

INFORMATIKA

PASCAL NYELV



A 2016-OS ÉRETTSÉGI KÖVETELMÉNYEINEK
MEGFELELŐ OKTATÁSI SEGÉDANYAG

© 2016 PRESSTERN SOLUTIONS

Tartalomjegyzék

Algoritmusok - pszeudókód.....	1
Abszolút érték	1
Hányados ismételt kivonással	1
Legnagyobb közös osztó.....	1
Páros számok szűrése	2
Palindrom számok	2
Orosz szorzás.....	2
Minimum keresés.....	3
Maximum keresés	3
Eukleidész algoritmus.....	4
Prímszámok.....	4
Fibonacci-számok	5
Háromszög	5
Fordított szám.....	6
Törzstényezők	7
Prímszámvizsgálat	7
Konverzió – Számrendszer átalakítás	9
Gyors hatványozás	10
Szekvenciális (lineáris) keresés	10
Megszámlálás	11
Minimum- és maximumkiválasztás	11
A Maximum helye	11
Kiválogatás	11
Szétválogatás	12
Sorozat halmazzá alakítása.....	13
Sorozatok keresztmetszete.....	13
Sorozatok egyesítése	14
Sorozatok összefésülése	14
Párok sorszáma egy sorozatban.....	15
Arány	16
Teljes négyzet	16
Osztályátlagok szétválasztása	16
Bűvös négyzet.....	17
Polinom értéke adott pontban.....	18
Polinomok összege.....	19
Polinomok szorzata.....	19
Buborékrendezés (Bubble-sort)	20

Egyszerű felcseréléses rendezés.....	21
Válogatásos rendezés	21
Minimum/maximum kiválasztásra épülő rendezés.....	21
Beszűrő rendezés.....	22
Leszámláló rendezés	22
Összefésülésen alapuló rendezés	23
Gyorsrendezés (QuickSort).....	23
Szavak sorrendjének megfordítása	24
Faktoriális.....	25
Számjegyösszeg	25
k elemű részhalmazok.....	25
Konverzió.....	26
Az $\{1, 2, \dots, n\}$ halmaz minden részhalmaza.....	26
Kamatos kamatok kiírása.....	26
Általános backtracking.....	27
Általános rekurzív backtracking	28
Elhelyezni 8 királynőt a sakktablán.....	28
Zárójelek.....	29
Játékok dobozva való elhelyezésének kiírása.....	30
X pénzösszeg kifizetése n bankjegy segítségével	31
X Összeg kifizetése, minimum számú bankjeggyel.....	31
Általános Divide Et Impera.....	32
Szorzat (DivImp).....	33
Minimumszámolás (DivImp).....	33
Hatványozás (DivImp).....	34
Bináris keresés (DivImp)	34
Általános mohó (Greedy) algoritmus	35
Összeg (Greedy).....	35
Hátizsák probléma (Greedy).....	35
Összegkifizetés legkevesebb számú bankjeggyel (Greedy)	36

A Pascal nyelv elemei..... 37

Azonosítók.....	37
Alapértelmezett egyszerű típusok.....	38
Változók.....	38
Sorszámozott típusú változók deklarálása.....	38
Nem sorszámozott típusú változók deklarálása	39
Konstansok	40
Egész típusú konstansok	40
Valós típusú konstansok	40
Karakter és karakterlánc típusú konstansok.....	41

Logikai konstansok	41
Kezdőértékkel rendelkező változók	41
Adatok beolvasása és kiírása	42
Szabványos függvények és eljárások	42
Matematikai függvények és eljárások	42
Sorszámozott típusú adatokra vonatkozó függvények és eljárások.....	44
Egyszerű programok készítése	46
Kifejezések.....	46
Bitszintű műveletek	47
Relációs műveletek	47
Pascal nyelvű programozás.....	48
Döntések.....	48
Nagyobbik szám kiírása	48
Csökkenő sorrend létrehozása	48
Római szám felismerése	49
Római szám felismerése (case utasítással)	50
Kis- nagybetű vagy más karakter.....	51
Jegyértékelés.....	51
Hónapok napjainak száma	52
Ciklusok.....	53
Legnagyobb közös osztó (Eukleidész algoritmus)	53
Prímszámvizsgálat	53
Szám számjegyeinek száma	54
Törzstényezőkre való bontás	55
N szám összege.....	55
Fibonacci sorozat n-edik eleme	56
Egydimenziós tömbök	57
Átlagos jegynél kisebbek kiírása	57
Számsorozat kiírása fordítva	57
Sorozatból való teljes négyzetek összege.....	58
Karakterláncban felmerülő betűk száma	59
Két polinom szorzata	60
Kétdimenziós tömbök	61
Két félévi átlagok.....	61
Négyzetes mátrix szimmetriája	62
Mátrix páros sorai elemeinek középárayosa.....	63
Karakterláncok.....	64
Relációs műveletek karakterláncokkal.....	65

