

ÉRETTSÉGI VIZSGA • 2007. május 25.

INFORMATIKAI ALAPISMERETEK

EMELT SZINTŰ ÍRÁSBELI ÉRETTSÉGI VIZSGA

JAVÍTÁSI-ÉRTÉKELÉSI ÚTMUTATÓ

**OKTATÁSI ÉS KULTURÁLIS
MINISZTERIUM**

Általános megjegyzések

- Ha egy kérdésre a jó válasz(ok) mellett a tanuló 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 tanuló egy hibás választ is bejelöl, akkor 0 pontot kell adni. Ez nem vonatkozik azokra a kérdésekre, ahol a **(minden helyes részválasz 1 pont)** szöveg szerepel.

- Egyes esetekben előfordulhat, hogy egy általánostól eltérő rendszer használata miatt valamely kérdésre a tanuló nem a várt választ 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 diákok munkájuk során angol beállítást használnak, vagy a tanuló odairja ezt megjegyzésként, akkor az előző helyett az angol beállítású környezetben használt **pont** lesz a helyes válasz.

Egyszerű, rövid ill. kifejtendő szöveges választ igénylő feladatok

Hardver

- 1) A jumper átkötést jelent. A jumper a számítógép kártyáin található apró elektronikus kapcsoló, melyet kézzel kell beállítani a működéshez szükséges állapotba 1 pont
- 2) c. 1 pont
- 3) Összesen: 2 pont
 A TFT (Thin Film Transistor) aktív képpontos képernyő. Az olcsóbb LCD képernyőknek ugyanis az a hátránya, hogy a képpontok lassan gyulladnak ki és alszanak el (gyors mozgásoknál elhúzódó képet lehet látni), ráadásul csak szemből, illetve 30–40°-os szögből nézve adnak szép színes képet. 1 pont
 Ezeket a hibákat küszöbölik ki a TFT (Thin Film Transistor) képernyők. Ezeknél a látómező 140–150°-ra növekedett, másrészt sokkal gyorsabb lett a képpontok reakcióideje. 1 pont
- 4) H; H; I; I 4 pont
- 5) Összesen: 3 pont
 Az Extended (kiterjesztett) partícióból csak egy lehet. 1 pont
 A kiterjesztett partíciót azért hozták létre, hogy az elsődleges partíciókra meghatározott maximális értéknél (4) több partíciót is létre lehessen hozni. Ez a partíció az adatokat nem közvetlenül tartalmazza 1 pont
 A kiterjesztett partíció a lehetséges négy elsődleges partícióhely közül az egyiknek a helyét foglalja el a logikai partíciók elhelyezéséhez. 1 pont
Az aktív partícióból is csak egy lehet, de ez nem jó válasz, mivel ez nem partíció típus (Erre a válasza, /vagy részválasza/ pontszám nem adható)
- 6) Sorrend: b, d, a, c (minden helyes részválasz 1 pont) 4 pont
 (Ha vonalakkal köti össze, az is elfogadható.)

- 7) Scanner: Összesen: 3 pont
1. A készülékbe helyezett lapot egy mozgó fénycsík világítja meg. 1 pont
 2. A visszaverődő fényt egy tükör és a hozzá tartozó optika segítségével egy vevőbe kerül (CCD), amely analóg jelet állít elő... 1 pont
 3. Az analóg jel egy analóg/digitál (A/D) átalakítón keresztül jut a számítógépbe feldolgozásra. 1 pont
- 8) Összesen: 2 pont
- A digitális zoom a képpontokat nagyítja fel, ezáltal romlik a kép minősége..... 1 pont
- Az optikai zoom valódi nagyítást jelent, ahol a kép minősége (felbontása) változatlan marad 1 pont

