

**ÉRETTSÉGI VIZSGA • 2011. október 17.**

# **INFORMATIKAI ALAPISMERETEK**

## **KÖZÉPSZINTŰ ÍRÁSBELI ÉRETTSÉGI VIZSGA**

### **JAVÍTÁSI-ÉRTÉKELÉSI ÚTMUTATÓ**

**NEMZETI ERŐFORRÁS  
MINISZTERIUM**

---

---

## Fontos tudnivalók

### *Általános megjegyzések:*

Ha egy kérdésre a jó válasz(ok) mellett a vizsgázó válaszában hibás választ is megjelöl, akkor a kérdésre adható pontszámból le kell vonni a rossz válaszok számát. Negatív pontszám nem adható, ezért több hibás válasz esetén a minimális pontszám nullánál kevesebb nem lehet.

Pl. Ha egy jó válasz mellett a vizsgázó egy hibás választ is bejelöl, akkor 0 pontot kell adni.

Egyes esetekben előfordulhat, hogy egy általánostól eltérő rendszer használata miatt valamely kérdésre a vizsgázó nem a várt válasz adja, de a válasza és az indoklása elfogadható. Ilyen esetben a kérdésre adható pontszámot meg kell adni.

Pl. Táblázatkezelőkben magyar beállításnál a tizedesek elválasztásának a jele a vessző, és ez a várt válasz. Ha a vizsgázók munkájuk során angol beállítást használnak, vagy a vizsgázó odaírja ezt megjegyzésként, akkor az előző helyett az angol beállítású környezetben használt pont lesz a helyes válasz.

**A javítási-értékelési útmutatóban feltüntetett válaszokra kizárólag a megadott pontszámok adhatók.**

**A megadott pontszámok további bontása csak ott lehetséges, ahol erre külön utalás van. Az így kialakult pontszámok csak egész pontok lehetnek.**

---

# I.

## Teszt jellegű, illetve egyszerű, rövid szöveges választ igénylő feladatok

### Hardver

- 1) b ..... 1 pont
- 2) d, a, b, c ..... 1+1+1+1 pont
- 3) b ..... 1 pont
- 4) H, H, I, I ..... 1+1+1+1 pont
- 5) d ..... 1 pont
- 6) c ..... 1 pont
- 7) c, d, b, a ..... 1+1+1+1 pont
- 8) a ..... 1 pont
- 9) I, I, H, H ..... 1+1+1+1 pont

### Szoftver

- 10) H, H, H, H ..... 1+1+1+1 pont
- 11) b ..... 1 pont
- 12) H, I, H, I ..... 1+1+1+1 pont
- 13) c ..... 1 pont
- 14) d, a, b, c ..... 1+1+1+1 pont

### Szövegszerkesztés, táblázatkezelés

- 15) b ..... 1 pont
- 16) c ..... 1 pont
- 17) a ..... 1 pont
- 18) b ..... 1 pont

### Informatikai alapok

- 19) d ..... 1 pont
  - 20) c ..... 1 pont
  - 21) H, H, I, I ..... 1+1+1+1 pont
-

**Hálózati ismeretek, HTML**

- 22) c..... 1 pont  
23) c..... 1 pont  
24) a..... 1 pont  
25) d..... 1 pont  
26) b..... 1 pont

A feladatokra adható összes pontszám: 50 pont

---

## II.

### Programozási, illetve adatbázis-feladatok számítógépes megoldása

#### 1. feladat

**10 pont**

Kódolja az alábbi algoritmust egy konkrét programozási nyelven!

*A feladat megoldásaként teljes, fordítható és futtatható kódot kérünk, mely az adatokat – ha szükséges – a billentyűzetről (standard input) olvassa, és a képernyőre (standard output) írja ki. Vizuális fejlesztőeszköz használata esetén az algoritmust konzol alkalmazásként (szöveges ablakban futó) kérjük elkészíteni! A Véletlenegész függvény a megadott határok közötti véletlen egész számot generál. Ha a függvény nem létezik a használt nyelvben, akkor alkalmazzon megfelelő képletet, amely a megadott intervallumba eső véletlenszámot generál!*

Konstans

N=20, M=30

Változó

I, J, L: egész

A[1..N], B[1..M], C[1..N]: egész elemű tömb

Program

A[1]:=-50, Ki: A[1]

Ciklus I:=2-től N-ig

A[I]:=A[I-1]+Véletlenegész(1,3) // 1 és 3 közötti vél. egész szám

Ki: A[I]

Ciklus vége

B[1]:=-40, Ki: B[1]

Ciklus I:=2-től M-ig

B[I]:=B[I-1]+Véletlenegész(1,3) // 1 és 3 közötti vél. egész szám

Ki: B[I]

Ciklus vége

I:=1, J:=1, L:=0

Ciklus amíg (I<=N) és (J<=M)

Ha A[I]<B[J]

akkor

I:=I+1

különben

Ha A[I]>B[J]

akkor J:=J+1

különben

L:=L+1

C[L]:=A[I]

I:=I+1, J:=J+1

Elágazás vége

Elágazás vége

Ciklus vége

Ciklus I:=1-től L-ig

KI: C[I]

---

---

Ciklus vége  
Program vége.

