

ÉRETTSÉGI VIZSGA • 2009. október 19.

INFORMATIKAI ALAPISMERETEK

KÖZÉPSZINTŰ ÍRÁSBELI ÉRETTSÉGI VIZSGA

JAVÍTÁSI-ÉRTÉKELÉSI ÚTMUTATÓ

**OKTATÁSI ÉS KULTURÁLIS
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.

Az írásbeli vizsgafeladatok pontszámainak összege csak egész szám lehet. Ha az írásbeli vizsgarész pontszáma nem egész szám, akkor a matematikai kerekítés szabályai szerint kell eljárni (az öttizedre vagy az a felett végződő pontszámokat felfelé, az öttized alatti pedig lefelé kerekítjük).

Teszt jellegű, illetve egyszerű, rövid szöveges választ igénylő feladatok megoldása**Hardver**

- 1) b, d, a, c (helyes válaszonként 1-1 pont) 4 pont
2) d..... 1 pont
3) b..... 1 pont
4) H, I, I, H (helyes válaszonként 1-1 pont) 4 pont
5) b..... 1 pont
6) c..... 1 pont
7) I, H, I, H (helyes válaszonként 1-1 pont) 4 pont
8) 1. sor: a, b ; 2. sor: a, c ; 3. sor: a, b, d ; 4. sor: a, b
(helyes soronként 1-1 pont)..... 4 pont
9) a..... 1 pont

Szoftver

- 10) b..... 1 pont
11) d, c, a, b (helyes válaszonként 1-1 pont)..... 4 pont
12) d..... 1 pont
13) a:féreg (worm), b:polimorf vírus, c:trójai program, d:makró vírus
(helyes válaszonként 1-1 pont)..... 4 pont
14) H, H, I, H (helyes válaszonként 1-1 pont) 4 pont

Szövegszerkesztés, táblázatkezelés

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

Informatikai alapok

- 19) c..... 1 pont
20) d..... 1 pont
-

21)

Decimális érték	Bináris kód (8 bites)	Hexadecimális kód (2 jegyű)	2-es komplement kód (8 bites)
93	01011101	\$5D	
-8			11111000
194		\$C2	

(helyes megoldásonként 1-1 pont)

4 pont

Hálózati ismeretek, HTML

22) d..... 1 pont

23) d..... 1 pont

24) a..... 1 pont

25) c..... 1 pont

26) b..... 1 pont

A feladatokra adható összes pontszám: 50 pont

Programozási, illetve adatbázis-feladatok számítógépes megoldása**1. feladat****10 pont**

Az alábbi algoritmus olyan 20 elemű, véletlen egész számokból álló sorozatot állít elő, amelyben pontosan 5 db prím van. Kódolja az algoritmust az alábbi programozási nyelvek egyikén: Pascal, Java, BASIC, C, C++ vagy C# !

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. (A határokat is beleértve.) Ha a függvény nem létezik a használt nyelvben, akkor alkalmazzon megfelelő képletet, amely a megadott intervallumba eső véletlenszámot generál! Az algoritmusban alkalmazott := operátor értékadást jelent!

Konstans N=20

Változó A[1..N]:egész elemű tömb

Függvény Prim(Szam:egész):logikai

Változó I:egész

L:logikai

L:=Hamis

I:=1

Ciklus amíg (I*I<Szam) és Nem(L)

I:=I+1

L:=Szam Mod I=0

Ciklus vége

Prim:=Nem(L)

Függvény vége

Eljárás Generalas:

Változó I,Db:egész

Ciklus

Db:=0

Ciklus I:=1-től N-ig

A[I]:=Véletlenegész(2,100)

Ha Prim(A[I])

akkor Db:=Db+1

Elágazás vége

Ciklus vége

amíg (Db<>5)

Eljárás vége

Eljárás Kiiras:

Változó I:egész

Ciklus I:=1-től N-ig

Ha Prim(A[I])

Akkor Ki: A[I], ' (prim) '

Különben Ki: A[I], ' (nem prim) '

Elágazás vége

Ciklus vége

Eljárás vége

Program Szamok

Generalas

Kiiras

Program vége.

