

Algoritmusok és adatstruktúrák

(GKNB_MSTM016, GKLB_MSTM016)

Táv-vizsga

Ez a dokumentum egyrészt a kialakult vírushelyzet miatt megváltozott vizsgáztatási szabályokat ismerteti, másrészt, a hallgatók felkészülését segítő, mintavizsga feladatokat tartalmaz.

A vizsga nem az egyetem területén lesz, a hallgatók saját, internettel rendelkező számítógépeken vizsgázhatnak az egyetem erre vonatkozó [általános szabályai](#) mellett.

A vizsga három részből áll. Az első két rész elméleti kérdéseket és feladatokat tartalmaz, a harmadik rész pedig gyakorlati (programozási) feladatokat. Az egyes részekre maximálisan 10, 12, ill. 90 perc áll a hallgatók rendelkezésére. A vizsgán először az elméleti részeket kell megoldani, azután a gyakorlati részt!

A szereshető pontszámok: 10, 10, ill. 20 pont. A sikeres vizsgához minden részt legalább 50%-os szinten teljesíteni kell! Például 5, 5, 10 pont már sikeres vizsgát jelent, de 4, 8, 15 nem jelent sikeres vizsgát. Egy ismételt vizsgán nem kell mindhárom részt újra teljesíteni, elegendő azt (ill. azokat), amelyikből még nincs meg az 50%-os szint! Egy rész akkor is újra írható, ha belőle már megvan az 50%-os szint (pl. jobb jegy érdekében), ekkor az új pontszám felülírja a régit (azaz ronthatunk is)!

Érdemjegyek: 0-19: elégtelen, 20-24: elégséges, 25-29: közepes, 30-34: jó, 35-40: jeles.

A vizsga a Feladatlap-kitöltő program (*Beszámoló.exe*, letöltés: impera.sze.hu) és a *PszKodIDE.exe* program (letöltés: www.sze.hu/~pusztai) segítségével történik. Az első két (elméleti) rész megírásakor csak a *Beszámoló.exe* program használható (azaz nem használhatók segédeszközök)! A 3. rész feladatainak megoldása közben segédeszközként a *PszKodIDE.exe* program súgója (*PszKodIDE.chm*) használható! Itt a 2-3. feladatok megoldásához szükséges *Segedszubrutinok.txt* fájl a feladatlap megfelelő link-jével letölthető! Célszerű ezt a fájl 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)!

Gyakorláshoz (mintavizsga) feladatok (korlátozott számban) a következő fájlokban találhatóak:

- 1. rész: TavVizsga_1_resz_minta.impera
- 2. rész: TavVizsga_2_resz_minta.impera
- 3. rész: TavVizsga_3_resz_minta.impera

A vizsgaminták használata:

