

Examenul național de bacalaureat 2025

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

Variantă 4

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.

- Adja meg az intervallumot, amelybe az **x** egész változóba tárolt érték tartozik, akkor és csakis akkor, ha a mellékelt C/C++ kifejezés értéke 1.
`!(x<=2020) && !(x>2025)`
a. (2020, 2025] b. (2020, 2025) c. [2020, 2025] d. (2021, 2025)
- A **k** változó egész típusú, és az **s** egy maximum 20 karaktert tartalmazó karakterlánc tárolására alkalmas. Adja meg a **k** változó értékét a mellékelt kód végrehajtása után.
`strcpy(s, "calculator");
k=strchr(s, s[1])-strchr(s, s[3]);`
a. 9 b. 3 c. 2 d. 1
- Egy táncverseny egy szakaszára 6 pár jutott tovább, különböző betűjelekkel ellátva az {**A**, **B**, **C**, **D**, **E**, **F**} rendezett halmazból, minden pár egy táncot mutat be. A backtracking módszert használva, előállítjuk az összes lehetséges sorrendet a versenyen való fellépésre úgy, hogy az **A** pár lépjen fel először és az **F** pár utoljára. Az első három előállított megoldás: (**A**, **B**, **C**, **D**, **E**, **F**), (**A**, **B**, **C**, **E**, **D**, **F**), (**A**, **B**, **D**, **C**, **E**, **F**). Adja meg az utolsó megoldást.
a. (**A**, **D**, **B**, **C**, **E**, **F**) b. (**A**, **E**, **D**, **B**, **C**, **F**) c. (**A**, **E**, **D**, **C**, **B**, **F**) d. (**A**, **F**, **E**, **D**, **C**, **B**)
- Egy 6 csúcsú irányított gráf, csúcsait 1-től 6-ig számozzuk, élei (3, 1), (4, 1), (4, 2), (4, 3), (5, 6), (6, 4). Adja meg azon csúcsok darabszámát, amelyek kifoka egyenlő a befokával.
a. 1 b. 2 c. 3 d. 4
- Egy 50 csúcsú fában, a csúcsokat 1-től 50-ig számozzuk, a gyökér az 1-es számú csúcs, és minden más **i** csúcsának apja az $\lfloor i/2 \rfloor$ számmal jelölt csúcs. Adja meg annak az elemi láncnak a hosszát, amelynek egyik végpontja a 32-es, a másik végpontja pedig a 23-as.
a. 5 b. 7 c. 9 d. 11

II. TÉTEL

(40 pont)

- Adott a mellékelt algoritmus pszeudokódban.
Az **a%b** az **a** természetes számnak a **b** nullától különböző természetes számmal való osztási maradékát és **[c]** a **c** valós szám egész részét jelöli.
a. Írja le, mi kerül kiírásra az algoritmus végrehajtása nyomán, ha a beolvasott számok, ebben a sorrendben 9767 és 8204. (6p.)
b. Ha az első beolvasott szám 2025, adjon meg két különböző értéket a **[10, 99]** intervallumból, amelyet az **y**-ba olvashatunk úgy, hogy mindegyik esetén, az algoritmus által kiírt üzenet a **DA** legyen, egy szám értékkel kiegészítve. (6p.)
c. Írja meg az adott algoritmusnak megfelelő C/C++ programot. (10p.)
d. Írjon az adott algoritmussal egyenértékű pszeudokód algoritmust, amelyben az első **amíg...végezd el** szerkezetet helyettesíti egy hátul tesztelő ismétlődő szerkezettel. (6p.)

```
beolvas x,y
(különböző természetes számok)
cx←0; cy←0
amíg x≥10 vagy y≥10 végezd el
    amíg x+y≠0 végezd el
        cx←cx+x%10; x←[x/10]
        cy←cy+y%10; y←[y/10]
    x←cx; cx←0; y←cy; cy←0
ha x=y akkor kiír "DA ",x
különben kiír "NU ",x," ",y
```

2. Adott az **f** alprogram mellékelve. Írja le az **f(3, 9)** és az **f(1, 1000)** eredményeit. (6p.)
- ```
int f(int x, int y)
{ if(x*5>y/5) return x;
 return f(x*5, y/5);
}
```

3. A **b** változó több adatot tartalmaz egy ékszerről: súlyát grammban megadva, és a nemesfémét, amelyből gyártották (név és egy gramm ilyen fém ára). Az ékszer súlya és a fém egy grammjának ára természetes számok az **[1, 10<sup>4</sup>]**, intervallumból, míg a fém neve egy legfeljebb 30 karakterből álló karakterlánc. Tudva, hogy az alábbi C/C++ kifejezések értékei a fém neve, amelyből az ékszer készült, illetve ennek ára, adj meg egy **bi\_juterie** nevű struktúra értelmezését, amely a fent leírt adatok tárolására alkalmas, és értelmezze megfelelőképpen a **b** változót.

