INFORMATIKA

PASCAL NYELV

A 2016-OS ÉRETTSÉGI KÖVETELMÉNYEINEK MEGFELELŐ OKTATÁSI SEGÉDANYAG

A PUSKÁZÁS CSALÁSNAK MINŐSÜL.
A PUSKÁK VIZSGÁN TÖRTÉNŐ HASZNÁLATÁT NEM AJÁNLJUK!

Tartalomjegyzék

Algoritmusok - pszeudókód 1–42
Abszolút érték 1
Hányados ismételt kivonással.....1
Legnagyobb közös osztó 1–2
Páros számok szűrése 2
Palindrom számok 2
Orosz szorzás.....3
Minimum keresés 3
Maximum keresés.....3–4
Eukleidész algoritmusa.....4
Prímszámok.....4–5
Fibonacci-számok.....5–6
Háromszög 6–7
Fordított szám.....7–8
Törzstényezők 8
Prímszámvizsgálat.....8–10
Konverzió – Számrendszer átalakítás.....10–11
Gyors hatványozás 11
Szekvenciális (lineáris) keresés 12
Megszámlálás 12
Minimum- és maximumkiválasztás.....12
A Maximum helye.....13
Kiválogatás.....13
Szétválogatás.....13–14
Sorozat halmazzá alakítása.....14–15
Sorozatok keresztmetszete.....15–16
Sorozatok egyesítése 16
Sorozatok összefésülése 16–17
Párok sorszáma egy sorozatban.....17–18
Arány.....18
Teljes négyzet.....18
Osztályátlagok szétválasztása.....19–20
Bűvös négyzet 20–21
Polinom értéke adott pontban.....21
Polinomok összege.....21–22
Polinomok szorzata 22
Buborékrendezés (Bubble-sort).....22–23

Egyszerű felcseréléses rendezés 23–24
Válogatásos rendezés 24
Minimum/maximum kiválasztásra épülő rendezés.....24–25
Beszúró rendezés.....25
Leszámláló rendezés.....25–26
Összefésülésen alapuló rendezés 26–27
Gyorsrendezés (QuickSort) 27
Szavak sorrendjének megfordítása 28
Faktoriális.....28
Számjegyösszeg 28
k elemű részhalmazok 29
Konverzió.....29
Az {1, 2, ..., n} halmaz minden részhalmaza.....30
Kamatos kamatok kiírása 30
Általános backtracking 31
Általános rekurzív backtracking 32
Elhelyezni 8 királynőt a sakktáblán.....32–33
Zárójelek 33–34
Játékok dobozva való elhelyezésének kiírása.....34–35
X pénzösszeg kifizetése n bankjegy segítségével 35–36
X Összeg kifizetése, minimum számú bankjeggyel.....36–37
Általános Divide Et Impera 37
Szorzat (DivImp).....37–38
Minimumszámolás (DivImp) 38
Hatványozás (DivImp) 39
Bináris keresés (DivImp).....39
Általános mohó (Greedy) algoritmus 40
Összeg (Greedy).....40
Hátizsák probléma (Greedy).....40–41
Összegkifizetés legkevesebb számú bankjeggyel (Greedy) 41–42

A Pascal nyelv elemei 42–54
Azonosítók 42–43
Alapértelmezett egyszerű típusok.....43
Változók 43–46
Konstansok.....46–48
Egész típusú konstansok 46
Valós típusú konstansok 47
Karakter és karakterlánc típusú konstansok 47

Logikai konstansok 48
Kezdőértékkel rendelkező változók 48
Adatok beolvasása és kiírása 48–49
Szabványos függvények és eljárások 49
Matematikai függvények és eljárások 49–52
Sorszámozott típusú adatokra vonatkozó függv. és elj. 52–54

Egyszerű progr. készítése – Elementáris algoritmusok 54–56
Kifejezések 55
Bitszintű műveletek 56
Relációs műveletek 56