-
- a) A programkód szintaktikailag hibátlan, lefordítható, eljárásokra tagolt 1 pont
- a pont csak abban az esetben jár, ha legalább egy eljárás vagy függvény definiálásra és meghívásra került
- Megjegyzés: A későbbiekben már nem kell pontot levonni, ha a program esetleg nem használ eljárásokat!*
- b) A globális konstans és tömbváltozó helyes definiálása, deklarációja 1 pont
- c) Prim függvény helyes kódolása 3 pont
- A függvény fejlécének a kódolása helyes; a paraméter neve és típusa megadásra került; a visszaadott érték típusa megadásra került; a lokális változók deklarációja került: 1 pont
 - Az előtesztelő ciklus kódolása helyes; a ciklusfeltétel helyes; a ciklusmag tartalmazza a megadott utasításokat: 1 pont
 - A ciklus előtti értékadások kódolása helyes; megtörténik az érték visszaadás: 1 pont
- d) A General eljárás helyes kódolása 3 pont
- A külső (háttesztelő) ciklus helyes szervezése; a ciklusfeltétel megfelelő kódolása: 1 pont
 - A belső (számláló) ciklus helyes szervezése; a ciklusváltozó deklarációja került; a ciklusmag blokkba foglalása; a tömb a megfelelő tartományba eső véletlen értékekkel feltöltésre került: 1 pont
 - A prim elemek számlálása megtörténik; a számláláshoz szükséges változó helyes deklarációja: 1 pont
- e) A Kiíras eljárás helyes kódolása 2 pont
- Helyes ciklusszervezés; a ciklusváltozó helyes deklarációja; a ciklusmagban elágazás található helyes feltétellel: 1 pont
 - Megtörténik a tömbelemek, illetve minden tömbelem esetében a 'Prím', 'Nem prím' tulajdonság kiírása: 1 pont
-

2. feladat**10 pont**

Írjon programot, amely előállítja a felhasználó által megadott 16 bites, bináris, előjel nélküli egész szám hexadecimális alakját! A hexadecimális forma előállításához az alábbiakban részletezett algoritmust használja!

Bináris érték átváltása hexadecimális formára:

- a bináris számjegyek sorozatát 4 bites csoportokra osztjuk
- meghatározzuk az egyes csoportok értékét 10-es számrendszerben
- a kapott értékeket egy-egy hexadecimális számjeggyé alakítjuk

Például:

Adott a következő 16 bites bináris szám: 1011001111011010

- 4 bites csoportokra bontjuk: 1011 0011 1101 1010
- a 4 bites csoportok 10-es számrendszerbeli értéke sorrendben: 11 3 13 10
- a kapott értékek hexadecimális számjegy formájában: B 3 D A
- A hexadecimális alak tehát: \$B3DA

A programnak semmiféle ellenőrzést nem kell végeznie, feltételezzük, hogy a felhasználó pontosan 16 bit hosszúságú bináris számot adott meg!

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! A feladat megoldása az alábbi nyelvek valamelyikén készüljön: Pascal, Java, BASIC, C, C++ vagy C#! 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 3 pontot érő részmegoldást!
- b) A változók helyes definiálása 1 pont
 - A pont abban az esetben adható meg, ha valamennyi, a feladat bemenő és kimenő adatait tárolni képes változó, illetve valamennyi segédváltozó megfelelően deklarálásra került
- c) Adatbekérés 1 pont
 - A pont abban az esetben adható meg, ha a bemenő adat bekérése megtörtént, a bekérés a felhasználó számára egyértelmű volt
- d) A bináris szám 4 bites csoportokra bontása 2 pont
 - A program legalább egy 4 bites csoportot helyesen elkülönít: 1 pont
 - A program minden 4 bites csoportot helyesen elkülönít: 1 pont

Megjegyzés: a pontok természetesen akkor is járnak, ha a csoportok nem kerülnek kimásolásra, de a feldolgozásuk megfelelően megtörténik.
- e) 4 bites csoportok értékének a meghatározása 3 pont
 - Helyes ciklusszervezés; a 4 bit bejárása megtörténik: 1 pont
 - A bitek helyi értékeinek a helyes meghatározása: 1 pont
 - A program a csoportok értékét helyesen határozza meg: 1 pont
- f) A hexadecimális alak meghatározása 2 pont
 - Minden hexadecimális számjegy meghatározásra kerül: 1 pont
 - A hexadecimális számjegyek megfelelő sorrendben kiírásra kerülnek: 1 pont

3. feladat**15 pont**

Egy akrobatikus snowboard versenyen részt vevő versenyzők kétféle pontszámot kapnak: az egyiket a bemutatott gyakorlat nehézségére, a másikat pedig a kivitelezés minőségére. Mindkét pontszám értéke legalább 1 és legfeljebb 10 lehet. Törtpontszám is adható, de csak olyan, amelynek a törtrésze 0,5, pl. 3,5 vagy 8,5. A versenyen legfeljebb 10-en vehetnek részt!

