

Algoritmusok és adatstruktúrák

(GKNB_MSTM016, GKLK_MSTM016)

Számítógépes vizsga

Ez a fajta vizsga számítógépes kabinetben kerül lebonyolításra. A hallgatóknak számítógépet kell használniuk a (programozási) feladatok megoldására. A vizsgán 3 db feladat lesz, amelyekre adatszerkezeti táblázatot és algoritmust kell készíteni. A szereshető pontszámok rendre 4, 6 és 10, azaz a szereshető maximális pontszám 20 pont. A dolgozat megírására 90 perc áll a hallgatók rendelkezésére.

Érdemjegyek: 0-9: elégtelen, 10-14: elégséges, 15-20: közepes.

A vizsga a Feladatlap-kitöltő program (*Beszámoló.exe*, letöltés: impera.sze.hu) és a *PszKodIDE.exe* program (letöltés: www.sze.hu/~pusztai) segítségével történik. A megoldáshoz segédeszközként csak a *PszKodIDE.exe* program súgója (*PszKodIDE.chm*) használható! A 2. és 3. feladat megoldásához esetlegesen szükséges *Segedszubrutinok.txt* fájl a feladatlap megfelelő link-jével letölthető! Célszerű ezt a fájlt projektként megnyitni (és a megoldást ebben a projektben elkészíteni), hogy a benne található konstans, típus és szubrutin deklarációk rendelkezésre álljanak (majd esetleg más néven elmenteni a projektet)!

A felkészülést és a gyakorlást segítő, ebben a dokumentumban (és a *GepesVizsga_algo_minta.impera* fájlban) mintavizsga feladatokat mutatunk be, ill. bocsátunk a hallgatók rendelkezésére.

A vizsgaminta fájl használata:

- A *Beszámoló.exe* program elindítása. A használathoz nem kell bejelentkezés!
- Egy véletlenszerű feladatlap megnyitása a *Példatár megnyitása* funkcióval történhet. A fájlmeinyitó párbeszédablakban az impera kiterjesztésű fájl kell kiválasztani!
- Amint megjelent a generált feladatlap, akkor elindul az óra, ami a még rendelkezésre álló időt mutatja. Ha ez az idő elfogy, akkor a dolgozat automatikusan beadásra kerül, egyébként meg csak akkor, ha ezt a felhasználó kéri (a *Beadás* menüponttal).
- A feladatlap beadása után megjelenik az értékelés, ami az elért %-os szintet mutatja. Az egyes feladatok helyes válaszai/eredményei/megoldásai mindaddig megnézhetők (pl. beviteli mezők esetén a mező fölé vitt egérkurzorral), amíg a dolgozatot be nem zárjuk. A gépes vizsga feladatainál csak az eredmény értéke nézhető így meg, az azt szolgáltató adatszerkezeti táblázat és algoritmus nem. Ezért az ebben a dokumentumban szereplő feladatoknál megadtunk egy-egy megoldást (az adatszerkezeti táblázat, valamint az algoritmus megadására szolgáló beviteli mezőkben), de megjegyezzük, hogy elképzelhetők más, szintén helyes eredményt adó megoldások is.
- A mintafeladatok megoldásánál csak az eredmény értéke ellenőrződik, az adatszerkezeti táblázat és az algoritmus mezők tartalma csak annyiban, hogy ne legyenek üresek.
- A minta feladatlapok kitöltésekor más szoftverek is használhatók (pl. Acrobat Reader).

A vizsga lebonyolítása:

- A *PszKodIDE.exe* program elindítása. Egyrészt ezt nem fogja bezárni a *Beszámoló.exe* program, másrészt az elindítására általában használt fájlkezelő alkalmazásokat (pl. Intéző, Total Commander) viszont igen.
- A *Beszámoló.exe* program elindítása, majd bejelentkezés. A bejelentkezéshez a Neptun-kód és a jelszó megadása szükséges! A jelszó alapértelmezésben az adott hallgató születési dátumának 8 db számjegye (pl. 20010131).
- Az éppen megírni kívánt beszámoló kiválasztása (ahol is egy adott vizsgán csak egyetlen beszámoló választható ki, az aznap).
- A beszámoló kiválasztása után, az *OK* gomb megnyomásával elindul az adott beszámoló megírása. Az esetlegesen még megnyitott, de nem engedélyezett alkalmazások bezárásra kerülnek. Letöltődik egy véletlenszerű feladatlap és elindul az óra, ami a még rendelkezésre álló időt mutatja. Ha ez az idő elfogy, akkor a dolgozat automatikusan beadásra kerül, egyébként meg csak akkor, ha ezt a felhasználó kéri (a *Beadás* menüponttal).
- A beszámoló beadása után megjelenik az értékelés, ami az elért %-os szintet mutatja. A *Beszámoló.exe* program nem ad részpontot, de az utólagos tanári átnézés során kaphatók részpontok. Előfordulhat

