

lines close - val karna ar alla  
vakkon dnoletu ar # adding per  
sambott dolgok wert ar eop - h  
a close paracs rukya % file  
egyre

rekordok a file a sornalmal  
keelduek, ha a 4 rekordna  
akarik lapur akko:  
seek (seek, 3)

matru  
Amiko megruhtou a filet automtikusan  
a 0 rekordna lep.

A rekordik veger

A lepuhloso rekord veger of van  
paracs van epp #10 jeldi a h  
hoy a rekordnak van voge, a #13  
redij epp euber jel.  
kinder rekord veger of van a #10, #13.

A pascal rekordna valvasaraval  
alla fitia epp boje a flevak,  
es len ejar ar eop  
ha mat ul tul kovekherst olvasni,  
valvasni, alko lep ar EOF  
etike igar. Aig talat kovekherst  
rekordot addig 0.

(6) (7)

## Kimutatasok

soh file ahidokasoket alud komportok  
suvut kerdem, oschalur.  
Eazallokas formidaban es vrasrak hi  
a cuccokat.  
Iskreseu kimutabas es vrasrak  
Eaz isrepe on lopot  
Fontos hoy mat uiker vrazitunak  
a rekordoket osbeu.  
Fontos hoy naczalik korepseu  
ha valasiljen korepseu  
0 etike van akko amuk ne  
E korepseu a naczalik wert ar  
0-val valas on tar es a pign  
fejvael. Erett ezt mindig kiffelur  
kell.

~~x-es program!~~





### 4. előadás

gyujto : array [1..3, 1..6, 1..6] of longint;

newc  
array of longint, integer

$$gy[n, i, c] := gy[n, i, c] + 1$$

	1	2	3	4	5	össz
1	x					
2	x					
3	x					
4	x					
5	x					
össz						

Ha kölről van nő, megadja a nőket, ha párőről a párokat ill. ezeket összesen.

New kell if-címel dolgozni, mert a szám jó indexnek.

Ahol adatfeldolgozás folyik, ott a nyomatokra kiírni is, nem a képernyőre.

Gyűjtés: hány onlop } ezeket 1-gyel nagyobbra kell válnatani,  
hány lop } mert van összerítés.

életkor: a new-számok kiválasztom a megfelelő jegyeket is a fel-  
dolgozás dátumából kivonva.

if utasítással is meg lehet ezt csinálni → sorok kell lennie → CSÚNYA

éle div 10+1 → elosztás 10-el és korrekciós egyet  
↳ dekor  
0-9 → 1. korosztály  
10-19 → 2. ....  
20-29 → 3. ....

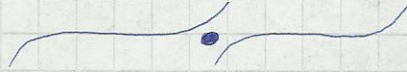
Ha a szám benűl: 0-19 : 1. korcs.  
20-29 : 2. korcs.  
30-39 : 3. korcs.

le kell vonni 20-at és venni kell a születési



abszolútértékét és eloutja a számosságát.

Hf.: a minta alapján elvadás az életkor és sz. vézettség alapján  
Ervált simulatás (nemese botva)  $\rightarrow$  vissza: életkor  
0-60 ill. 60 fött  
10-ésrd



Vektorindexes rendezés:

Alpina	Bela	Ági	Támas	Linda	Cecilia
0	1	2	3	4	5

Sorrendezés: 

2	0	1	5	4	3
---	---	---	---	---	---

Nem az adatok sorrendjét változtatjuk meg, hanem a  
elválasztás sorrendjét. (Az a mutatóban lesz átrendezve.)

Program:

for ciklus 1 elemtől az utolsóig megvizsgálja és elővni azt,  
amelyik a j mutatóban van.

Megvizsgálja. Amelyik kisebb volt, azt rakja be a minimumba.

2760 - től FOR CIKLUST átvesni

289: sor: hogyan kell az ddals 20-ával imi

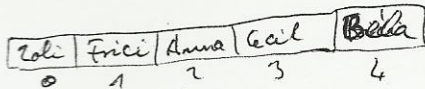
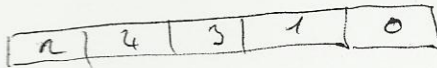


Rendelhetik:

Programozás / előadás / 2003.03.05.

11

- Függőleges felcsatlakozás az adatok
- Logikai: nem csatlakoz, csak megadható, de nem



mintafel. PAS zhadat.dat

→ Helyeztem fenn len a  
K: \Tanard a diazobnal

Keresés egy alomágyban:

Ha egy alomágyban keresek minőségi különbség az hogy le rendezett v. rendezetlen.

Bináris keresés: csak rendezett alomágyban lehet végrehajtani.

Kezeli: nem kell n lépést végrehajtani, hanem  $\log_2 n$  idővéghajtás  
átlagosan!

$\frac{n+1}{2}$  átlagos keresési lépésszám  
 $2^{13} =$  egy lépésből találom meg az adatot.

Időkevesztés történhet kereséssel is lehet árszám a bin. keresésk.