- a) A programkód szintaktikailag hibátlan, lefordítható ..... 1 pont  
 – bármilyen elemi hiba esetén csak 0 pont adható
- b) A konstansok és változók helyes definiálása ..... 2 pont  
 – alkalmas egész típus használata: 1 pont  
 – pontos, az algoritmussal egyező definíciók és deklarációk: 1 pont
- c) A tömbök feltöltése a megfelelő véletlenszámokkal ..... 3 pont  
 – Ha az 1..3 intervallumba eső véletlenszámok generálása helyes: 1 pont  
 – Mindkét vektor generálása az algoritmus alapján történik, a vektorok teljesen feltöltöttek, és mindkét vektorba növekvő számsorozat kerül: 1 pont  
 – Ha a generált számok kiírásra kerülnek: 1 pont
- d) A feldolgozás (metszetképzés összefuttatással) helyes kódolása ..... 3 pont  
 – Ciklus helyes kódolása: 1 pont  
 – Külső elágazás helyes kódolása: 1 pont  
 – Belső elágazás helyes kódolása: 1 pont
- e) A C tömb kiírás ..... 1 pont

### Megoldás Delphi nyelven:

```
program Feladat1;

{$APPTYPE CONSOLE}
{
  Informatikai Alapismeretek
  Középpszint
  1. feladat
}
uses
  SysUtils, Math;
const N=20;
      M=30;
var I,J,L:integer;
    A:Array[1..N] of integer;
    B:Array[1..M] of integer;
    C:Array[1..N] of integer;

begin
  Randomize;
  WriteLn;
  WriteLn('A sorozat: ');
  A[1]:=-50; Write(A[1], ' ');
  For I:=2 to N Do
    Begin
      A[I]:=A[I-1]+RandomRange(1,3);
      Write(A[I], ' ');
    End;
  WriteLn;
```

---

---

```
WriteLn;
WriteLn('B sorozat: ');
B[1]:=-40; Write(B[1], ' ');
For I:=2 to M Do
  Begin
    B[I]:=B[I-1]+RandomRange(1,3);
    Write(B[I], ' ');
  End;
WriteLn;

I:=1; J:=1; L:=0;
While (I<=N) and (J<=M) do
  begin
    If A[I]<B[J]
    then
      I:=I+1
    else
      If A[I]>B[J]
      then J:=J+1
      else
        begin
          L:=L+1;
          C[L]:=A[I];
          I:=I+1;
          J:=J+1;
        end;
    end;

  WriteLn;
  Writeln('A két sorozat metszete:');
  For I:=1 to L Do
    Write(C[I], ' ');
  writeln;

  ReadLn;

end.
```

**2. feladat****10 pont**

A matematikában barátságos számoknak nevezzük azokat a pozitív egész számpárokat, amelyekre teljesül, hogy az egyik szám felírható a másik szám (saját magánál kisebb) osztóinak az összegeként, és fordítva!

Például: a 220 és a 284 barátságos számpár, mert

- 220 önmagánál kisebb osztói: 1, 2, 4, 5, 10, 11, 20, 22, 44, 55, 110 és  $1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284$ ; illetve
- 284 önmagánál kisebb osztói: 1, 2, 4, 71, 142 és  $1 + 2 + 4 + 71 + 142 = 220$ .

A következő számpárok szintén barátságosak: (220;284), (1184;1210), (2620;2924), (5020;5564), (6232;6368), (10744;10856).

Írjon programot, amely két megadott számról eldönti, hogy barátságos számok-e!  
Az adatbekérésnél ellenőrzi nem kell!