azonban olyan eset is, amikor a *Beszámoló.exe* program maximális pontot ad egy feladat megoldására, de ha a megoldás nem teljesen jó (annak ellenére, hogy jó eredményt kaptunk), akkor a tanári értékelés lepontozhatja a megoldást. A vizsgák eredményei tehát a tanári értékelés után válnak véglegessé, és csak ezután jelennek meg a megfelelő eredménylistán, ill. a megfelelő érdemjegyek a Neptun-ban.

- Egy már megírt beszámoló hallgatói jogosultsággal nem nézhető meg, de a már megírt beszámolókon szerzett összpontszámok megtekinthetők (a *Beszámoló kiválasztása* menüpont legördülő listájában lévő *Korábbi eredmények* menüponttal). A dolgozatok fogadódőben megtekinthetők (a tárgyat oktató kollégáknál).

Megjegyzés: Ha valaki diszlexiás, akkor azt nem a vizsgán kell jeleznie (hiába van a beszámoló kiválasztása ablakban egy erre vonatkozó jelölőnégyzet), hanem a vizsga előtt (pl. e-mail-ben), hogy az ezzel járó adminisztrációt el lehessen végezni. Ilyen esetben ugyanis a hallgató több időt kap a vizsgára.


A továbbiakban néhány minta vizsgafeladatot mutatunk be a számítógépes vizsgára vonatkozóan. Az itt bemutatott (valamint az impera fájlban található) feladatok minták, azaz a vizsgán lehetnek más, hasonló jellegű/nehézségű feladatok is!

Algoritmusok és adatstruktúrák

Tantárgykód: GKNB_MSTM016, GKLB_MSTM016

Beszámoló: Számítógépes vizsga minta


MEGJEGYZÉS A feladatok megoldására a PszKodIDE.exe programot használja! Az adott pontszám eléréséhez a helyes eredmény mellett az eredményt előállító megoldás (adatszerkezeti táblázat és algoritmus) is szükséges! Ezeket az adatokat a vágólap segítségével másolhatja át!

FONTOS A 2-3. feladatok megoldásához töltsse le a [Segedszubrutinok.txt](#)  fájlt, és az adatok generálásához az abban lévő szubrutinokat használja! Ha egy feladat megoldásához saját konstansokat és/vagy típusokat is deklarált, akkor azokat a feladathoz tartozó, az adatszerkezeti táblázat megadására szolgáló mezőbe, a táblázat elé illessze be!

1. Darabszám (4p)

Egy adott intervallum adott tulajdonságú számainak feldolgozása.

1. feladat

 Készítsen adatszerkezeti táblázatot és algoritmust az alábbi feladat megoldására!

Hány olyan egész szám van a [253,687] intervallumban, amelyben a tízes helyi értéken legfeljebb akkora szám áll, mint az egyes helyi értéken?

Eredmény:

Az Ön által készített adatszerkezeti táblázat:

Funkció	Azonosító	Típus	Jelleg
Az eredmény darabszám	DB	Egész	M, O
Az éppen vizsgált szám	I	Egész	M
Az éppen vizsgált szám sztringként	ST	Sztring	M

Az Ön által készített algoritmus:

```
DB<-0
for I<-253,687
  ST<-STR(I)
  if ST[2]<=ST[3]
    DB<-DB+1
Ki:DB
```

4 pont

1. feladat



Készítsen adatszerkezeti táblázatot és algoritmust az alábbi feladat megoldására!

Hány olyan egész szám van az $[1234, 3456]$ intervallumban, amelyre a következő két állítás közül pontosan az egyik igaz?

- a) A szám nem osztható 5-tel.
- b) A szám osztható 7-tel.

MEGJEGYZÉS Oszthatóságon azt értjük, ha az osztó maradék nélkül osztja az osztandót!

Eredmény:

Az Ön által készített adatszerkezeti táblázat:

Funkció	Azonosító	Típus	Jelleg
Az eredmény darabszám	DB	Egész	M, O
Az éppen vizsgált szám	I	Egész	M

Az Ön által készített algoritmus:

```
DB<-0
for I<-1234,3456
  /* Ha a két feltétel különböző értékű, akkor az egyik igaz, */
  /* a másik nem, így éppen azt vizsgáljuk, amit kell */
  /* (nevezetesen a két feltétel között a kizáró vagy műveletet) */
  if (I MOD 5<>0)<>(I MOD 7=0)
    DB<-DB+1
Ki:DB
```

4 pont

2. Összeg, átlag, minimum/maximum generált adatokra (6p)

2. feladat

 Készítsen adatszerkezeti táblázatot és algoritmust az alábbi feladat megoldására!

