

Algoritmusok és adatstruktúrák

7. gyakorlat

1. HETI TÉMAKÖR

Egyszerű rendező algoritmusok. Lineáris és bináris keresés. Indextáblás rendezések és keresések.

2. MEGOLDOTT FELADATOK

Készítsünk adatszerkezeti táblázatot és algoritmust az alábbi feladatok megoldására!

2.1. Korcsolyaverseny

Egy műkorcsolyaversenyen N ($N \leq 20$, input adat) versenyző indul és M ($M \leq 10$, input adat) bíró pontoz. Ismerjük a versenyzők nevét és a bírótól kapott pontjaikat. Minden bíró minden versenyzőnek egy 0-tól 6-ig terjedő pontot (valós szám) ad. Kérjük be az adatokat és készítsük el a verseny eredménylistáját! Az eredménylista a szerzett összpontszám szerint csökkenően rendezett és tartalmazza a versenyzők összes adatát (név, pontok, összpontszám).

Megoldás: Egy versenyző adatait (a könnyebb kezelhetőség érdekében) egy rekordtípusba fogjuk össze. A versenyző nevére egy rövid sztring típust használunk (40 karakteres maximális hosszban). A rendezést *buborékrende*zéssel végezzük. A megoldás bekéri az adatokat (tájékoztató üzenetekkel), kiszámolja az összpontokat, elvégzi a rendezést, végezetül kiírja az eredménylistát a képernyőre.

Konstans

```
/* A versenyzők maximális száma */
NMAX 20
/* A bírók maximális száma */
MMAX 10
/* A nevek maximális hossza */
NEVMAXH 40
```

Típus

```
/* Egy versenyző adatai */
VERSENYZO Rekord
    NEV Sztring[NEVMAXH]
    P Egydimenziós Valós tömb[MMAX]
    OP Valós
/* A versenyzők adatai */
VERSENYZOK Egydimenziós VERSENYZO rekordokból álló tömb[NMAX]
```

Funkció	Azonosító	Típus	Jelleg
A versenyzők száma	N	Egész	I
A bírók száma	M	Egész	I
A versenyzők adatai	V	VERSENYZOK	I, M, O
Csereváltozó	CS	VERSENYZO	M
Segédváltozók	I, J	Egész	M
/* Adatbekérés */			
Ki: "Versenyzők száma (1-" , NMAX, "): "			

```

Be:N
Ki:"Bírók száma (1-",MMAX,"):"
Be:M
for I<-1,N
  Ki:I,". versenyző adatai"
  Ki:"Név:"
  Be:V[I].NEV
  for J<-1,M
    Ki:J,". bírótól kapott pont:"
    Be:V[I].P[J]
/* Összpontszámok meghatározása */
for I<-1,N
  V[I].OP<-0
  for J<-1,M
    V[I].OP<- V[I].OP+V[I].P[J]
/* Rendezés */
for I<-1,N-1
  for J<-N,I+1,-1
    if V[J].OP>V[J-1].OP
      /* A J-dik és J-1-dik versenyző cseréje */
      CS<-V[J]
      V[J]<-V[J-1]
      V[J-1]<-CS
/* Eredmények kiírása */
for I<-1,N
  Ki:I,". ",V[I].NEV," ",
  for J<-1,M
    Ki:V[I].P[J]," ",
  Ki:V[I].OP

```

Megjegyzés: A megoldásban szereplő kiírások után néhol kell soremelést végezni, néhol meg nem. Ha nem szeretnénk sort emelni egy kiírás után (azaz a következő kiírást ugyanabban a sorban szeretnénk folytatni), akkor ezt az utasítás végén álló vesszővel jelezhetjük (mint ahogyan a megfelelő helyeken meg is tettük ezt a megoldásunkban).

3. MEGOLDANDÓ FELADATOK

Módosítsuk az előző fejezet megoldását az alábbiak szerint!

- A pontozók részrehajlását elkerülendő az egyes versenyzők összpontszámába ne vegyük bele az adott versenyző legkisebb és legnagyobb pontjának egy-egy előfordulását! Feltehető, hogy ekkor van legalább három bíró (azaz $M \geq 3$).
- A rendezéshez más rendezési módot (pl. indextáblás beszúró rendezést) használjunk!
- A megoldást szubrutinnal végezzük! Készítsük el a megoldó szubrutin hívását tartalmazó programrészt (pl. főprogram) is (adatszerkezeti táblázat és algoritmus), amelyik bekéri az adatokat, meghívja a megoldó szubrutint, majd kiírja a kapott eredményeket a képernyőre!