Karakterlánc kezelő függvények	65
Karakterláncokat feldolgozó eljárások	66
Szóköz törlése egy karakterláncból	67
Karakterláncban folytatott szöcsere	67
Halmazok	68
Műveletek halmazokkal	69
Halmazokra vonatkozó relációk és vizsgálatok	69
Egy halmaz elemeinek a kiírása	70
Karakterlánc betűinek kiírása ábécésorrendben	70
Erasztoténész algoritmus (prím számok)	70
Rekord vagy bejegyzés típusok	71
Alkalmazottak adatainak nyilvántartása	72
Kereső algoritmusok	74
Lineáris keresés	74
Bináris keresés	74
Rendező algoritmusok	76
Buborekrendezés	76
Minimumkiválasztásos rendezés	77
Beszúrásos rendezés	77
Összefésülések	78
Összefésülés strázsza (ütköző) nélkül	78
Összefésülés strázsával (ütközővel)	80

Rekurzivitás **82**

Faktoriális kiszámítása	82
Fibonacci sorozat n-edik eleme	82
Két szám legnagyobb közös osztója	83
Hatványozás	84
10-es számrendszerből való alakítás	85
Első n páratlan szám összege	85
Természetes szám számjegyeinek összege	86
Számsorozat negatív elemeinek száma	87
Szám előfordulása egy sorozatban	88
Vektor páros elemeinek összege	89
Sorozat legnagyobb prímszáma	90
N hosszúságú a és b karakterű sorozatok	91
Természetes szám partíciói	92
Labirintus-feladatok	93

„Oszd meg és uralkodj” – „Divide Et Impera” **97**

Sorozat legnagyobb száma.....	97
N valós szám szorzata	98
Keresés számsorozatban.....	98
Hanoi tornyok.....	100
QuickSort.....	100
MergeSort.....	102

Kombinatorikus feladatok 104

Permutációk	104
Variációk	105
Kombinációk	106

Listák..... 108

Lista tartalmának kiírása.....	108
Listákat kezelő eljárások és függvények.....	108
Lista k.-adik elemének kiírása.....	109
Lista bővítése rendezetten	109
Két rendezett lista összefésülése.....	112
Verem	115

Szöveges állományok 117

Állomány megnyitása.....	117
Állomány bezárása	118
Írás állományba.....	118
Olvásás állományból	119
Szöveges állományok feldolgozása	120
Hibakezelés	120
A sor végének az ellenőrzése.....	121
Az állomány végének az ellenőrzése	121

Gráfok 123

Irányított és nem irányított gráfok.....	123
Definíciók	123
Dijkstra-algoritmus.....	125
Floyd-algoritmus.....	128
Minimális értékű Hamilton-kör bejárása.....	130
Szélességi bejárás.....	131
Keresési fa	133

Algoritmusok - pszeudókód

Abszolút érték

Határozzuk meg és írjuk ki adott valós szám abszolút értékét!

Algoritmus Abszolút_érték(x, mod):

```
Ha  $x \geq 0$  akkor                                {bemeneti adat:  $x$ , kimeneti adat:  $mod$ }  
     $mod \leftarrow x$   
különben  
     $mod \leftarrow -x$   
vége(ha)  
Vége(algoritmus)
```

Hányados ismételt kivonással

Számítsuk ki két természetes szám egész hányadosát ismételt kivonásokkal!

Algoritmus Osztas($a, b, hányados$):

```
     $hányados \leftarrow 0$                                 {bemeneti adatok:  $a, b$ , kimeneti adat:  $hányados$ }  
Amíg  $a \geq b$  végezd el:  
     $hányados \leftarrow hányados + 1$   
     $a \leftarrow a - b$   
vége(amíg)  
Vége(algoritmus)
```

Legnagyobb közös osztó

Számítsuk ki két természetes szám legnagyobb közös osztóját!