A NUM_VEKT_GEN szubrutin segítségével generáljon 20 db síkbeli pontot úgy, hogy a pontok X és Y koordinátái a [-50,50] intervallumba essenek! Az X értékeket 2-es, az Y értékeket 3-as Seed értékkel generálja! Határozza meg és adja meg a két, egymástól legtávolabb lévő pont távolságát!

FONTOS Az eredményt két tizedesjegyre kerekítve adja meg!

Eredmény:

Az Ön által készített adatszerkezeti táblázat:


Funkció	Azonosító	Típus	Jelleg
A pontok X koordinátái	X	NUM_VEKTOR	M
A pontok Y koordinátái	Y	NUM_VEKTOR	M
Két pont távolsága	T	Valós	M
A legnagyobb távolság	MAX	Valós	M,O
Segédváltozó	I	Egész	M
Segédváltozó	J	Egész	M

Az Ön által készített algoritmus:

```
/* A pontok (koordinátáinak) generálása */
NUM_VEKT_GEN(X,20,-50,50,2)
NUM_VEKT_GEN(Y,20,-50,50,3)
/* Az eredmény kezdőértéke */
MAX<-0
/* Minden pontpár távolságát kiszámoljuk */
for I<-1,19
  for J<-I+1,20
    /* Az I-dik és J-dik pont távolsága */
    T<-SQRT(SQR(X[I]-X[J])+SQR(Y[I]-Y[J]))
    /* Ha jobb, mint az eddigi legjobb, akkor módosítjuk az eredményt */
    if T>MAX
      MAX<-T
Ki:MAX
```

6 pont

2. feladat

 Készítsen adatszerkezeti táblázatot és algoritmust az alábbi feladat megoldására!

A STR_VEKT_GEN szubrutin segítségével generáljon 30 darab 20 karakter hosszú sztringet 2-es Seed értékkel, majd határozza meg a generált adatokra, hogy hány db számjegytől különböző karaktert tartalmaz az a sztring, amelyikben a legkevesebb számjegytől különböző karakter található!

Eredmény:

Az Ön által készített adatszerkezeti táblázat:

Funkció	Azonosító	Típus	Jelleg
A sztringek	A	STR_VEKTOR	M
Az aktuális sztringhez tartozó db szám	DB	Egész	M
A minimális db szám	MIN	Egész	M, O
Az aktuálisan vizsgált sztring indexe	I	Egész	M
Az aktuálisan vizsgált karakter indexe	J	Egész	M

Az Ön által készített algoritmus:

```
/* A sztringek generálása */
STR_VEKT_GEN(A,30,20,2)
/* A sztringek vizsgálata */
for I<-1,30
  /* Hány db nem számjegy karakter van az I-dik sztringben */
  DB<-0
  for J<-1,LENGTH(A[I])
    if NOT (A[I][J] IN ['0'..'9'])
      DB<-DB+1
  /* Az első sztringet vizsgáljuk? */
  if I=1
    /* Igen -> beállítjuk az eredmény kezdőértékét */
    MIN<-DB
  else
    /* Nem -> jobb, mint az eddigi legjobb? */
    if DB<MIN
      /* Igen -> most már ez legyen az eredmény */
      MIN<-DB
Ki:MIN
```

6 pont

3. Rendezések, keresések, fájlkezelés (10p)

3. feladat

🧐 Készítsen adatszerkezeti táblázatot és algoritmust az alábbi feladat megoldására!

A STR_MTX_GEN szubrutin segítségével generáljon egy 10 soros és 5 oszlopos, 25 karakter hosszú sztringeket tartalmazó mátrixt 1-es Seed értékkel, majd rendezze a mátrix sorait a 3. oszlop szerint növekvő sorrendbe! Adja meg az első 5 db karakterét annak a sztringnek, amelyik az így előállt mátrix 1. sorának 5. oszlopába került!

