Gaál B. - Hermann Gy. - Horváth L. - Várady T.: Szoborszerü felületek tervezése és megmunkálása

89/1979 Ruda Mihály: A SIS77 statisztikai információs rendszer /a felhasznált számitástechnikai eszközök, a rendszer szerkezete és programjai/

90/1979 Bányász Cs. - Keviczky L.: Optimum Insensitivity of the Linear-continuous Transformation

91/1979 Téli iskola /Szentendre/

92/1979 Bolla M. - Csáki P. - Fischer J. - Herodek S. - Hoffman Gy. - Kutas T. - Telegdi L. - Wittmann I.: A balatoni ökoszisztéma modellezése

93/1979 Andor László: Kisgépes adatbázis kezelő rendszer

94/1979 Gertler János: Egy statisztikus szürési eljárás számitógépes folyamatirányitásához

95/1979 Báthory M. - Galló V. - Kovács E. - Mérő L. - Siegler A. - Vajta L.: Festőrobot vezérlésére alkalmas alafelsimerési berendezés

96/1979 Mérő László: Konturkeresés zajos digitalizát képekben

97/1979 Pásztorné - Matavovszky T.: Boole-függvény kezelőrendszer

98/1979 Kecskés Zsuzsa: Három dimenziós tárgyak drótvázának ábrázolása vonalrajzoló grafikus berendezésekkel

99/1979 Ivics József: KGST Riga

100/1979 Téli iskola

1980-ban jelentek meg:

101/1980 Gerencsér László - Hangos Katalin: Diszkrét lineáris sztochasztikus rendszerek önhangoló szabályozása.

102/1980 Pásztorné Varga Katalin: Rekurziv eljárás

103/1980 Gerencsér Piroska - Szép Endre - Zilahy Ferenc Marton Zsolt: Robotmrgfogók adaptivitása I.

104/1980 Knuth Előd - Radó Péter - Tóth Árpád: Az SDLA előzetes ismertetése

105/1980 E. Knuth - P. Radó - Á. Tóth: Preliminary description of SDLA

106/1980 Prékopa András: Sztochasztikus programozási modellek és alkalmazásuk

107/1980 Kelle Péter: Megbizhatósági készletmodellek és alkalmazásuk

108/1980 Almásy Gedeon: Mérlegegyenletek és mérési hibák

109/1980 Békéssy A.-Demetrovics J.-Gyepesi Gy.: Relációs adatbázis logikai szintü vizsgálata funkcionális függőségek szempontjából