Algoritmus Eukleidész($a, b, loko$):

```
Ismételd                                {bemeneti adatok:  $a, b$ , kimeneti adat:  $loko$ }  
     $r \leftarrow maradék[a/b]$                 {kiszámítjuk az aktuális maradékot}  
     $a \leftarrow b$                             {az osztandót felülírjuk az osztóval}  
     $b \leftarrow r$                             {az osztót felülírjuk a maradékkal}  
    ameddig  $r = 0$                             {amikor a maradék 0, véget ér az algoritmus}  
     $loko \leftarrow a$                             {loko egyenlő az utolsó osztó értékével}  
Vége(algoritmus)
```

Páros számok szűrése

Számoljuk meg n beolvasott szám közül a páros számokat!

{bemenet n és a számok, kimenet: db , a páros számok száma}

Algoritmus Páros(n,db):

$db \leftarrow 0$

Minden $i=1,n$ **végezd el:**

Be: szám

Ha szám *páros akkor*

$db \leftarrow db + 1$

vége(ha)

vége(minden)

Vége(algoritmus)

Palindrom számok

Döntsük el egy adott számról, hogy palindromszám-e vagy sem!

Algoritmus Palindrom(szám,válasz):

másolat \leftarrow szám

{bemeneti adat: *szám*, kimeneti adat: *válasz*}

újszám $\leftarrow 0$

Amíg szám > 0 **végezd el:**

számjegy \leftarrow maradék[szám/10]

újszám \leftarrow újszám*10 + számjegy

szám \leftarrow [szám/10]

vége(amíg)

válasz \leftarrow újszám = másolat

{ha *újszám = másolat*, akkor *válasz* értéke *igaz*}

{ha *újszám \neq másolat*, akkor *válasz* értéke *hamis*}

Vége(algoritmus)

Orosz szorzás

Legyen $a, b \in \mathbb{N}^*$. Számítsuk ki a és b szorzatát!

Algoritmus Orosz_szorzás(a,b,p):

$x \leftarrow a$

{bemeneti adatok: a, b }

$y \leftarrow b$

{kimeneti adat: p }

$p \leftarrow 0$

Amíg $x > 0$ **végezd el:**

$\{xy + p = ab (*)\}$

Ha x *páratlan* **akkor**

$p \leftarrow p + y$

vége(ha)

$x \leftarrow \lfloor x/2 \rfloor$

$y \leftarrow y + y$

vége(amíg)

Vége(algoritmus)

Minimum keresés

Határozzuk meg egy n elemű sorozat minimumát!

Algoritmus Minimum(n, a, \min):

$\min \leftarrow a_1$

Minden $i=2, n$ **végezd el:**

Ha $a_i < \min$ **akkor**

$\min \leftarrow a_i$

vége(ha)

vége(minden)

Vége(algoritmus)

Maximum keresés

Írjuk ki három, páronként különböző valós szám közül a legnagyobbat!

Algoritmus Maximum(a, b, c):

Ha $(a > b)$ **és** $(a > c)$ **akkor**

{bemeneti adatok: a, b, c }

Ki: 'A legnagyobb: ', a

vége(ha)

Ha $(b > c)$ **és** $(b > a)$ **akkor**

Ki: 'A legnagyobb: ', b

vége(ha)

Ha $(c > a)$ **és** $(c > b)$ **akkor**

Ki: 'A legnagyobb: ', c

vége(ha)

Vége(algoritmus)

A Pascal nyelv elemei

Azonosítók

A Pascalban használt azonosítók szerepe hasonlít az algoritmusoknál bemutatottakéhoz. Az azonosítók az angol ábécé kis- és nagybetűiből, számokból és az aláhúzásjelből állhatnak, azzal a kikötéssel, hogy első karakterük nem lehet számjegy, és hosszuk nem haladhatja meg a 63 karaktert.

Ajánlatos az úgynevezett „beszédesebb” azonosítók használata, amelyek az általuk képviselt jelentésre utalnak. Például előnyös, ha egy minimumot kereső program neve minimum, vagy ha egy hatványt megőrző változó neve hatvány, és nem x.

Azonosítóként sosem használhatunk kulcsszavakat. Példák azonosítókra:

- minimum
- x
- _x1
- matrix
- _1a
- alfa
- sebesseg

Mivel a Pascal-fordító nem tesz különbséget a kis- és a nagybetűk között, a következő azonosítók egyenértékűek az előbbiekkel:

- Minimum
- X
- _X1
- MatriX
- _1A
- ALFA
- SEBESSEG

Hibás azonosítók:

- 1x { nem kezdődhet számmal }
- a-b { nem tartalmazhat - karaktert }
- m i n { nem lehetnek benne szóközők }
- összead { nem tartalmazhat ékezetes betűt }

Alapértelmezett egyszerű típusok

A Pascal nyelv a következő egyszerű típusokat használja:

- sorszámozott típusok: egész, logikai, karakter, felsorolt, résztartomány;
- nem sorszámozott típusok: valós, karakterlánc.

