

Tartalomjegyzék

Algoritmusok - pszeudókód	1
Abszolút érték	1
Hányados ismételt kivonással.....	1
Legnagyobb közös osztó.....	1
Páros számok szűrése	2
Palindrom számok	2
Orosz szorzás	3
Minimum keresés	3
Maximum keresés	4
Eukleidész algoritmus.....	4
Prímszámok	5
Fibonacci-számok	6
Háromszög.....	6
Fordított szám.....	7
Törzstényezők	8
Prímszámvizsgálat	9
Konverzió – Számrendszer átalakítás.....	10
Gyors hatványozás	12
Szekvenciális (lineáris) keresés.....	12
Megszámolás.....	13
Minimum- és maximumkiválasztás.....	13
A Maximum helye.....	13
Kiválogatás.....	14
Szétválogatás.....	14
Sorozat halmazzá alakítása	15
Sorozatok keresztmetszete.....	15
Sorozatok egyesítése	16
Sorozatok összefésülése	17
Párok sorszáma egy sorozatban	18
Arány.....	18
Teljes négyzet.....	19
Osztályátlagok szétválasztása	19
Bűvös négyzet	21
Polinom értéke adott pontban	22
Polinomok összege.....	22

Polinomok szorzata	23
Buborékrendezés (Bubble-sort)	23
Egyszerű felcseréléses rendezés	24
Válogatásos rendezés	25
Minimum/maximum kiválasztásra épülő rendezés	25
Beszűrő rendezés	26
Leszámláló rendezés	26
Összefésülésen alapuló rendezés	27
Gyorsrendezés (QuickSort)	28
Szavak sorrendjének megfordítása	28
Faktoriális	29
Számjegyösszeg	29
k elemű részhalmazok	30
Konverzió	30
Az $\{1, 2, \dots, n\}$ halmaz minden részhalmaza	31
Kamatos kamatok kiírása	31
Általános backtracking	32
Általános rekurzív backtracking	33
Elhelyezni 8 királynőt a sakktáblán	33
Zárójelek	34
Játékok dobozva való elhelyezésének kiírása	35
X pénzösszeg kifizetése n bankjegy segítségével	36
X Összeg kifizetése, minimum számú bankjeggyel	37
Általános Divide Et Impera	38
Szorzat (DivImp)	38
Minimumszámolás (DivImp)	39
Hatványozás (DivImp)	40
Bináris keresés (DivImp)	40
Általános mohó (Greedy) algoritmus	41
Összeg (Greedy)	41
Hátizsák probléma (Greedy)	42
Összegkifizetés legkevesebb számú bankjeggyel (Greedy)	43
A Pascal nyelv elemei	44
Azonosítók	44
Alapértelmezett egyszerű típusok	45
Változók	45
Sorszámozott típusú változók deklarálása	46
Nem sorszámozott típusú változók deklarálása	47
Konstansok	48
Egész típusú konstansok	48

Valós típusú konstansok	48
Karakter és karakterlánc típusú konstansok.....	49
Logikai konstansok	49
Kezdőértékkel rendelkező változók	50
Adatok beolvasása és kiírása.....	50
Szabványos függvények és eljárások.....	51
Matematikai függvények és eljárások	51
Sorszámozott típusú adatokra vonatkozó függvények és eljárások... 54	
Egyszerű programok készítése.....	56
Kifejezések	57
Bitszintű műveletek	57
Relációs műveletek	58
Pascal nyelvű programozás.....	59
Döntések	59
Nagyobbik szám kiírása	59
Csökkenő sorrend létrehozása	59
Római szám felismerése.....	60
Római szám felismerése (case utasítással)	61
Kis- nagybetű vagy más karakter	62
Jegyértékelés.....	63
Hónapok napjainak száma.....	63
Ciklusok	64
Legnagyobb közös osztó (Eukleidész algoritmus).....	64
Prímszámvizsgálat	65
Szám számjegyeinek száma.....	66
Törzstényezőkre való bontás.....	67
N szám összege.....	68
Fibonacci sorozat n-edik eleme.....	68
Egydimenziós tömbök	69
Átlagos jegynél kisebbek kiírása.....	69
Számsorozat kiírása fordítva	70
Sorozatból való teljes négyzetek összege.....	71
Karakterláncban felmerülő betűk száma	72
Két polinom szorzata.....	73
Kétdimenziós tömbök.....	74
Két félevení átlagok.....	74
Négyzetes mátrix szimmetriája.....	75
Mátrix páros sorai elemeinek középarányosa.....	76
Karakterláncok	78