**b.metal.denumire**      **b.greutate\*b.metal.pret**

(6p.)

### III. TÉTEL

(30 pont)

1. Egy gazdának egy téglalap alakú földje van, amelyet azonos területű parcellákra akar bontani úgy, hogy a parcellák száma és területe is természetes számok legyenek, a parcellák száma páros legyen és szigorúan kisebb mint egy parcella területe. A **teren** alprogramnak két paramétere van, **x** és **y**, amelyen keresztül két természetes számot kap a **[2, 100]** intervallumból, a föld méterben megadott hossza és szélessége. Az alprogram írja ki a föld összes lehetséges felosztási módját, minden megoldás a képernyő egy külön sorába kerüljön, két szám formájában, amelyeket a **parcele de arie** szöveg és a szükséges szóközők válasszanak el, ahol az első szám a parcellák számának, a másik egy parcella területének felel meg, akár a példában. Ha nincs egyetlen megoldás sem, az alprogram írja ki a képernyőre a **nu exista** üzenetet. Adj meg az C/C++ alprogram teljes leírását.

**Példa:** ha **x=6** és **y=8**, az alprogram a mellékelt értékeket írja a képernyőre, nem feltétlenül ebben a sorrendben (a földet feloszthatjuk 3 darab 16 területű parcellára és 8 darab 6 területű parcellára, de ezek nem felelnek meg az elvárásoknak)

```
2 parcele de arie 24
4 parcele de arie 12
6 parcele de arie 8
```

(10p.)

2. Írjon egy C/C++ programot, amely két természetes számot olvas be a **[2, 10<sup>2</sup>]** intervallumból az **m**-et és **n**-et, majd egy kétdimenziós, **m** soros és **n** oszlopos tömb elemeit, amelyek különböző természetes számok a **[0, 10<sup>9</sup>]** intervallumból. A program módosítsa a memóriában a tömböt, lecserélve azon oszlop minden elemét, amely a tömb legkisebb elemét tartalmazza, azzal az értékkel, amely az utolsó sorban és utolsó oszlopban található. Az így kapott tömb a képernyőre íródik, a tömb sorai a képernyő egy-egy sorába, minden sor elemeit egy-egy szóközzel elválasztva.

**Példa:** ha **m=3**, **n=4** és a tömb

|    |    |   |    |
|----|----|---|----|
| 17 | 12 | 5 | 19 |
| 10 | 16 | 2 | 4  |
| 11 | 21 | 8 | 9  |

a módosított  
tömb

|    |    |   |    |
|----|----|---|----|
| 17 | 12 | 9 | 19 |
| 10 | 16 | 9 | 4  |
| 11 | 21 | 9 | 9  |

(10p.)

3. Egy napelempark esetén figyelik a **napi** és **össz**-energiatermelést, minden időszakra. A megfigyelt napok egymásutáni természetes számokkal vannak jelölve, kronológiai sorrendben, az 1-es naptól kezdődően. Egy időszak legalább két egymásutáni megfigyelési napból áll, és a rá vonatkozó össz-energiatermelést a megfelelő napi energiatermelések összegeként számítjuk ki. Egy **nap validált**, ha a napi termelés legalább egyenlő a napi határértékkel, a **minZ**-vel; egy **időszak validált**, ha a rá vonatkozó össztermelés legalább egyenlő a rövidtávú határértékkel, a **minP**-vel, minden napja validált és az időszak maximális hosszúságú ezzel a tulajdonsággal (nem adhatunk validált napot hozzá.)

A **bac.in** állomány legtöbb 10<sup>6</sup> természetes számot tartalmaz az **[1, 10<sup>3</sup>]** intervallumból: az első sorban a **minZ** és a **minP**, a leírt határértékeknek megfelelően, a második sorban a megfigyelés sorrendjében, az egymásutáni napok napi energiatermeléseit. Az azonos sorban levő számokat szóközőkkel választjuk el.

A képernyőre írásson, a bemeneti adatoknak megfelelő, minden validált termelési időszak esetén egy számhármast, az időszak első és utolsó napjának megfelelően, illetve az össz-energiatermelést. Minden számhármast értékeit a képernyő egy-egy sorába íratjuk, egy-egy szóközzel elválasztva, ha nem létezik ilyen időszak, a kiírt üzenet a **nu exista**. Tervezzon a futási idő és a felhasznált memória szempontjából hatékony algoritmust.

**Példa:** ha az állomány a mellékelt értékeket tartalmazza, a képernyőre kiírt értékek:

3 6 71

13 15 90

10 40

65 9 20 25 12 14 7 3 11 15 12 8 19 50 21

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

(2p.)

b. Írja meg a leírt algoritmusnak megfelelő C/C++ programot.

(8p.)