Változók

A deklarációban a változó nevét és típusát kettőspont választja el egymástól, a típust pontosvessző követi. Ha több változóhoz is ugyanazt a típust szeretnénk rendelni, akkor ezek nevét vesszővel kell elválasztanunk egymástól.

Például:

```
Var  
a, b: Byte;  
c: Char;
```

Sorszámozott típusú változók deklarációja

- egész típusú változók deklarációja:

```
Var  
a: Byte;  
osszeg: Integer;  
i, j: Word;  
x: Shortint;  
eredmeny: Longint;
```

- logikai típusú változók deklarációja:

```
Var  
b: Boolean;
```

- karakter típusú változók deklarációja:

```
Var  
c: Char;  
betu: Char;
```

- felsorolt típusú változók deklarációja:

```
Var  
nyelvek: (angol, nemet, roman, magyar, latin);
```

```
napok: (hetfo, kedd, szerda, csutortok, pentek, szombat, vasarnap);
```

Látható, hogy ékezetes betűk nem szerepelhetnek a felsorolásban. Ezeknek az azonosítóknak a deklarációs részben egyedieknek kell lenniük. Nem lehet például német nevű azonosítónk, és az angol nem szerepelhet egynél több felsorolásban.

Helytelenek tehát a következő deklarációk:

```
Var
```

```
nyelvek: (angol, nemet, roman, magyar, latin);  
kedvenc: (angol, nemet);
```

- résztartomány típusú változók deklarációja:

```
Var
```

```
szamok: 0..9;  
kisbetuk: 'a'..'z';  
nagybetuk: 'A'..'Z';  
munkanapok: hetfo..pentek;
```

Hibásak például a következő deklarációk:

```
Var
```

```
napok = hetfo..vasarnap;  
kisbetuk: Char;  
x, kisbetuk: Byte;
```

Nem sorszámozott típusú változók deklarációja

- valós típusú változók deklarációja:

```
Var
```

```
a, b: Real;
```

- illetve:

```
{SN+}
```

```
Var
```

```
a, b: Double;  
osszeg: Extended;
```

- karakterlánc típusú változók deklarációja:

```
Var
```

```
nev: string;
```

A példában a csnev maximális hossza 20 karakter, a knev-é pedig 15 karakter.

Pascal nyelvű programozás

Döntések

Nagyobbik szám kiírása

Olvassunk be két egész számot, majd megfelelő szöveg kíséretében írjuk ki a nagyobbikat!

```
Program maximum;
Var
  szam1, szam2, max: Integer;
Begin
  WriteLn;
  Write('Az első szám: ');
  ReadLn(szam1);
  Write('A második szám: ');
  ReadLn(szam2);
  If szam1 > szam2 Then
    max := szam1
  Else
    max := szam2;
  WriteLn('A nagyobbik szám: ', max);
  ReadLn;
End.
```

Csökkenő sorrend létrehozása

Olvassunk be három egész számot, majd írjuk ki ezeket csökkenő sorrendben!

```
Program csokkeno_sorrend;
Var
  szam1, szam2, szam3: Integer;
Begin
  WriteLn;
  Write('Az első szám: ');
  ReadLn(szam1);
  Write('A második szám: ');
  ReadLn(szam2);
  Write('A harmadik szám: ');
```

```

ReadLn(szam3);
Write('A szamok csokkeno sorrendben: ');
If szam1 > szam2 Then
  If szam1 > szam3 Then
    If szam2 > szam3 Then
      WriteLn(szam1, ', ', szam2, ', ', szam3)
    Else
      WriteLn(szam1, ', ', szam3, ', ', szam2)
  Else
    WriteLn(szam3, ', ', szam1, ', ', szam2)
Else
  If szam2 > szam3 Then
    If szam1 > szam3 Then
      WriteLn(szam2, ', ', szam1, ', ', szam3)
    Else
      WriteLn(szam2, ', ', szam3, ', ', szam1)
  Else
    WriteLn(szam3, ', ', szam2, ', ', szam1);
ReadLn;
End.