Szoftver

- 9) d. 1 pont
- 10) Naplófájl: Összesen: 2 pont
- Valamilyen alkalmazás, szolgáltatás vagy az operációs rendszer által létrehozott üzeneteket tároló fájl. 1 pont
- Az üzenetek segítségével a rendszerben végrehajtott műveleteket lehet nyomon követni. A naplófájlok általában egyszerű szöveges (ASCII) fájlok, gyakran a „log” kiterjesztéssel..... 1 pont
- 11) H; I; H; H..... 4 pont
- 12) Fájlok töredezettsége: Összesen: 3 pont
- Egy lemezen lévő fájl részeinek elszórt elhelyezkedése a lemez különböző területein. 1 pont
- A töredezettség a lemezen lévő fájlok törlése után az új fájlok lemezre írásakor jön létre. 1 pont
- Töredezett fájlok esetén a számítógépnek a fájl minden megnyitásakor meg kell keresnie a merevlemezben a fájl részeit, ami megnöveli a válaszadási időt..... 1 pont
- 13) (minden helyes részválasz 1 pont) 4 pont

	Szoftver		Hardver	
	operációs rendszer	alkalmazás	periféria	egyéb
Digitális kamera				<u>X</u>
PenDrive			<u>X</u>	
Explorer		<u>X</u>		
Tömörítő program		X		

- 14) A HTTP (Hypertext Transfer Protocol) a weben keresztüli adatátvitel módját meghatározó protokoll. A HTTP-cím az URL-cím egy fajtája. Formátuma pl. a következő: <http://www.valami.com>. 1 pont

Szövegszerkesztés, Táblázatkezelés

- 15) Sorrend: b, c, a (minden helyes részválasz 1 pont)..... 3 pont
(Ha vonalakkal köti össze az is elfogadható)
- 16) Összesen:..... 4 pont
 INDEX (A1 : A6 ; HOL . VAN (MAX (C1 : C6) ; C1 : C6)) **Jaguár** 2 pont
 HOL . VAN (MAX (C1 : C6) ; C1 : C6 ; 0) **3** 2 pont

Informatikai alapok

- 17) 8 biten 2^8 db, azaz 256 különböző szám adható meg..... 1 pont
- 18) c. 1 pont
- 19) HW konfiguráció: Összesen:..... 2 pont
 Általában egy számítógépes rendszer teljesítményét és szolgáltatásait behatóró hardver elemek (számítógép és perifériák) összessége..... 1 pont
 Minimális hardver konfigurációról beszélünk, ha egy konkrét alkalmazási környezethez írjuk le a minimálisan szükséges hardver elemeket. 1 pont

Hálózati ismeretek, HTML

- 20) b. 1 pont
- 21) IP cím 1 pont
- 22) A HTML – Hyper Text Markup Language. A WWW szerverek számára kezelhető ún. információs lapok készítésének szabályait leíró nyelv. Szövegek, képek, hangok összefűzésére ad lehetőséget ún. linkek segítségével. 1 pont
- 23) d. 1 pont

A feladatokra adható összes pontszám: 50 pont

Programozási feladatok számítógépes megoldása**1. feladat (10 pont)**

Készítsen programot, amely beolvassa egy kocka **e** élhosszát, valamint egy papírlap **a** és **b** oldalhosszait, majd meghatározza, hogy a kocka elkészíthető-e az **a*b** méretű téglalap alakú papírból! Ha a kocka elkészíthető, a program jelenítse meg az „ELKÉSZÍTHETŐ”, ha nem, akkor a „NEM KÉSZÍTHETŐ EL” szöveget a képernyőn! A két szöveg közül természetesen csak az egyiket szabad megjeleníteni!

Definíció: Egy kockát egy téglalap alakú papírból elkészíthetőnek nevezünk, ha a papír rövidebb oldala legalább $3*e$, a hosszabb oldala pedig legalább $4*e$ hosszúságú.

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 változók helyes definiálása..... 2 pont
(Hibánként -1 pont, minimum 0 pont.)
- b) A program helyes kezdéséért és befejezéséért 2 pont
(megfelelő formátum és szintaktikai helyesség)
- c) Az adatok helyes feltöltése..... 2 pont
(Hibánként -1 pont, minimum 0 pont.)
- d) A helyes eredmény megjelenítése 4 pont
(Hibánként -1 pont, minimum 0 pont.)