Pascal nyelvű programozás 57–96
Döntések 57–62
Nagyobbik szám kiírása 57
Csökkenő sorrend létrehozása 57–58
Római szám felismerése 58–59
Római szám felismerése (case utasítással) 59–60
Kis- nagybetű vagy más karakter 60
Jegyértékelés 61
Hónapok napjainak száma 61–62
Ciklusok 62–67
Legnagyobb közös osztó (Eukleidész algoritmus) 62–63
Prímszámvizsgálat 63–64
Szám számjegyeinek száma 64–65
Törzstényezőkre való bontás 65
N szám összege 66
Fibonacci sorozat n-edik eleme 66–67
Egydimenziós tömbök 67–71
Átlagos jegynél kisebbek kiírása 67–68
Számsorozat kiírása fordítva 68
Sorozatból való teljes négyzetek összege 69
Karakterláncban felmerülő betük száma 70
Két polinom szorzata 71
Kétdimenziós tömbök 72–75
Két félévi átlagok 72–73
Négyzetes mátrix szimmetriája 73–74
Mátrix páros sorai elemeinek középarányosa 74–75
Karakterláncok 76–80

Relációs műveletek karakterláncokkal 77
Karakterlánc kezelő függvények 77–78
Karakterláncokat feldolgozó eljárások 78–79
Szóköz törlése egy karakterláncból 79–80
Karakterláncban folytatott szócsere 80
Halmazok 81–84
Műveletek halmazokkal 81–82
Halmazokra vonatkozó relációk és vizsgálatok 82
Egy halmaz elemeinek a kiírása 82
Karakterlánc betűinek kiírása ábécésorrendben 83
Erasztotenész algoritmusa (prímszámok) 83–84
Rekord vagy bejegyzés típusok 84–87
Alkalmazottak adatainak nyilvántartása 86–87
Kereső algoritmusok 87–89
Lineáris keresés 87–88
Bináris keresés 88–89
Rendező algoritmusok 90–93
Buborékrendezés 90–91
Minimumkiválasztásos rendezés 91–92
Beszúrásos rendezés 92–93
Összefésülések 93–96
Összefésülés strázsa nélkül 93–94
Összefésülés strázsával (ütközővel) 95–96

Rekurzivitás 96–113
Faktoriális kiszámítása 96–97
Fibonacci sorozat n-edik eleme 97
Két szám legnagyobb közös osztója 98
Hatványozás 98–99
10-es számrendszerből való alakítás 99–100
Első n páratlan szám összege 100–101
Természetes szám számjegyeinek összege 101
Számsorozat negatív elemeinek száma 102
Szám előfordulása egy sorozatban 103–104
Vektor páros elemeinek összege 104–105
Sorozat legnagyobb prímszáma 105–106
N hosszúságú a és b karakterű sorozatok 106–107
Természetes szám particiói 108–109
Labirintus-feladatok 109–113

„Oszd meg és uralkodj” – „Divide Et Impera” 113–121
Sorozat legnagyobb száma 113–114
N valós szám szorzata 114–115
Keresés számsorozatban 115–116
Hanoi tornyok 117
QuickSort 118–119
MergeSort 119–121

Kombinatorikus feladatok 121–125
Permutációk 121–122
Variációk 122–124
Kombinációk 124–125

Listák 125–135
Lista tartalmának kiírása 126
Listákat kezelő eljárások és függvények 126–127
Lista k.-adik elemének kiírása 127
Lista bővítése rendezetten 128–130
Két rendezett lista összefésülése 130–134
Verem 134–135

Szöveges állományok 136–143
Állomány megnyitása 137
Állomány bezárása 138
Írás állományba 138–139
Olvasás állományból 139–140
Szöveges állományok feldolgozása 140–143
Hibakezelés 140–141
A sor végének az ellenőrzése 141
Az állomány végének az ellenőrzése 141–143