```

Római szám felismerése

Olvassunk be egy karaktert, amely egy római számjegynek felel meg, majd írjuk ki a neki megfelelő arab számot! Ha a karakter nem római számjegy, írjunk ki hibaüzenetet!

```

Program romai;
Var
  c: Char;
Begin
  WriteLn;
  Write('Kerem a romai szamjegyet: ');
  ReadLn(c);
  Write('Az arab szam: ');
  If c = 'I' Then
    Write(1)
  Else
    If c = 'V' Then
      Write(5)
    Else

```

```

If c = 'X' Then
  Write(10)
Else
  If c = 'L' Then
    Write(50)
  Else
    If c = 'C' Then
      Write(100)
    Else
      If c = 'D' Then
        Write(500)
      Else
        If c = 'M' Then
          Write(1000)
        Else
          Write('Nem romai szamjegy. ');
ReadLn;
End.

```

Római szám felismerése (case utasítással)

```

Program romai_case;
Var
  c: Char;
Begin
  WriteLn;
  Write('Kerem a romai szamjegyet: ');
  ReadLn(c);
  Write('Az arab szam: ');
  Case c Of
    'I' : Write(1);
    'V' : Write(5);
    'X' : Write(10);
    'L' : Write(50);
    'C' : Write(100);
    'D' : Write(500);
    'M' : Write(1000);
  Else
    Write('Nem romai szamjegy. ');
  End;
  ReadLn;
End.

```

„Oszd meg és uralkodj” – „Divide Et Impera”

Sorozat legnagyobb száma

```
Program maximumszamias_oszd_meg_es_uralkodjal;
Var
  x: array[1..20] of Integer;
  i, n: Byte;

Function maximum(bal, jobb: Byte): Integer;
Var
  max1, max2: Integer;
  kozepe: Byte;
Begin
  If bal = jobb Then
    maximum := x[bal]
  Else
    Begin
      kozepe := (bal + jobb) div 2;
      max1 := maximum(bal, kozepe);
      max2 := maximum(kozepe + 1, jobb);
      If max1 < max2 Then
        maximum := max2
      Else
        maximum := max1;
    End;
  End;

Begin
  Write('n = ');
  ReadLn(n);
  Write('A számok: ');
  For i := 1 To n Do
    Read(x[i]);
  ReadLn;
  WriteLn('Maximum: ', maximum(1,n));
  ReadLn;
End.
```


N valós szám szorzata

Számítsuk ki n valós szám szorzatát oszd meg és uralkodj módszerrel!

```
Program szorzas_oszd_meg_es_uralkodjal;
```

```
Var
```

```
  x: array[1..20] of Integer;
```

```
  i, n: Byte;
```

```
Function szorzas(bal, jobb: Byte): Integer;
```

```
Var
```

```
  sz1, sz2: Integer;
```

```
  kozepe: Byte;
```

```
Begin
```

```
  If bal = jobb Then
```

```
    szorzas := x[bal]
```

```
  Else
```

```
    Begin
```

```
      kozepe := (bal + jobb) div 2;
```

```
      sz1 := szorzas(bal, kozepe);
```

```
      sz2 := szorzas(kozepe + 1, jobb);
```

```
      szorzas := sz1 * sz2;
```

```
    End;
```

```
End;
```

```
Begin
```

```
  Write('n = ');
```

```
  ReadLn(n);
```

```
  Write('A számok: ');
```

```
  For i := 1 To n Do
```

```
    Read(x[i]);
```

```
  ReadLn;
```

```
  WriteLn('A számok szorzata: ', szorzas(1,n));
```

```
  ReadLn;
```

```
End.
```

Keresés számsorozatban

Olvassunk be egy rendezett számsorozatot, és keressünk meg benne egy beolvasott számot! Ha megtaláltuk írjuk ki a pozícióját. Alkalmazzuk a bináris keresés módszerét!

```

Program binaris_kereses;
Var
  x: array[1..20] of Integer;
  ker: Integer;
  i, n: Byte;

Procedure bin_kereses(bal, jobb: Byte);
Var
  kozepe: Byte;
Begin
  If bal > jobb Then
    Begin
      WriteLn(ker, ' nincs az adott sorozatban');
      Exit;
    End;
  kozepe := (bal + jobb) div 2;
  If ker = x[kozepe] Then
    WriteLn(ker, ' a(z) ', kozepe, 'h. talalhato')
  Else
    If ker < x[kozepe] Then
      bin_kereses(bal, kozepe - 1)
    Else
      bin_kereses(kozepe + 1, jobb);
End;

Begin
  Write('n = ');
  ReadLn(n);
  Write('A novekvő sorozat: ');
  For i:=1 To n Do
    Read(x[i]);
  Write('Keresett szám: ');
  ReadLn(ker);
  bin_kereses(1,n);
  ReadLn;
End.

```