2. feladat (10 pont)

Készítsen programot, amely egy mérőszorozat eredményeként adódó –1000 és 1000 közé eső számokat tárol egy tömbben (15 darabot, melyeket a billentyűzeten keresztül ad meg a felhasználó), majd meghatározza és képernyőre írja, hogy hány lokális csúcs volt a mérőszorozatban! Egy elemet lokális csúcsnak nevezünk, ha nagyobb a közvetlenül előtte és közvetlenül mögötte álló elemnél. Az első és az utolsó elem nem lehet lokális csúcs, mivel nincs őket megelőző, ill. követő elem.

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 változók helyes definiálására..... 2 pont
Hibás megoldásért arányosan kevesebb pont adható.
- b) A program helyes kezdéséért és befejezéséért 1 pont
(megfelelő formátum és szintaktikai helyesség)
- c) Adatok beolvasása..... 1 pont
Hibás megoldásért arányosan kevesebb pont adható.
- d) A lokális csúcsok számának helyes meghatározása 5 pont
Hibás megoldásért arányosan kevesebb pont adható.
- e) Az eredmény kiírása..... 1 pont

3. feladat (15 pont)

Statisztikai kísérlet sorozatot végzünk kockadobással. Készítsen programot, amely billentyűzetről beolvassa és tárolja 50 dobás eredményét (egy dobás eredmény az 1, 2, 3, 4, 5, 6 számok valamelyike), majd meghatározza, és képernyőre írja az előforduló leghosszabb „szigorúan monoton csökkenő” rész sorozat hosszát! A sorozat hosszán a benne szereplő dobások darabszámát értjük!

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 feladatnak és a megoldás menetének megfelelő változókat helyesen definiálja:..... 2 pont
Hibánként -1 pont, de minimum 0 pont.
- b.) A változókat megfelelő módon olvassa be: 2 pont
Hibánként -1 pont, de minimum 0 pont.
- c.) A szigorúan monoton csökkenő dobássorozatok helyes meghatározása: 3 pont
Hibánként -1 pont, de minimum 0 pont.
- d.) A dobássorozatok hosszainak helyes meghatározása: 3 pont
Hibánként -1 pont, de minimum 0 pont.
- e.) A minimum helyes meghatározása: 3 pont
Hibánként -1 pont, de minimum 0 pont.
- f.) Az eredményt megfelelő módon írhatja a képernyőre: 2 pont
Hibánként -1 pont, de minimum 0 pont.

4. feladat (15 pont)

Futóversenyek és futók adatainak tárolása és ezek lekérdezése a feladat. Legyen adott a **futver** nevű adatbázis, mely a futók és versenyek adatait tárolja. (Az adatbázist a vizsgabizottság által megadott helyen találhatja ASCII TEXT, illetve MS ACCESS 2000 formátumban).

Az adatbázis elsősorban feladatkitűzési céllal készült, így természetesen nem modellezi tökéletesen a való életben felmerülő cikknnyilvántartással kapcsolatos összes lehetséges helyzetet. A feladatokat az adott modell keretein belül kell megoldani.

Az adatbázis az alábbi táblákat (relációkat) tartalmazza:

(A „:” után az adott adat típusát adjuk meg, a „->” karakterek után pedig az esetlegesen meghatározott kapcsolatot. Az egyes tábláknál a kulcsot aláhúzott karakterekkel jelöljük.)

EGYESULET (
EAZON : Szöveg -> FUTO.EAZON
 ENEV : Szöveg
)

FUTO (
FAZON : Egész szám -> EREDMENY.FAZON
 FNEV : Szöveg
)

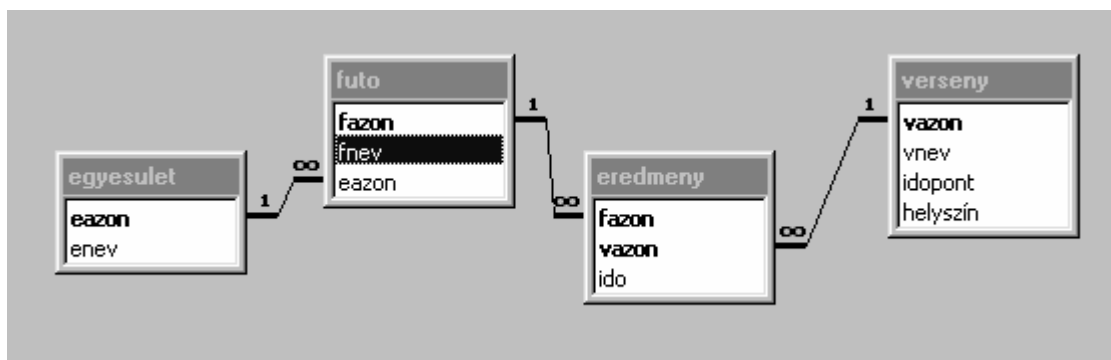
```
EAZON          : Szöveg          -> EGYESULET.EAZON
)
```