Gráfok 143–160
Irányított és nem irányított gráfok 143
Definíciók 143–147
Dijkstra-algoritmus 147–150
Floyd-algoritmus 150–152
Minimális értékű Hamilton-kör bejárása 152–154
Szélességi bejárás 154–156
Keresési fa 156–160

Algoritmusok - pszeudókód

Abszolút érték

Határozzuk meg és írjuk ki adott valós szám abszolút értékét!

```
Algoritmus Abszolút_érték(x,mod):
  Ha x ≥ 0 akkor          {bemeneti adat: x, kimeneti adat: mod}
    mod ← x
  különben
    mod ← -x
  vége(ha)
Vége(algoritmus)
```

Hányados ismételt kivonással

Számítsuk ki két természetes szám egész hányadosát ismételt kivonásokkal!

```
Algoritmus Osztás(a,b,hányados):
  hányados ← 0                {bemeneti adatok:a, b, kimeneti
adat:hányados}
  Amíg a ≥ b végezd el:
    hányados ← hányados + 1
    a ← a - b
  vége(amíg)
Vége(algoritmus)
```

Legnagyobb közös osztó

Számítsuk ki két természetes szám legnagyobb közös osztóját!

```
Algoritmus Eukleidész(a,b,lnko):
  Ismételd          {bemeneti adatok: a, b, kimeneti adat: lnko}
    r ← maradék[a/b]        {kiszámítjuk az aktuális maradékot}
    a ← b                   {az osztandót felülírjuk az osztóval}
    b ← r                    {az osztót felülírjuk a maradékkal}
```

```
  ameddig r = 0    {amikor a maradék 0, véget ér az algoritmus}
  lnko ← a                {lnko 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 ← 0
  Minden i=1,n végezd el:
    Be: szám
    Ha szám páros akkor
      db ← 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 ← szám {bemeneti adat: szám, kimeneti adat: válasz}
  újszám ← 0
  Amíg szám > 0 végezd el:
    számjegy ← maradék[szám/10]
    újszám ← újszám*10 + számjegy
    szám ← [szám/10]
  vége(amíg)
  válasz ← újszám = másolat
                    {ha újszám = másolat, akkor válasz értéke igaz}
                  {ha újszám ≠ másolat, akkor válasz értéke hamis}
Vége(algoritmus)
```

```
  vége(minden)
  Ki: 'Az algoritmus nem talált megoldást.'
Vége(algoritmus)

Algoritmus Kiír(i,sz,b,nrb,X):
  Ki: X, ' kifizethető ', nrb, ' bankjeggyel: '
  Minden j=1,i végezd el:
    Ha nrj > 0 akkor
      Ki: szj, ' darab a ', bj, ' bankjegyből.'
    vége(ha)
  vége(minden)
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 „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 hatvany, 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 szóközök }
- 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;
```

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 elso szam: ');
  ReadLn(szam1);
  Write('A masodik szam: ');
  ReadLn(szam2);
  If szam1 > szam2 Then
    max := szam1
  Else
    max := szam2;
  WriteLn('A nagyobbik szam: ', 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 elso szam: ');
  ReadLn(szam1);
```

```
  Write('A masodik szam: ');
  ReadLn(szam2);
  Write('A harmadik szam: ');
  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.
```

Kis- nagybetű vagy más karakter

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

```
Program vizsgal1;
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.
```

```
      Ellenorzes;
    Until not modositott or megfelel;
    If modositott Then
    Begin
      Vizsgal;
      If tele Then
        Kiiras
      Else
      Begin
        Inc(stP);
        Init
      End
    End
    Else
      Dec(stP)
  End;
  ReadLn;
  Close(fout)
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: ');
  For i := 1 To n Do
    Read(x[i]);
  ReadLn;
  WriteLn('A szamok 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 novekvo sorozat: ');
  For i:=1 To n Do
    Read(x[i]);
  Write('Keresett szam: ');
  ReadLn(ker);
  bin_kereses(1,n);
  ReadLn;
End.
```