*A feladat megoldásaként teljes, fordítható és futtatható kódot kérünk, mely az adatokat – ha szükséges – billentyűzetről (standard input) olvassa, és a képernyőre (standard output) írja ki. Vizuális fejlesztőeszköz használata esetén az algoritmust konzol alkalmazásként (szöveges ablakban futó) kérjük elkészíteni! Beadandó a feladatot megoldó program forráskódja!*

- a) A programkód szintaktikailag hibátlan, lefordítható ..... 1 pont
  - Ez a pont csak abban az esetben adható meg, ha a programkód tartalmaz a b-d. szakaszokba tartozó, összességében legalább 3 pontot érő részmegoldást!
- b) Adatbekérés ..... 2 pont
  - A bemenő adatok deklarálása megfelelő egész típussal: 1 pont
  - Az adatok helyes bekérése: 1 pont
- c) Az egyik szám osztóinak az összegzése, és az összeg összehasonlítása a másik számmal..... 5 pont
  - Van olyan programrészlet, amely ezt a feladatot végzi el: 1 pont (ez a pont minden olyan esetben megadható, amikor a vizsgázó nyilvánvalóan felismerte, hogy ezt a részfeladatot meg kell valósítani!)
  - Helyes az összegzés intervalluma: 1 pont
  - Helyes az oszthatósági feltétel: 1 pont
  - Helyes az összegzés: 1 pont
  - Megtörténik az összeg és a szám összehasonlítása: 1 pont
- d) A barátságosság vizsgálata ..... 2 pont
  - Mindkét irányban megtörténik az előző pontban leírt tulajdonság vizsgálata: 1 pont
  - A két részeredményt ÉS művelet kapcsolja össze, és a végeredmény kiírásra kerül: 1 pont



---

**Megoldás Delphi nyelven:**

```
program Feladat2;

{$APPTYPE CONSOLE}
{
  Informatikai Alapismeretek
  Középszint
  2. feladat
}

uses
  SysUtils;
Var A,B:Longint;

Function Tulajdonsag(A,B:longint):Boolean;
// A függvény logikai igaz értéket ad vissza, ha az A
önmagánál kisebb
// osztóinak az összege megegyezik B-vel!
Var I,S:Longint;
begin
  S:=0;
  // A önmagánál kisebb osztóinak az összegzése
  For I:=1 to A div 2 do
    Begin
      If A mod I=0
        then S:=S+I;
    end;
  Tulajdonsag:=S=B;
  // Igazat ad vissza, ha ez az összeg megegyezik B-vel!
end;

begin
  Writeln;
  Write('Adja meg az egyik számot: '); Readln(A);
  Write('Adja meg a másik számot : '); Readln(B);
  Writeln;
  Write('A megadott számok ');
  If tulajdonsag(A,B) and Tulajdonsag(B,A)
  // Ha a tulajdonság kölcsönösen teljesül,
  Then Writeln('baratsagos számok!')
    //akkor a számpár barátságos
  Else Writeln('nem baratsagos számok!');
    //különben nem!
  Readln;
end.
```

**3. feladat****15 pont**

Írjon programot, amely egy dolgozat eredményeinek az elemzését segíti!

- A felhasználónak legyen lehetősége beírni a tanulók nevét és pontszámát! Az elérhető maximális pontszám 50, amelyet konstans adatként tároljon a program! (A konstans értékének a változása a program helyes működését ne befolyásolja!) A pontszám bevitele ellenőrzött legyen! A nevek bevitelét nem kell ellenőrizni! Az adatbevitel végét a felhasználó a név helyett beírt üres stringgel jelezze! Maximum 20 tanuló eredményének a bevitelére legyen lehetőség!
- A program minden tanuló esetében határozza meg a százalékos eredményt, illetve az érdemjegyet, majd írja ki ezeket táblázatos formában! A 2-es érdemjegyhez legalább 40%-os, a 3-ashoz legalább 55%-os, a 4-eshez legalább 70%-os, az 5-öshöz legalább 85%-os eredmény szükséges!
- A program határozza meg az osztályzatok átlagát, és írja ki két tizedesjegy pontossággal !

*A feladat megoldásaként teljes, fordítható és futtatható kódot kérünk, mely az adatokat billentyűzetről (standard input) olvassa, és a képernyőre (standard output) írja ki. Vizuális fejlesztőeszköz használata esetén az algoritmust konzol alkalmazásként (szöveges ablakban futó) kérjük elkészíteni!*

