

Examenul național de bacalaureat 2024

Proba E. d)  
INFORMATICĂ  
Limbajul C/C++

Varianta 3

*Filieră teoretică, profil real, specializare științe ale naturii*

- Toate subiectele sunt obligatorii. Se acordă zece puncte din oficiu.
- Timpul de lucru efectiv este de trei ore.
- Identificatorii utilizați în rezolvări trebuie să respecte precizările din enunț (**bold**), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.

**I. TÊTEL** (20 pont)

Az 1-től 5-ig számozott ítemek esetén, írja a vizsgalapra a helyes válasznak megfelelő betűt. Minden helyes válasz 4 pontot ér.

- Adj meg azt a C/C++ kifejezést, amelynek az értéke akkor és csakis akkor **1**, ha az **x** és **y** egész változóban tárolt számok párosok.
  - $x \% 2 == 0 \ \&\& \ (y + 1) \% 2 != 0$**
  - $(x - y) \% 2 == 0$**
  - $(x + y) \% 2 == 0$**
  - $x \% 2 == y \% 2$**
- Annak ellenőrzésére, hogy egy egydimenziós tömbnek van **x=16** értékű eleme, a bináris keresés módszerét használjuk, és a számok, amelyekkel az **x**-et összehasonlítjuk a módszer alkalmazása során **14, 24, 16**. Adj meg azt értéksort, amely a megadott sorrendben, lehet a tömb tartalma.
  - (2, 14, 7, 24, 12, 16, 48)**
  - (14, 24, 16, 14, 24, 16)**
  - (4, 8, 9, 14, 16, 24, 48)**
  - (14, 14, 24, 24, 16, 16)**
- Adj meg egy C/C++ kifejezést, amelynek az értéke akkor és csakis akkor **1**, ha az **x** egész típusú változóban tárolt szám a **{-3, -1, 1, 3}** intervallum eleme.
  - $\text{abs}(x) > 3 \ || \ x != 0$**
  - $\text{abs}(x - 3) <= 3 \ || \ \text{abs}(x \% 2) == 0$**
  - $\text{abs}(x - 3) > 0$**
  - $\text{abs}(x) <= 3 \ \&\& \ \text{abs}(x \% 2) == 1$**
- A mellékelt részletben minden változó egész típusú, és az **x** eredetileg a **[10, 10<sup>5</sup>]** intervallumból tárol egy számot.  
Adj meg egy kifejezést, amely a pontozott részt helyettesítheti úgy, hogy a kapott kód végrehajtása után az **u** változó az **x** számjegyei szorzatának az utolsó számjegyét tartalmazza.

```
u=1;
while(x!=0)
{
    u=.....;
    x=x/10;
}
```

  - $u * (x \% 10)$**
  - $(u * x) \% 10$**
  - $u \% 10 * x$**
  - $(u \% 10) * (x \% 10)$**
- Az alábbi utasítássorban minden változó egész típusú.

```
for(i=0; i<9; i++)
{
    for(j=0; j<9; j++)
        if(.....)
            cout<<"4 "; | printf("4 ");
        else
            cout<<"2 "; | printf("2 ");
    cout<<endl; | printf("\n");
}
```

4	4	4	4	2	2	2	2	2
4	4	4	2	2	2	2	2	2
4	4	2	2	2	2	2	2	2
4	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	4
2	2	2	2	2	2	2	4	4
2	2	2	2	2	2	4	4	4
2	2	2	2	2	4	4	4	4

Adj meg egy kifejezést, amely helyettesítheti a pontozott részt úgy, hogy a kód végrehajtása után a képernyőre a fenti képen látható értékek kerüljenek, ebben a sorrendben.

  - $i - j > 4 \ || \ i + j < 12$**
  - $i - j > 4 \ \&\& \ i + j < 12$**
  - $i + j < 4 \ || \ i + j > 12$**
  - $i + j > 4 \ \&\& \ i + j < 12$**

## II. TÉTEL

(40 pont)

### 1. Adott a mellékelt algoritmus pszeudokódban.

A **c** változó egész részét **[c]**-vel jelöltük.

- Adja meg a kiírt értéket, ha a beolvasott számok 5, 15, 27, 10, 1, 17, ebben a sorrendben. (6p.)
- Ha az **n**-be beolvasott érték 2, írjon egy különböző számokból álló halmazt, a **[0, 10<sup>3</sup>]** intervallumból, amelyeket folyamatosan beolvashat úgy, hogy az algoritmus végrehajtása után a kiírt érték 4 legyen. (6p.)
- Írja meg az adott algoritmusnak megfelelő C/C++ programot. (10p.)
- Írjon az adott algoritmusmal egyenértékű pszeudokód algoritmust, amelyben a **minden...végezd el** szerkezetet megfelelőképpen helyettesíti egy **amíg...végezd el** ismétlődő szerkezettel. (6p.)

```
beolvas n
(nem nulla természetes szám)
p ← 1
minden i ← 1, n végezd el
    beolvas x
    (természetes szám)
    ismételd
        x ← [x/3]
    ameddig x ≤ 3
        ha x ≠ 0 akkor
            p ← p * x
        kiír p
```

- Az **A** és **B** egydimenziós tömbök elemei: **A = (8, 4, 2)** és **B = (3, 8, 10, 17)**. Adja meg azon tömb elemeit, a megjelenés sorrendjében, amelyet az **A** és **B** tömbök növekvő sorrendbe való összefésülése során kap.