443. szál: Procedure nevelkedés; (Névvizsgálati keresés)

Var  
 rendezett név táblázat  
 el, ve, zo: word; a bináris kereséshez kell.  
 el, ve, zo: word;

Keresés:

```

repeat zo := (el + ve) div 2; gotoxy(22,4);
  if zenev <> neve [mutato[zo]] then
    if zenev < neve [mutato[zo]] then ve := zo - 1 else
      el := zo + 1
until (neve [mutato[zo]] = zenev) or (el > ve);
if el > ve then begin gotoxy(23,2);
  Uptu (Mines igen no, ENTER-vel tovább);

```

(Ingyen szöveg vizsgálati keresés)

Procedure iranyban;

var m: string[4] ← Keresett ~~szöveg~~ irányítású szöveg

db: byte;

kirando: boolean; ← Kirando, Előre vagy se

mindig 0-va kell állítani

előre-utólagos elöl kezdés keresés

3011	2048
6000	2143
6853	848
3522	7091
3300	5125
3301	1650
3301	6015
3301	1915
4826	
4419	
4469	
4191	
4706	
1446	19
8002	

a kirando igaz ha  $db = 4$

kirando :=  $db = 4$ ;

if kirando then begin

adatok;

gatox;

...

} megjelenítés a képernyőn

end;

until eof (sorok); ← ismétlés végéig a megadott adatot.

end;

until sor = '0' ← lényeg a kezdés kód

end;



var allowany : text

alma a fa alatt #13 #10

nyári piros alma

- 1-1 mondat 1 rekord, amelyet 1 betűje egy elem.
- $\text{H}$  karakter tartalmazhat, amelyet az ASCII táblázat tartalmazza.
- #13 #10 minden sort és a új karakter zár
- Nem kell tudni, hogy hol van vége az allowany-uak, mert az eof jel zárja (kódja #26)

Az allowany megnyitása:

assign (allowany, 'A : szovegek.txt');

rewrite  
reset  
append (f. logikai neve)

→ legelső karaktertől kezdve lehet olvasni

→ folytatásra való megnyitás: az eddig meglévő szöveg mögé írni.

úll. lezárása:

close (fnév)

A valóságban úgy néz ki, mintha vektorban lennének, 1-1 elem 1-1 betűtor. A teljes allowany vektorként képezhető el.

Mykódváló:

readln (fnév, input puffer)

string (255 méretűre állítani kell)



írás:

write (file, output puffer)

- A stringben elhelyezkedő karakterek kiíródnak a lemezre.

dehet egyenlőre is olvasni a karaktereket

read (file, input puffer)

"char" típus

Neu kell sortemelni, mert a #13#10 automatikusan sortemel és a sor elején lép.

write (file, output puffer)

char

EOLN

- itt gondoskodni kell arról is, hogy érhessen a #13#10-et is!
- lezárhatjuk close-zal, és akkor automatikusan kirakja az eof jelet!

Program:

PROBITXT.PAS

readln (file, input puffer) → addig olvasza a szöveget, amíg a #13#10-zel nem találkozik.

write (kar) → megjeleníti a leírt karaktert, mert úgy is ráteszi a lemezre, ha nem jeleníti meg. (Csapán azért van, hogy lássuk, mit írunk.)



kar := readkey : beolvas 1 karaktert

A programorba be van építve, hogy az enter tartalmassa a #13-at és #10-et is! Különböző sor elejére menve, és nem mindig sort.

### Abszolút tárciumzés: (redefiniálás)

formája:

var változónev: típus absolute tárcium;

↗ (megnevezés: offset)

Tudni kell, h. milyen szabad tartomány van, mert olyan adhatunk meg dolgokat?

vált 1 : típus 1;

vált 2 : típus 2 absolute vált 1;

vált 1

vált 2

vált 1: word

Valami : array [1..2] of byte absolute vált 1;

sequence: offset cím

\$ a 860 : \$

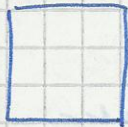
↘ a megnevezés cím + az offset cím  
abszolút címek

látszik a pépenyőn, de a neműnk nem látja, de mi ebbe írunk.

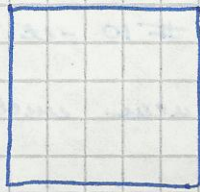
memória terület: 25 x 80 = 2000 karakter fér el, de ehhez 4000 byte-ra van szükség.



25x80



$k[i, j, z]$



$25 \times 80 \times 2 \stackrel{2 \text{ db lehet}}{=} 4000 \text{ byte}$

$k[18, 7, 1]$  = az 18. sor 7. karaktere  $\rightarrow$  mi van ott

$k[18, 7, 2]$  : 1 1 1 1 1 1  
                  ↓                  ↓  
                  karakterek          részletek  
                  villófény (1) vagy szem (0)

0 000 0012  
          ↓          ↓  
          ferek          nőstények  
nem villog

ennek példa : rádióadások 1 tonnával

váltakozó-g-val generáló kódokat és  
keijűt oda

lényegesen hosszággenerálással  $\times 16$ -ot  $\Rightarrow$  a hálók

is mint kap.