*Beadandó a feladatot megoldó program forráskódja!*

- a) A programkód szintaktikailag hibátlan, lefordítható ..... 1 pont
  - Ez a pont csak abban az esetben adható meg, ha a programkód tartalmaz a b-f szakaszokba tartozó, összességében legalább 5 pontot érő részmegoldást!
- b) A konstansok és változók helyes definiálása ..... 2 pont
  - A nevek tárolására alkalmas tömb helyes deklarálása: 1 pont
  - A pontszámok tárolására alkalmas tömb helyes deklarálása: 1 pont
  - A százalék és osztályzat tömb deklarálása nem kötelező, a program ezek nélkül is megírható (bár a javasolt megoldás a számított adatok tömbben tárolása!)
  - Amennyiben konstansdefiníciókban, vagy egyéb változódeklarációkban hiba van, hibánként -1 pont, de minimum 0 pont jár erre a feladatrésze.
- c) Ellenőrzött beolvasás ..... 4 pont
  - Nevek és pontszámok eltárolásra kerülnek: 1 pont
  - A pontszámok beolvasása ellenőrzött: 1 pont  
(Elég, ha hibaüzenet nélkül működik az ellenőrzött adatbevitel)
  - A beolvasás végjelig tart: 1 pont  
(Csak akkor adható meg a pont, ha nem történik indextúllépés)
  - A nevek darabszáma tárolásra kerül  
(vagy a program további része megállapítja azt): 1 pont
- d) Százalék és osztályzatok megállapítása ..... 3 pont
  - Százalékos teljesítmények helyes kiszámítása: 1 pont
  - Osztályzatok helyes megállapítása: 2 pont (a pontszám nem osztható)
  - A pontok akkor is járnak, ha az eredmények nem kerülnek tárolásra, de a program kiszámítja azokat
- e) Táblázatszerű kiírás ..... 3 pont  
(Hibánként -1 pont, minimum 0 pont)  
Lehetséges hibák:
  - Nincs név, százalék, vagy érdemjegy
  - Nincs fejléc

- 
- f) Átlageredmény kiszámítása ..... 2 pont
- Összegzés tétel helyes alkalmazása: 1 pont
  - Átlageredmény helyes: 1 pont

**Megoldás Delphi nyelven:**

```
program Feladat3;

{$APPTYPE CONSOLE}
{
  Informatikai Alapismeretek
  Középszint
  3. feladat
}

uses
  SysUtils;

Const Max=20;
      Maxpont=50;
Var Nevek:Array[1..Max] of String[40];
    Pont,Szazalek,Osztalyzat:Array[1..Max] of Byte;
    I,Db:byte;
    Atlag:real;

begin
  {Beolvasás}
  WriteLn;
  I:=0;
  Repeat // Nevek és pontok beolvasása, üres string végjelig
    Inc(I);
    Writeln(I, '. tanulo:');
    Write(' Nev: ');
    ReadLn(Nevek[I]);
    If Nevek[I]<>' ' // Ha nem akar kilépni a felhasználó a bevitelből
    Then
      Repeat // Következő pont ellenőrzött beolvasása
        Write(' Pontszám: ');
        ReadLn(Pont[I]);
        Until (Pont[I] In [0..Maxpont]); // Tovább lép, ha jó a pont
      Writeln;
  Until (Nevek[I]=' ') Or (I=Max); // Vége, ha végjel, vagy megtelt
  If (Nevek[I]=' ') Then Db:=I-1 // Darabszám korrigálás, ha szükséges
  Else Db:=I;
```

---

---

```
{Feldolgozás}
For I:=1 to Db do
  Begin
    Szazalek[I]:=Trunc((Pont[I]/Maxpont)*100);
    // A százalékos teljesítmények meghatározása
    Case Szazalek[I] of
      // Az osztályzatok meghatározása
      0..39 :Osztalyszat[I]:=1;
      40..54 :Osztalyszat[I]:=2;
      55..69 :Osztalyszat[I]:=3;
      70..84 :Osztalyszat[I]:=4;
      85..100:Osztalyszat[I]:=5;
    End;
  End;

// Táblázatszerű kiírás
Writeln('Nev':20,'Pont':10,'%':10,'Jegy':10);
For I:=1 to Db do
  Begin
    Writeln(Nev[I]:20,' ',Pont[I]:8,' ',Szazalek[I]:10,'
',Osztalyszat[I]:8);
  End;

// Osztályzatok átlagának meghatározás
Atlag:=0;
For I:=1 to Db do
  Begin
    Atlag:=Atlag+Osztalyszat[I];
  End;
Atlag:=Atlag/Db;
Writeln;
Writeln('Osztalyszatlag:',Atlag:8:2);
ReadLn;
End.
```

**4. feladat****15 pont**

- A. Hozzon létre egy „**tanulok**” nevű adatbázist! Az adatbázison belül hozzon létre egy „**pontok**” nevű táblát! Hozza létre a szükséges mezőket a megfelelő típussal és állítsa be a kulcsmezőt! Töltse fel a megadott adatokkal! A táblázat a tavalyi informatikai alapismeretek vizsgára vonatkozó adatokat tartalmaz!