- A *Beszámoló.exe* program elindítása. A használathoz nem kell bejelentkezés!
- Egy véletlenszerű feladatlap megnyitása a *Példatár megnyitása* funkcióval történhet. A fájlmegegyítő párbeszédablakban azt az impera kiterjesztésű fájl kell kiválasztani, amelyikből feladatlapot szeretnénk generálni!
- 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 vizsga 3. részében csak az eredmény értéke nézhető így meg, az azt szolgáló 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 3. részhez tartozó 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 vizsga előtt be kell lépni a megfelelő Google Meet szobába (<https://meet.google.com/qcr-habt-bos>), ahol is megtörténik a vizsgázó személyazonosságának ellenőrzése ([általános előírások](#)). A támogatott böngészők: *Chrome, Firefox, Opera*.
- A kapcsolattartásra használt böngészőn kívül zárjunk be minden más alkalmazást! A vizsga megkezdésével ugyanis az összes megnyitott, nem megengedett alkalmazást (pl. Acrobat Reader, Fájlkészlő, Total Commander) a *Beszámoló.exe* program bezár!
- 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 (1. rész, 2. rész, 3. rész sorrendben). A vizsgákon mind a három vizsgarész megírható, de egy ismételt vizsgán elegendő csak azt/azokat, amelyikből még nincs meg az 50%-os szint!
- A vizsga 3. részének megírása előtt célszerű a *PszKodIDE.exe* programot elindítani! Egyrészt ez nem fog bezáródni, másrészt az elindítására általában használt fájlkezelő alkalmazások viszont igen, így nehogy azért ne tudjunk a tudásunkról számot adni a vizsgán, mert csak fájlkezelővel tudunk elindítani egy programot! Célszerű egy, a programot elindító parancsikont létrehozni az asztalra, amivel gond nélkül elindítható a program akkor is, ha az összes nem megengedett alkalmazás bezáródik!
- A beszámoló (vizsgarész) 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 (azaz egy adott vizsgarész) beadása után megjelenik az értékelés, ami az elért %-os szintet mutatja. Az 1. és 2. részben (ahol 1 vagy 2 pontosak a feladatok) egy feladat megoldása vagy jó, vagy nem, részpont nem kapható. A 3. részben (ahol 4, 6 és 10 pontosak a feladatok) a *Beszámoló.exe* program nem ad részpontot, de ez utólagos tanári átnézés során részpontok kaphatók. A vizsgák eredményei tehát a tanári értékelés után válnak véglegessé, és csak ezután jelennek meg a megfelelő eredménylistán, Neptun-ban.
- Mivel a vizsga elején elvégzett ellenőrzés még nem garantálja azt, hogy valóban az a személy jelentkezik be és vizsgázik le (aki belépett a Google Meet szobába), ezért egy második ellenőrzésre is sor kerül! A vizsga végén, az utóljára megírt rész eredményéről tájékoztató ablakot a hallgató ne zárja be (akármilyen eredményt is lát rajta)! Az ablak bezárása csak a vizsgáztató szóbeli engedélyével történhet, miközben a hallgató prezentálással megosztja számítógépének képernyőképét! Ha a személyazonosság ellenőrzése okostelefonnal történt (pl. kamera nélküli számítógép használata esetén), akkor a szóbeli engedélyezés is azon keresztül történik! Mivel ez a második ellenőrzés minden vizsgán minden hallgatónál pontosan egyszer történik meg, ezért egy adott hallgatónak egy adott vizsgán az összes megírt vizsgarészt ugyanazon a számítógépen kell megírnia! Ellenkező esetben a vizsgát érvénytelennek tekintjük, a vizsgán megírt mindegyik vizsgarész értékelése 0 pont!
- 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 fogadói időben 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 vizsga egyes részeihez igazodóan. Az itt bemutatott (valamint az impera fájlokban található) feladatok minták, azaz a vizsgán lehetnek más, hasonló jellegű/nehézségű feladatok is!

Algoritmusok és adatstruktúrák

Tantárgykód: GKNB_MSTM016, GKLB_MSTM016

Beszámoló: Táv-vizsga 1. rész minta

1. feladat


 Döntse el az alábbi állításokról, hogy igazak vagy hamisak!

A balról jobbra szabály akkor él, ha van zárójel a kifejezésben.

Egy tömb indextípusa lehet bármilyen egyszerű típus.

1 pont

2. feladat


 Rakja a műveleteket a végrehajtásuk szerinti sorrendbe! Kezdje a legmagasabb prioritásával!

1. tartalmazás (IN)
2. szorzás és osztás (*, /)
3. összeadás és kivonás (+, -)
4. egész osztás maradéka (MOD, %)
5. előjel (+, -)



1 pont

3. feladat

 Az alábbiak közül melyik fogalom „található meg” az alábbi kifejezésben?

$(A[I] \cdot B[J]) \text{ IN } ['A' \dots 'Z']$

- függvényhívás
- halmaz
- operandus
- felesleges zárójel
- A felsorolt lehetőségek egyike sem.


1 pont

Algoritmusok és adatstruktúrák

Tantárgykód: GKNB_MSTM016, GKLB_MSTM016

Beszámoló: Táv-vizsga 1. rész minta

4. feladat

 Csoportosítsa az alábbi függvényeket! Egy függvény több csoportba is tarozhat.