EREDMENY (

```
FAZON         :Egész szám       -> FUTO.FAZON
VAZON         :Szöveg          -> VERSENY.VAZON
IDO            :Dátum/Idő
)
```

VERSENY (

```
VAZON         :Szöveg          -> EREDMENY.VAZON
VNEV          :Szöveg
IDOPONT        :Dátum/Idő
HELYSZIN       :Szöveg
)
```



Az **EGYESULET** tábla a sportegyesületek azonosítóját (EAZON) és nevét (ENEV) tartalmazza. Az EAZON tulajdonképpen az egyesület nevének rövidítése.

A **FUTO** tábla a futók adataival van feltöltve. Egy sora tartalmazza a kulcsot (FAZON) kívül a futó nevét (FNEV) és egyesületének azonosítóját (EAZON).

Az **EREDMENY** tábla egy adott sora tartalmazza az adott futó (FAZON) adott versenyen (VAZON) elért időeredményét (IDO).

A **VERSENY** tábla a verseny azonosítóját (VAZON), nevét (VNEV), megrendezésének időpontját (IDOPONT), valamint a verseny helyszínét (HELYSZIN) tartalmazza.

Megjegyzés: A dátum/idő típusú mezőknél, a text formátumoknál, a konverzió miatt speciális null értékek szerepelnek (1899.12.30). Ezek a megoldás során figyelmen kívül hagyandóak, vagy az importálás során törölhetőek, vagy az adott rendszer számára megfelelő módon konvertálhatóak.

a.) Készítsen lekérdezést, mely megadja a versenyzők (futók) teljes listáját! A lekérdezés eredményében szerepeljen az egyesület neve és a versenyző neve („enev”, illetve „fnev” attribútum néven) és az eredmény legyen az egyesületek neve, azon belül pedig a versenyzők neve szerint növekvően rendezett!

SELECT enev,fnev..... 1 pont
 FROM futo,egyesulet..... 2 pont
 WHERE futo.eazon=egyesulet.eazon 1 pont
 ORDER BY 1,2;..... 1 pont

b.) Készítsen lekérdezést, mely megadja az „Olimpia” nevű verseny eredményét! A lekérdezés eredményében szerepeljen a futó neve, egyesületének azonosítója és az „Olimpia” nevű versenyen elért idő eredménye. Az eredménylista legyen „ido” szerint növekvően rendezett!

SELECT fnev,eazon,ido..... 1 pont
 FROM futo,verseny,eredmeny 1 pont
 WHERE futo.fazon=eredmeny.fazon AND
 verseny.vazon=eredmeny.vazon AND..... 1 pont
 verseny.vnev="Olimpia" 1 pont
 ORDER BY 3;..... 1 pont

c.) Készítsen lekérdezést, mely megadja az „Olimpia” nevű verseny első helyezettjét (helyezetteit) és eredményét (eredményeiket)! A lekérdezés eredményében szerepeljen a futó neve, egyesületének azonosítója és az „Olimpia” nevű versenyen elért idő eredménye. Az eredménylista holtverseny esetén az összes első helyezett adatait tartalmazza!

SELECT fnev,eazon,ido..... 1 pont
 FROM futo,verseny,eredmeny 1 pont
 WHERE futo.fazon=eredmeny.fazon AND
 verseny.vazon=eredmeny.vazon AND
 verseny.vnev="Olimpia" AND 1 pont
 ido = (SELECT MIN(ido)
 FROM verseny,eredmeny
 WHERE verseny.vazon=eredmeny.vazon
 AND verseny.vnev="Olimpia"); 2 pont

Összesen: 50 pont