| Név<br>(kulcsmező) | Osztály | Szóbeli ideje | Teszt | Gyakorlat | Szóbeli |
|--------------------|---------|---------------|-------|-----------|---------|
| Horváth Lénárd     | 12.D    | 2006.05.27    | 31    | 36        | 46      |
| Kiss Tamás         | 12.D    | 2006.05.28    | 20    | 49        | 40      |
| Kovács Lajos       | 11.E    | 2006.05.29    | 26    | 35        | 46      |
| Nagy Lujza         | 11.E    | 2006.05.27    | 39    | 40        | 48      |
| Pál László         | 12.E    | 2006.05.27    | 21    | 30        | 46      |
| Sajó Andrea        | 12.D    | 2006.05.28    | 27    | 46        | 45      |
| Szabó Laura        | 12.E    | 2006.05.27    | 30    | 43        | 46      |

- B. Készítsen lekérdezést, mely listázza az olyan tanulók összes adatát, akiknek a keresztnéve L-lel kezdődik, nem előrehozott érettségit tettek, és mindhárom vizsgarészből jobb eredményt értek el, mint az adott feladatrésze vonatkozó átlageredmény! (Tehát pl. a teszten jobb eredményt ért el, mint a tesztek átlageredménye, és így tovább!)

**Megjegyzés:** Azon adatbázis-kezelőknél, ahol adatbázisokat nem tudunk létrehozni, csak táblákat, ott adatbázis helyett alkönyvtár (mappát) készítsünk, és ebben hozzuk létre a táblát megvalósító fájlt. Ekkor a beadandó a létrehozott alkönyvtár (mappa) és tartalma.

Amennyiben az adatbázis létrehozása és feltöltése nem az adott keretrendszerből, hanem valamilyen programnyelvi kóddal (pl. SQL) történik, beadandó a használt forrásnyelvű kód is.

- a) Az adatbázis és a tábla létrehozása ..... 4 pont
- Létezik az adatbázis és a tábla, a nevük a megadott: 1 pont
  - Léteznek a megfelelő típusú és nevű adatmezők: 2 pont  
(hibánként -1 pont, minimum 0 pont)
  - A kulcs beállításra került: 1 pont
- b) Az adatok helyes és pontos felvitele ..... 3 pont
- A 3 pont csak abban az esetben adható meg, ha az adatbevitel semmiféle hibát nem tartalmaz!
  - Hibánként -1 pont, minimum 0 pont.
- c) A lekérdezés helyes megfogalmazása ..... 8 pont
- A lekérdezés létezik és a tanulók összes adatát listázza <sup>1</sup>: 1 pont
  - Legalább egy pontszám átlagának a meghatározása, segédlekérdezéssel vagy beágyazott lekérdezéssel <sup>2</sup>: 1 pont
  - Mindhárom pontszám átlagának a meghatározása, segédlekérdezéssel vagy beágyazott lekérdezéssel <sup>3</sup>: 1 pont
  - Helyes szűrés a keresztnévre <sup>4</sup>: 1 pont
  - Helyes szűrés az előrehozott érettségire <sup>5</sup>: 1 pont
  - Helyes szűrés legalább egy vizsgarészre <sup>6</sup>: 1 pont
  - Helyes szűrés mindhárom vizsgarészre <sup>7</sup>: 1 pont
  - A lekérdezés a mintának megfelelő eredményt adja: 1 pont

---

Egy lehetséges megoldás a következő:

Figyelem: a megoldásban szerepeltetett felső indexek az előbbieken felsorolt részfeladatokat jelölik, nem részei az SQL lekérdezésnek!

Segédlekérdezés:

```
SELECT Avg(pontok.Teszt) AS AvgOfTeszt, Avg(pontok.Gyakorlat) AS AvgOfGyakorlat,  
Avg(pontok.Szóbeli) AS AvgOfSzóbeli  
FROM pontok;2;3
```

A fő lekérdezés:

```
SELECT pontok.Név, pontok.Osztály, pontok.[Szóbeli ideje], pontok.Teszt,  
pontok.Gyakorlat, pontok.Szóbeli  
FROM pontok, Segéd1  
WHERE (((pontok.Név) Like "* L*")4 AND ((pontok.Osztály) Not Like "11*")5 AND  
((pontok.Teszt)>[Segéd]![AvgOfTeszt]) AND  
((pontok.Gyakorlat)>[Segéd]![AvgOfGyakorlat]) AND  
((pontok.Szóbeli)>[Segéd]![AvgOfSzóbeli]))6;7;
```