 LENGTH | SIN | RANDOM | COPY | STR

Paramétere tetszőleges valós szám lehet:


Eredménye mindig egész szám:

Eredménye sztring típusú:

Egyik fenti kategóriába sem sorolható:

1 pont

5. feladat


 Értékelje ki az alábbi kifejezést, majd adja meg az eredmény értékét! Hiba esetén egy # karaktert írjon a mezőbe!

`SQRT(-ABS(-9))`

Az eredmény értéke:

2 pont

6. feladat

 Döntse el az alábbi állításról, hogy a tagmondatok tartalma igaz-e, illetve van-e összefüggés a két tagmondat között! Jelölje be a helyes válasz betűjelét!

- A:** Az első tagmondat igaz, a második is igaz, továbbá ok és okozati összefüggés van köztük.
- B:** Az első tagmondat igaz, a második is igaz, de ok és okozati összefüggés nincs köztük.
- C:** Az első tagmondat igaz, a második hamis.
- D:** Az első tagmondat hamis, a második igaz.
- E:** Mindkét tagmondat önmagában is hamis.

Az előltesztelés ciklus eshet végtelen ciklusba is, mert a ciklusmag a ciklust vezérlő feltétel kiértékelése előtt hajtódik végre.

A B C D E


2 pont

Algoritmusok és adatstruktúrák

Tantárgykód: GKNB_MSTM016, GKLB_MSTM016

Beszámoló: Táv-vizsga 1. rész minta

7. feladat

 Mit ír ki eredményül az alábbi algoritmus? Az I változó egész típusú.

Konstans

```
A = (2, 2, 3, 3, 5, 5, 5, 6, 7, 8)
```

```
I ← 1
```

```
while (I < 10) AND (A[I] <= 8)
```

```
    I ← I + 1
```

```
Ki: I
```

A megjelenő eredmény:


2 pont

Algoritmusok és adatstruktúrák

Tantárgykód: GKNB_MSTM016, GKLB_MSTM016

Beszámoló: Táv-vizsga 2. rész minta

1. feladat


 Döntse el az alábbi állításokról, hogy igazak vagy hamisak!

A dinamikus változó mutatóváltozó segítségével hozható létre.

A veremkezelésnek alapvetően két művelete van, a ráhelyezés és a levétel, amelyek mindig sikeresen végrehajthatók.

1 pont

2. feladat

 Csoportosítsa az alábbi segédsubrutinokat! Egy szubrutin több csoportba is tartozhat.

 NYIT | [ZAR](#) | [FAJLVEGE](#) | [FELSZABADIT](#) | [HELYFOGLAL](#)

A szubrutin eljárásként definiált:


Van fájlváltozó paramétere:

Egynél több paramétere van:

Egyik fenti kategóriába sem sorolható:

1 pont

3. feladat

 Döntse el az alábbi állításról, hogy a tagmondatok tartalma igaz-e, illetve van-e összefüggés a két tagmondat között! Jelölje be a helyes válasz betűjelét!

A: Az első tagmondat igaz, a második is igaz, továbbá ok és okozati összefüggés van köztük.

B: Az első tagmondat igaz, a második is igaz, de ok és okozati összefüggés nincs köztük.

C: Az első tagmondat igaz, a második hamis.

D: Az első tagmondat hamis, a második igaz.

E: Mindkét tagmondat önmagában is hamis.

Az érték szerinti paraméterátadásnál az aktuális paraméter csak konstans lehet, mert a paraméter címe adódik át.

A **B** **C** **D** **E**

2 pont

Algoritmusok és adatstruktúrák

Tantárgykód: GKNB_MSTM016, GKLB_MSTM016

Beszámoló: Táv-vizsga 2. rész minta

4. feladat

🔍 Mit ír ki eredményül a TESZT szubrutin? Hiba esetén egy # karaktert írjon a mezőbe!

```
Konstans
  NMAX 5
Típus
  TOMB Egydimenziós egészekből álló tömb[NMAX]
```

A VALAMI szubrutin az adatszerkezeti táblázatával együtt:

Funkció	Azonosító	Típus	Jelleg
Adatok	A	TOMB	I
Adatok száma	DB	egész	I
Eredmény	ER	egész	M, O
Segédváltozó	I	egész	M

```
VALAMI (A, DB, ER)
ER ← 0
for I ← 1, DB, 2
  ER ← ER + A[I]
```