Írjon programot, amely megvalósítja a versennyel kapcsolatosan az alábbi feladatokat!

- A program tegye lehetővé a versenyzők pontszámainak a beolvasását a billentyűzetről! A beolvasás során a program jelenítse meg a következő versenyző sorszámát, és adjon lehetőséget a kétféle pontszám beírására!
- A program ellenőrizze, hogy a beírt pontszámok a megfelelő intervallumba esnek-e, illetve, hogy a felhasználó nem írt-e be nem megengedett törtpontszámot (pl. 7,2)! Helytelen pontszám megadása esetén lehetőséget kell adni az újbóli beírásra, akár többször is, de hibaüzenetet nem kell adni! Típusellenőrzést nem kell végezni!
- Az adatbevitel végét a felhasználó határozza meg, úgy, hogy a következő versenyző első pontszámaként 0 végjelet ír be! Ekkor a másik pontszámot már ne is kérje a program!
- A program listázza ki táblázatszerűen (az alábbi mintához hasonlóan) a versenyzők sorszámát, valamint a gyakorlat nehézségére, illetve a kivitelezésre kapott pontszámot! A táblázatnak legyen fejléce!

Minta:

Sorszám	Nehézség	Kivitelezés
1.	6	4.5
2.	3.5	9

- A program határozza meg az első olyan versenyző sorszámát, akinek a gyakorlatára kapott kétféle pontszám között a különbség 2-nél nagyobb, majd írja ki azt!
A program ne vizsgálja meg a szükségesnél több versenyző pontszámait!

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! A feladat megoldása az alábbi nyelvek valamelyikén készüljön: Pascal, Java, BASIC, C, C++ vagy C#! 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-e szakaszokba tartozó, összességében legalább 5 pontot érő részmegoldást!
- b) A konstansok és változók helyes definiálása 2 pont
 - A pontszámok tárolására alkalmas tömb, vagy tömbök helyes deklarálása: 1 pont
 - Az egyéb szükséges konstansok, változók (pl. ciklusváltozók, a versenyzők számát tároló változó) helyes deklarálása: 1 pont

-
- c) Ellenőrzött beolvasás 5 pont
- A beolvasott pontszámok a későbbi feladatok végrehajtása érdekében tárolásra kerülnek: 1 pont
 - A program ellenőrzi, hogy a pontszámok a megfelelő intervallumba esnek, és hogy a törtpontszámok törtrésze csak 0,5 lehet; a részfeltételek között megfelelő logikai kapcsolat: 1 pont
 - Hibás adat beírása esetén a program lehetőséget ad a többszöri javításra: 1 pont
- Megjegyzés: Elég, ha hibaüzenet nélkül működik az ellenőrzött adatbevitel, hibaüzenetre nem jár külön pont!*
- A beolvasás a 0 végjelig tart, a végjel beírása után a másik pontszámot már nem kéri a program: 1 pont
- Csak akkor adható meg a pont, ha nem történik indextúllépés!*
- A versenyzők létszáma tárolásra kerül, vagy a program további része megállapítja azt: 1 pont
- d) Táblázatszerű kiírás: 2 pont
- Van megfelelő fejléc: 1 pont
 - A sorok tartalmazzák a versenyzők sorszámát és a két pontszámot: .. 1 pont
- e) A feladatban megadott tulajdonságú első versenyző sorszámának a meghatározása. 5 pont
- A keresés tesztelős ciklussal történik: 1 pont
- A pont csak akkor adható meg, ha üres sorozatra is értelemszerű eredményt ad a keresés!*
- Helyes a keresési intervallumra kezdetét meghatározó értékadás, illetve annak a végét meghatározó részfeltétel: 1 pont
 - Helyes a keresési tulajdonságot megfogalmazó részfeltétel, és helyes a részfeltételek közötti logikai kapcsolat: 1 pont
- Megjegyzés: Ha a program számlálós ciklussal helyes eredményt ad, akkor az előző 3 pont helyett 1 pont adható!*
- Eredményes keresés esetén a megfelelő sorszám kiírása: 1 pont
 - Eredménytelen keresés esetén üzenet kiírása: 1 pont
-

4. feladat**15 pont**

Egy természetjáró szakkörbe az iskola több osztályából járnak a tanulók.