(6p.)

- A **pret** egész változó egy csokoládéfajta árát tárolja lejben megadva (természetes szám), míg a **tip** változó a neki megfelelő betűt tárolja: **N** betű a **ciocolata neagra** (fekete csoki) esetén vagy **I** betű a **ciocolata cu lapte** (tejes csoki) esetén.

Értelmezze a **tip** változót és írjon egy C/C++ utasítássort, amelynek nyomán a képernyőre íródik a csokoládé típusa és a következő sorba a **scumpa** szöveg, ha az ára szigorúan nagyobb mint 10 lej, vagy a **ieftina** üzenet, ellenkező esetben.

**Példa:** ha a **pret** változó a 16 értéket és a **tip** változó az **N** betűt tárolja, a képernyőre kiíródik

**ciocolata neagra**  
**scumpa**

(6p.)

## III. TÉTEL

(30 pont)

- Egy természetes számot **major impar**-nak nevezzük, ha a páratlan valódi osztóinak összege szigorúan nagyobb, mint a páros valódi osztóinak összege. A szám valódi osztói az 1-től és önmagától különböző természetes osztói.

**Példa:** a 18 „major impar” szám (a páros valódi osztói 2, 6, míg a páratlanok 3, 9, és  $3+9 > 2+6$ ).

Két természetes számot olvassunk be, az **a**-t és **b**-t ( $2 \leq a \leq b \leq 10^4$ ) és írassuk ki a legkisebb „major impar” számot az **[a, b]** intervallumból vagy a 0 értéket, ha az intervallumban nincs ilyen szám. Írja meg az algoritmust pszeudokódban.

**Példa:** ha **a=16**, **b=30**, akkor a kiírt érték 18.

(10p.)

- Írjon egy C/C++ programot, amely a billentyűzetről beolvas egy **n** ( $n \in [2, 50]$ ) természetes számot és egy egydimenziós tömb **n** elemét, egész számok a **[10, 10<sup>4</sup>]** intervallumból. A program írja ki a képernyőre a **DA** üzenetet és egy **k** természetes számot, egy szóközzel elválasztva, ha a tömb minden elemének **k** darab számjegye van vagy a **NU** üzenetet abban az esetben, ha nem minden értéknek van ugyanannyi számjegye.

**Példa:** ha **n=6** és a tömb (123, 241, 896, 908, 100, 536)

a képernyőre kiíródik a **DA 3**

és ha **n=6** és a tömb (123, 21, 896, 908, 100, 1536)

a képernyőre kiíródik a **NU**

(10p.)

- Egy túraútvonal mentén egy sor turisztikai jelzést használunk, amelyeket sorrendben kell követni. Minden jelzésre ismert a tengerszint feletti magasság (méterben kifejezve), amelyen található. **Lépcsőnek** nevezzük egy útvonalon belül egy jelzéssort, amely egymásutáni pozíciókat foglal el az útvonalon, és amely magasságai egymásutáni értékek, szigorúan növekvő sorrendben megadva.

Egy lépcsőt legalább két jelzés alkot, a hossza pedig egyenlő az őt alkotó jelzések számával.

A **bac.txt** állomány tartalma egy legfeljebb  $10^6$  tagú természetes számsor, a **[10, 10<sup>4</sup>]** intervallumból, egy-egy szóközzel elválasztva, amely egy túraútvonal jelzéseinek a magasságainak felelnek meg, előfordulásuk sorrendjében. Írassa ki a képernyőre az ezen az útvonalon található leghosszabb lépcső hosszát. Ha az útvonalon nem létezik egy lépcső sem, a képernyőre írja ki a **nu exista** üzenetet.

Tervezen memóriahasználat és végrehajtási idő szempontjából is hatékony algoritmust.

**Példa:** ha az állomány tartalmazza az 500 600 601 405 569 570 700 701 625 626 627 520 számokat, akkor a képernyőre kiírt érték a 3.

- Írja le saját szavaival a használt algoritmust és indokolja annak hatékonyságát.

(2p.)

- Írja meg a megtervezett algoritmusnak megfelelő C/C++ programot.

(8p.)