Eredmény:

Az Ön által készített adatszerkezeti táblázat:


Funkció	Azonosító	Típus	Jelleg
A sztring mátrix	A	STR_MTX	M
Sorok száma	N	Egész	M
Oszlopok száma	M	Egész	M
Csereváltozó a rendezéshez	CS	Sztring	M
Segédváltozó	I	Egész	M
Segédváltozó	J	Egész	M
Segédváltozó	K	Egész	M
Segédváltozó	L	Egész	M


Az Ön által készített algoritmus:

```
/* A mátrix méretei */
N<-10
M<-5
/* A sztringmátrix generálása */
STR_MTX_GEN(A,N,M,25,1)
/* Rendezés */
for I<-1,N-1
  K<-I
  for J<-I+1,N
    /* A 3. oszlop szerint növekvően */
    if A[J,3]<A[K,3]
      K<-J
  if K>I
    /* Az I. és K. sor cseréje */
    for L<-1,M
      CS<-A[I,L]
      A[I,L]<-A[K,L]
      A[K,L]<-CS
/* Az eredmény kiírása */
Ki: COPY(A[1,5],1,5)
```

10 pont

3. feladat

 Készítsen adatszerkezeti táblázatot és algoritmust az alábbi feladat megoldására!

Az **Adatok.txt**  szövegfájl mérési adatokat tartalmaz. Minden egyes sorban két adat található, az első adat a mérés dátuma (éééé.hh.nn alakban), a második adat a mért érték (valós szám), ahol az adatokat pontosan egy db szóköz választja el.

Határozza meg a fájlban található adatokra azt, hogy mennyi a mért értékek átlaga a második negyedében!

FONTOS Az eredményt három tizedesjegyre kerekítve adja meg!

Eredmény:

Az Ön által készített adatszerkezeti táblázat:

Funkció	Azonosító	Típus	Jelleg
Szövegfájl	F	Szövegfájl	M
A fájl egy sora	SOR	Sztring	M
Összeg	OSSZ	Valós	M
Darabszám	DB	Egész	M
Hónap	HO	Egész	M


Az Ön által készített algoritmus:


```
NYIT(F,"ADATOK.TXT","I")
DB<-OSSZ<-0
while NOT FAJLVEGE(F)
  Be F:SOR
  HO<-VAL(COPY(SOR,6,2))
  if HO IN [4..6]
    DB<-DB+1
    OSSZ<-OSSZ+VAL(COPY(SOR,12,LENGTH(SOR)-11))
ZAR(F)
if DB=0
  Ki:"Nincs ilyen adat!"
else
  Ki:"Az eredmény:",VSTR(OSSZ/DB,3)
```

10 pont

Megjegyzés: A megoldás kiírásakor használtuk a **VSTR** standard függvényt, amellyel most a 3 tizedesjegyre való kerekítést végeztük. A *PszKodIDE* program kiíró utasítása ugyanis a valós számokat alapértelmezésben 2 tizedesjegyre kerekítve jeleníti meg. A program beállításai között kérhető normálalakú, ill. más tizedesjegyre kerekített megjelenítés is, amely egy 'másik útja' az eredmény megfelelő pontosságú 'megkapásának'.

3. feladat

 Készítsen adatszerkezeti táblázatot és algoritmust az alábbi feladat megoldására!

Az `Adatok.dat`  típusos fájl mérési adatokat tartalmaz. Az adatfájl rekordjának típusdeklarációja (amit a megoldásban használni kell):

```
ADATREK Rekord
  DATUM Sztring[10] /* A mérés dátuma éééé.hh.nn alakban */
  ADAT Valós        /* A mért adat */
```

Határozza meg a fájlban található adatokra azt, hogy mekkora a legnagyobb, júliusban mért adat!

Eredmény:

Az Ön által készített adatszerkezeti táblázat:

Funkció	Azonosító	Típus	Jelleg
A típusos fájl	F	ADATREK adatokból álló típusos fájl	M
Egy rekord	A	ADATREK	M
Az eredmény	MAX	Valós	M, O
Darabszám	DB	Egész	M

Az Ön által készített algoritmus:

```
NYIT(F, "ADATOK.DAT", "I")
DB<-0
while NOT FAJLVEGE(F)
  Be F:A
  if COPY(A.DATUM, 6, 2) = "07"
    DB<-DB+1
    if DB=1
      MAX<-A.ADAT
    else
      if A.ADAT>MAX
        MAX<-A.ADAT
ZAR(F)
if DB=0
  Ki:"Nincs ilyen adat!"
else
  Ki:"Az eredmény:", MAX
```

10 pont

Megjegyzés: A megoldás adatszerkezeti táblázatában a típusos fájl típusának szövegét lerövidítve adtuk meg, hogy az adatszerkezeti táblázat teljes szélességében látható legyen. A *PszKodIDE* programmal 'előállított' típusleírás szó szerint a következő lesz: ADATREK típusú adatokból álló típusos fájl. Természetesen a megoldásban használhatunk saját típust is a típusos fájl típusának deklarálására, mert ekkor elegendő a deklarált típus kiválasztása a fájlváltozó (F) típusának megadásakor (csakúgy, mint ahogyan azt meg kell tennünk az ADATREK típus és az A változó deklarálásakor).

Pusztai Pál