Az első félév során mindenki befizetett egy összeget a kirándulások céljára. A befizetések adatai az alábbi táblázatban láthatók, ezeket szeretnénk egy adatbázisban tárolni!

- A. Hozzon létre egy „szakkorpenz” nevű adatbázist! Az adatbázison belül hozzon létre egy „befizetesek” nevű táblát! Hozza létre a szükséges mezőket a megfelelő típussal és állítsa be elsődleges kulcsként a „nev” mezőt! Töltse fel az adattáblát az alább megadott adatokkal!

nev (elsődleges kulcs)	osztaly	szulev	datum	osszeg
Apor Vilmos	10.E	1993	2008.10.25.	1 500 Ft
Bódy Elek	10.D	1993	2008.11.09.	2 300 Ft
Gyetzay Nándor	10.E	1992	2008.09.05.	2 000 Ft
Kalmár Petra	11.E	1991	2008.09.02.	1 800 Ft
Kiss Rózsa	11.F	1991	2008.10.23.	1 600 Ft
Suhajda László	10.D	1992	2008.11.30.	1 200 Ft
Varga Csaba	10.E	1993	2008.12.05.	3 000 Ft

- B. Készítsen lekérdezést, mely megadja azon tanulók nevét, osztályát és életkorát, akik a minimális befizetéshez képest legalább 500 Ft-tal többet fizettek be a szakköri költségekre! A lista életkor szerint csökkenően legyen rendezve!

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)
 - Az elsődleges kulcs megfelelően beállításra került: 1 pont
- b) Az adatok helyes és pontos felvitele 4 pont
- A 4 pont csak abban az esetben adható meg, ha az adatbevitel semmiféle hibát nem tartalmaz!
 - Hibásan bevitt értékenként -1 pont, minimum 0 pont.
- c) A lekérdezés helyes megfogalmazása 7 pont

Fő lekérdezés:

- A lekérdezés létezik, listázza a tanuló(k) nevét, osztályát, és a kért adatokon kívül más adatot nem listáz¹: 1 pont
- Az életkorra vonatkozó számított mező létezik és helyes, az életkor a listában látható²: 1 pont
- Létezik és helyes a befizetett összegre vonatkozó szűrés: ³ 1 pont
- A lista az életkor szerint csökkenően rendezett ⁴: 1 pont

A minimális befizetett összeg meghatározása (pl. segédlekérdezéssel^{5, 6}) 2 pont

A lekérdezés a mintának megfelelő eredményt adja: 1 pont

(Amennyiben a vizsgázó indokolta (pl. leírta), hogy a feladat szövegéből nem derül ki vagy nem derülhet ki, hogy a minimális befizetést függvénnel (vagy milyen más módon) kell meghatározni, akkor részére a fenti 2+1 pont megadható.)

Egy lehetséges megoldás a következő:

(A megoldásban szerepeltetett felső indexek az előbbieken felsorolt részfeladatokat jelölik, nem részei az SQL lekérdezésnek!)

A fő lekérdezés:

```
SELECT befizetesek.nev1, befizetesek.osztaly1, Year(Now())-befizetesek.szulev2 AS Kor  
FROM befizetesek, Seged  
WHERE (((befizetesek.osszeg)>Seged.MinOfosszeg+500))3  
ORDER BY Year(Now())-befizetesek.szulev DESC4;
```

Segédlekérdezés:

```
SELECT5 Min(befizetesek.osszeg)6 AS MinOfosszeg  
FROM befizetesek;
```

Értékelés:

A feladatokra adható összes pontszám: 50 pont

A két írásbeli rész összes pontszáma 100 pont.

Ahhoz, hogy a feladatok megoldását az érettségi jegy megállapításakor figyelembe lehessen venni, az összes pontszámnak legalább a 10%-át kell teljesíteni, ezért csak a legalább 10 pontos dolgozatok fogadhatók el a tantárgy végső osztályzatának a megállapításához.