Relációs műveletek karakterláncokkal	79
Karakterlánc kezelő függvények	79
Karakterláncokat feldolgozó eljárások	80
Szóköz törlése egy karakterláncból.....	81
Karakterláncban folytatott szócsere	82
Halmazok	83
Műveletek halmazokkal.....	83
Halmazokra vonatkozó relációk és vizsgálatok	84
Egy halmaz elemeinek a kiírása.....	85
Karakterlánc betűinek kiírása ábécésorrendben.....	85
Erasztoténész algoritmus (prímszámok)	86
Rekord vagy bejegyzés típusok.....	87
Alkalmazottak adatainak nyilvántartása	88
Kereső algoritmusok	89
Lineáris keresés	89
Bináris keresés.....	90
Rendező algoritmusok.....	92
Buborékrendezés	92
Minimumkiválasztásos rendezés.....	93
Beszűrős rendezés	94
Összefésülések.....	95
Összefésülés strázsa (ütköző) nélkül	95
Összefésülés strázásával (ütközővel)	96
Rekurzivitás	98
Faktoriális kiszámítása.....	98
Fibonacci sorozat n-edik eleme	99
Két szám legnagyobb közös osztója.....	99
Hatványozás	100
10-es számrendszerből való alakítás	101
Első n páratlan szám összege	102
Természetes szám számjegyeinek összege	103
Számsorozat negatív elemeinek száma	103
Szám előfordulása egy sorozatban	104
Vektor páros elemeinek összege.....	106
Sorozat legnagyobb prímszáma.....	107
N hosszúságú a és b karakterű sorozatok.....	108
Természetes szám partíciói	109
Labirintus-feladatok.....	111

„Oszd meg és uralkodj” – „Divide Et Impera” ...115

Sorozat legnagyobb száma.....	115
N valós szám szorzata.....	116
Keresés számsorozatban.....	117
Hanoi tornyok.....	118
QuickSort.....	119
MergeSort.....	121
Kombinatorikus feladatok.....	123
Permutációk.....	123
Variációk.....	124
Kombinációk.....	126
Listák.....	128
Lista tartalmának kiírása.....	128
Listákat kezelő eljárások és függvények.....	129
Lista k.-adik elemének kiírása.....	129
Lista bővítése rendezetten.....	130
Két rendezett lista összefésülése.....	133
Verem.....	136
Szöveges állományok.....	138
Állomány megnyitása.....	139
Állomány bezárása.....	140
Írás állományba.....	140
Olvadás állományból.....	141
Szöveges állományok feldolgozása.....	142
Hibakezelés.....	142
A sor végének az ellenőrzése.....	143
Az állomány végének az ellenőrzése.....	144
Gráfok.....	146
Irányított és nem irányított gráfok.....	146
Definíciók.....	146
Dijkstra-algoritmus.....	149
Floyd-algoritmus.....	152
Minimális értékű Hamilton-kör bejárása.....	154
Szélességi bejárás.....	156
Keresési fa.....	158

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($a, b, luko$):

Ismételd {bemeneti adatok: a, b , kimeneti adat: $luko$ }
 $r \leftarrow \text{maradék}[a/b]$ {kiszámítjuk az aktuális maradékot}
 $a \leftarrow b$ {az osztandót felülírjuk az osztóval}
 $b \leftarrow r$ {az osztót felülírjuk a maradékkal}
amedig $r = 0$ {amikor a maradék 0, véget ér az algoritmus}
 $luko \leftarrow a$ {luko 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:**

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 „beszédes” 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 deklarálása

- egész típusú változók deklarálása:

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

- logikai típusú változók deklarálása:

```
Var  
b: Boolean;
```

- karakter típusú változók deklarálása:

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

- felsorolt típusú változók deklarálása:

```
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 deklarálása:

```
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 deklarálása

- valós típusú változók deklarálása:

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

- illetve:

```
{SN+}  
Var  
a, b: Double;  
osszeg: Extended;
```

- karakterlánc típusú változók deklarálása:

```
Var  
nev: string;
```

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

```
Var  
csnev: string[20];  
knev: string[15];
```

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 számok csökkenő 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 hibüzenetet!

```

Program romai;
Var
  c: Char;

```

```

Begin
  WriteLn;
  Write('Kerem a romai számjegyet: ');
  ReadLn(c);
  Write('Az arab szám: ');
  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 számjegy. ');
  ReadLn;
End.

```

Római szám felismerése (case utasítással)

```

Program romai_case;
Var
  c: Char;
Begin
  WriteLn;
  Write('Kerem a romai számjegyet: ');

```

```

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.

```

Kis- nagybetű vagy más karakter

Olvassunk be egy karaktert, és döntsük el, hogy nagybetű, kisbetű, szám vagy más karakter!

```

Program vizsga1;
Var
  karakter: Char;
Begin
  WriteLn;
  Write('Kerek egy karaktert: ');
  ReadLn(karakter);
  Write('A beolvasott karakter ');
  Case karakter Of
    'A'..'Z': Write('nagybetu. ');
    'a'..'z': Write('kisbetu. ');
    '0'..'9': Write('szam. ');
  Else
    Write('egyeb. ');
  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 szamok: ');
```

```
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 szamok: ');
```