A TESZT szubrutin az adatszerkezeti táblázatával együtt:

Funkció	Azonosító	Típus	Jelleg
Adatok	A	TOMB	M
Eredmény	ER	egész	O
Segédváltozó	I	egész	M

```
TESZT()
for I ← 1, 5
  A[I] ← 2 * I
VALAMI (A, 5, -ER)
Ki: ER
```

A megjelenő eredmény:


2 pont

Algoritmusok és adatstruktúrák

Tantárgykód: GKNB_MSTM016, GKLB_MSTM016

Beszámoló: Táv-vizsga 2. rész minta

5. feladat

 Mit ír ki eredményül a TESZT szubrutin? Hiba esetén egy # karaktert írjon a mezőbe!

A VALAMI szubrutin az adatszerkezeti táblázatával együtt:

Funkció	Azonosító	Típus	Jelleg
Az adat	ST	sztring	I
Eredmény	ER	sztring	M,O
Ciklusváltozó	I	egész	M

```
VALAMI(ST)
ER ← ST[1]
for I ← 2,LENGTH(ST)
  if ST[I] > ER
    ER ← ST[I]
return ER
```

A TESZT szubrutin az adatszerkezeti táblázatával együtt:

Funkció	Azonosító	Típus	Jelleg
Eredmény	ER	sztring	M

```
TESZT()
ER ← "ribizli"
ER ← VALAMI(ER) + ER
Ki: ER
```

A megjelenő eredmény:

FONTOS Ha az eredmény karakter vagy sztring, akkor azt aposztrófok, illetve macskakörök nélkül adja meg!


2 pont

Algoritmusok és adatstruktúrák

Tantárgykód: GKNB_MSTM016, GKLB_MSTM016

Beszámoló: Táv-vizsga 2. rész minta

6. feladat

 Mit ír ki eredményül a TESZT szubrutin? Hiba esetén egy # karaktert írjon a mezőbe!

Az ADATOK.DTA típusos fájlban az alábbi egész számok találhatóak: 2 4 1 7 5 8 3 6 9 0

A TESZT szubrutin az adatszerkezeti táblázatával együtt:

Funkció	Azonosító	Típus	Jelleg
A fájlváltozó	F	egész számokból álló típusos fájl	M
Segédváltozó	A	egész	M

```
TESZT()  
NYIT(F, "ADATOK.DTA", "I+")  
POZICIONAL(F, 3)  
Be F: A  
A ← A + 1  
Ki F: A  
ZAR(F)  
Ki: A
```

A megjelenő eredmény:


2 pont

Algoritmusok és adatstruktúrák

Tantárgykód: GKNB_MSTM016, GKLB_MSTM016

Beszámoló: Táv-vizsga 3. rész 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! Részpont nem kapható!

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 80 darab egész számot a [0,5000] intervallumban 2-es Seed értékkel, majd határozza meg a generált adatokra az 1000-nél nagyobb értékű adatok átlagát!

FONTOS Az eredményt egy tizedesjegyre kerekítve adja meg!

Eredmény:

Az Ön által készített adatszerkezeti táblázat:


Funkció	Azonosító	Típus	Jelleg
A generált számok	A	NUM_VEKTOR	M
A megfelelő elemek darabszáma	DB	Egész	M
A megfelelő elemek összege	OSSZ	Egész	M
Az aktuálisan vizsgált szám	I	Egész	M

Az Ön által készített algoritmus:

```
/* A számok generálása */
NUM_VEKT_GEN(A,80,0,5000,2)
/* A számok feldolgozása */
DB<-OSSZ<-0
for I<-1,80
  if A[I]>1000
    DB<-DB+1
    OSSZ<-OSSZ+A[I]
/* 1.0-val szorzunk, hogy az eredmény valós szám legyen */
Ki:1.0*OSSZ/DB
```

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 negyedévben!

FONTOS Az eredményt egy 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:", OSSZ/DB
```

10 pont

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	Ecéssz	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).