

**Examenul național de bacalaureat 2026 Proba E. d) INFORMATICĂ**  
**Limbajul C/C++**

Simulare

*Filieră teoretică, profil real, specializare matematică-informatică / matematică-informatică intensiv informatică*  
*Filieră vocațională, profil militar, specializare matematică-informatică*

- 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.
- În grafurile din cerințe oricare arc/muchie are extremități distincte și oricare două arce/muchii diferă prin cel puțin una dintre extremități.

**I. TÉTEL** **(20 pont)**

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

1. Az **a** és **b** változók **int** típusúak, míg a **c** és **d** változók **float** típusúak. Az alábbi értékadás utasítások közül, melyik szintaktikailag helytelen?

- a. **c=d+2\*a;**                      b. **a=(d<=c);**                      c. **d=2-c%2;**                      d. **c=sqrt(b\*b);**

2. Az **f** alprogram, az alábbi módon van értelmezve. Adjátok meg, hogy mit jelenít meg az alábbi meghívás: **f(5);**

```
void f (int x)
{
    while (x>1)
    {
        x=x-1;
        f(x-1);
    }
    cout<<x;
}
```

- a. **11111111**                      b. **01010101**                      c. **10101101**                      d. **00000000**

3. A backtracking módszert használva, előállítjuk az összes lehetőséget arra, hogy a **10**-es számot prímszámok összegeként írjuk fel. A generált megoldások sorrendben a következők: **2+2+2+2+2**, **2+2+3+3**, **2+3+5**, **3+7**, **5+5**. Ugyanezt a módszert használva, előállítjuk az összes lehetőséget arra, hogy a **9**-es számot prímszámok összegeként írjuk fel. Add meg a harmadik generált megoldást.

- a. **2+7**                      b. **2+2+5**                      c. **2+2+2+3**                      d. **3+3+3**

4. Az alábbi **9** csomópontú gyökeres fában, a csomópontokat **1-től 9-ig** számozzuk, és a fa ősvektora: **T=(8,7,6,6,7,7,8,0,8)**. Hány leszármazottja van a **7-es** csomópontnak?

- a. **3**                      b. **2**                      c. **1**                      d. **5**

5. Egy irányítatlan gráfnak **9** csomópontja van és élei: **[1,3]**, **[2,1]**, **[3,2]**, **[4,5]**, **[5,6]**, **[6,4]**, **[7,9]**, **[8,7]**, **[9,8]**. Legtöbb hány élet adhatunk hozzá a gráfhoz úgy, hogy a kapott gráf összefüggő komponenseinek száma megegyezzen az eredeti gráf összefüggő komponenseinek számával.

- a. **9**                      b. **36**                      c. **0**                      d. **27**

**1. Adott a mellékelt algoritmus pszeudokódban.**

Az  $x\%y$  az  $x$  természetes szám,  $y$  nem nulla természetes számmal való osztási maradéka, és a  $[z]$  a  $z$  valós szám egész része.

a. Mit ír ki az algoritmus, ha a beolvasott értékek, ebben a sorrendben: **12 és 48. (6p)**

b. Adjatok meg két, különböző számpárt a **[10,100]** intervallumból, amelyeket beolvasva az **a** és **b** változókba, az algoritmus a **9-es** értéket jeleníti meg. **(6p)**

c. Írjátok meg az adott algoritmusnak megfelelő C/C++ programot. **(10p)**

d. Írjátok az adott algoritmussal egyenértékű pszeudokód algoritmust, amelyben a második ismétlő struktúrát egy hátul tesztelő ismétlő struktúrával helyettesítitek. **(6p)**

```
beolvas a,b (nem nulla
természetes számok)
p←1
d←2
amíg d≤a és d≤b végezd el
    n←0
    amíg a%d=0 és b%d=0 végezd el
        p←p*d
        n←n+1
        a←[a/d]
        b←[b/d]
    amíg_vége
    ha n%2=1 akkor
        p←[p/d]
    ha_vége
    d←d+1
amíg_vége
kiír p
```

2. Írjátok le egy **7** csomópontú, **1-től 7-ig** számozott, összefüggő, irányítatlan gráf szomszédsági listáját, tudva azt, hogy a gráf Euler gráf, de nem Hamilton gráf. **(6p)**

3. Az **i** és **j** változók egész típusúak, és az **a** változó egy **7** sort és **6** oszlopot tartalmazó kétdimenziós tömböt tárol, amelynek sorait és oszlopait **0-tól** sorszámozzuk, és kezdetben az összes eleme nulla.

Írjátok le az alábbi utasítássorozatot, megfelelő utasításokkal helyettesítve a kipontozott részt, amelyek között legtöbb négy értékadás van úgy, hogy az így kapott utasítássorozat végrehajtása során az **a** változó a mellékelt tömböt tárolja.

```
for(i=0;i<7;i++)
    for(j=0;j<6;j++)
        ..... (6p)
```

1	2	3	4	5	6
12	11	10	9	8	7
13	14	15	16	17	18
24	23	22	21	20	19
25	26	27	28	29	30
36	35	34	33	32	31
37	38	39	40	41	42

**III-as TÉTEL**

(30 pont)

**1. (10p.)** A **sub** alprogramnak két paramétere van: **v**, amelyen keresztül egy legfeljebb 111 elemű, egész számokat tartalmazó egydimenziós tömböt kap, valamint **n**, a tömb elemeinek száma. Az alprogram visszatéríti a tömb legnagyobb "meleg" számának pozícióját. Ha több ilyen szám is van, akkor a szám első előfordulásának pozícióját adja vissza. Ha nem létezik ilyen szám, az alprogram a **-1-et** adja vissza.

Egy egész szám "meleg", ha az első és az utolsó számjegyének paritása megegyezik.

Írjátok meg C/C++ nyelven a **sub** alprogram teljes definícióját.

**Példa:** ha **n = 12** és **v = (21, 412, 2, 2026, 208, 523, 7, 128, 3174, 2026, 46, 8)**, akkor a visszaadott érték **4** (a sorozat "meleg" számai: **412, 2, 2026, 208, 523, 7, 2026, 46, 8**. A legnagyobb érték **2026**. Ennek első előfordulása a **4.** pozíción van).

**2. (10p.)** Írjátok egy C/C++ programot, amely beolvas a billentyűzetről egy legfeljebb 500 karakterből álló szöveget, amely csak az angol ábécé betűiből és szóközőkből áll, és amelyben a szavakat egyetlen szóköz választja el. A szövegre a következő feldolgozást kell alkalmazni: minden lehetséges

szóhosszra ( $lu = 1, 2, 3, \dots$ , a szövegben előforduló leghosszabb szó hosszáig) az első  $lu$  hosszúságú szót fel kell cserélni az utolsó, ugyanilyen hosszúságú szóval; a második  $lu$  hosszúságú szót a hátulról második ilyen szóval, és így tovább, ha léteznek ilyen szavak.

A program jelenítse meg a képernyőn, a végrehajtott összes csere után kapott szöveget.

**Példa:** ha a beolvasott szöveg: **Ana a mers azi cu bicicleta patru kilometri spre mare**, akkor a képernyőn megjelenik: **azi a mare Ana cu kilometri patru bicicleta spre mers** ha pedig a beolvasott szöveg: **Mihai are trei pisici**, akkor az eredmény: **Mihai are trei pisici**

3. A **bac.txt** állomány első sorában két természetes szám található: **n** ( $2 \leq n \leq 100000$ ) és **p** ( $2 \leq p \leq 1000$ ), a második sorban pedig egy  $n$  elemű természetes számokból álló sorozat, amelyek az  $[1; 30000]$  intervallumból lehetnek. Határozzátok meg, hogy hány különböző, különböző pozíciókon levő elem-pár létezik a sorozatban, amelyek összege  $p$ -nek többszöröse. Tervezzetek egy, a futási idő szempontjából hatékony algoritmust!

**Példa:** ha  $n=5$ ,  $p=3$  és a sorozat **9 2 4 7 3**, az eredmény **3**, mivel azok a párok, amelyek összege **3** többszöröse: **(9,3)**, **(2,4)** és **(2,7)**; ha  $n=4$ ,  $p=6$  és a sorozat **1 1 6 4**, az eredmény **0**, mert nincs olyan pár, amelynek az összege **6** többszöröse lenne; ha  $n=8$ ,  $p=4$  és a sorozat **9 22 3 2 6 4 8 14**, az eredmény **8**, mivel az összege **4** többszöröse a következő pároknak: **(9,3)**, **(22,2)**, **(22,6)**, **(22,14)**, **(2,6)**, **(2,14)**, **(6,14)** és **(4,8)**.

a. Írjátok le saját szavaitokkal a megtervezett algoritmust, indokolva annak hatékonyságát. **(2p.)**

b. Írjátok meg a megtervezett algoritmusnak megfelelő C/C++ programot. **(8p.)**