

Examenul național de bacalaureat 2024

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

Varianta 3

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 ítemek esetén, írja a vizsgalapra a helyes válasznak megfelelő betűt. Minden helyes válasz 4 pontot ér.

1. Adja meg azt a C/C++ kifejezést, amelynek értéke akkor és csak akkor **1**, ha az **x** és **y** egész változóban tárolt számok párosok.

a. **`x%2==0 && (y+1)%2!=0`**

b. **`(x-y)%2==0`**

c. **`(x+y)%2==0`**

d. **`x%2==y%2`**

2. Az **f** alprogram a mellékelt módon van definiálva. Adja meg, hogy mit ír ki az alábbi hivatkozás esetén.
f(2020, 0);

```
void f(int x, int y)
{ if (x<10) cout<<x; | printf("%d",x);
  else
  { f(x/10,y+1);
    cout<<x%10; | printf("%d",x%10);
  }
  cout<<y; | printf("%d",y);
}
```

a. **23020**

b. **2022100**

c. **02023210**

d. **23022100**

3. A backtracking módszert használva generáljuk a következő rendezett halmaz elemeinek összes permutációját: {**1, 2, 3, 4, 5, 6**}. Minden permutáció esetén az első három pozíción csak páros elemek vannak, valamint az utolsó három pozíción csak páratlan elemek vannak. Az első hat permutáció, ebben a sorrendben: (**2,4,6,1,3,5**), (**2,4,6,1,5,3**), (**2,4,6,3,1,5**), (**2,4,6,3,5,1**), (**2,4,6,5,1,3**), (**2,4,6,5,3,1**). Jelölje meg a hetedik generált permutációt.

a. **(4,2,6,1,5,3)**

b. **(4,2,6,1,3,5)**

c. **(2,6,4,1,3,5)**

d. **(2,4,6,5,3,2)**

4. Az **x** változó tárolja **20** féle csokoládé mindegyike esetén a következő adatokat: típus (**N** betű fekete csokoládé esetén, és **L** betű tejcsokoládé esetén) és a termék ára. Jelöljön meg egy kifejezést, amelynek értéke egyenlő a **11**-ik féle csokoládé típusával.

```
struct ciocolata
{ char tip;
  float pret;
}x[20];
```

a. **`x.ciocolata[10].tip`**

b. **`x.tip[10]`**

c. **`x[10].ciocolata.tip`**

d. **`x[10].tip`**

5. Egy **10** élel tartalmazó nem irányított gráfban két csúcs foka **0**, hat csúcs foka páratlan, és a többi csúcs foka nem nulla páros szám. Jelölje meg a gráf csúcsainak maximális számát.

a. **17**

b. **15**

c. **12**

d. **10**

II. TÉTEL

(40 punct)

1. Adott a mellékelt algoritmus pszeudokódban.

[c] a c valós szám egész részét jelöli.

- Írja le a kiírt értéket, ha a beolvasott számok ebben a sorrendben 5, 15, 27, 10, 1, 17. (6p.)
- Ha az n változóba beolvasott érték 2, írjon egy különböző értékekből álló értéksort a [0,10³] intervallumból, amelyeket a továbbiakban beolvashatunk úgy, hogy az algoritmus végrehajtása során a kiírt érték 4 legyen. (6p.)
- Írja meg az adott algoritmusnak megfelelő C/C++ programot. (10p.)
- Írjon az adott algoritmusmal egyenértékű pszeudokód algoritmust, amelyben megfelelően helyettesíti a minden...végezd el struktúrát egy amíg...végezd el ismétlődő struktúrával. (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
```

- Egy 1-től 8-ig sorszámozott, 8 csúcsot tartalmazó fát a (3,0,2,5,2,5,1,5) ösvektorral ábrázoljuk. Sorolja fel annak a leghosszabb útnak a pontjait a bejárás sorrendjében, amelynek kiindulópontja a gyökér. (6p.)
- Az i és j változók egész típusúak, és az a változó egy 9 sor és 9 oszlopot tartalmazó kétdimenziós tömböt tárol, amelynek sorait és oszlopait 0-tól sorszámozzuk, és kezdetben az összes eleme nulla. Írja le az alábbi utasítássorozatot, helyettesítve a pontozott részt megfelelő utasításokkal, 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<9; i++)
    for (j=0; j<9; j++)
        .....
```

(6p.)

```
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
```

III. TÉTEL

(30 punct)

- Egy természetes számot **major impar**-nak nevezünk, ha a páratlan valódi osztóinak összege szigorúan nagyobb, mint a páros valódi osztóinak összege. Egy szám valódi osztói az 1-től és önmagától különböző természetes osztói. **Példa:** 18 egy „major impar” szám (a páros valódi osztói 2, 6, és a páratlanok 3, 9, valamint 3+9>2+6). A **ma jImp** alprogramnak két paramétere van, a és b, amelyeken keresztül egy-egy természetes számot kap ($2 \leq a \leq b \leq 10^4$). Az alprogram visszatéríti a legkisebb „major impar” számot az [a, b] intervallumból, vagy 0-t, ha az intervallumban nincs ilyen szám. Írja le C/C++ nyelven a teljes alprogramot. **Példa:** ha a=16, b=30, akkor az alprogram által visszatérített érték 18. (10p.)
- Egy legtöbb 100 karaktert tartalmazó szövegben a szavak az angol ábécé betűit tartalmazzák, és egy-egy szóközzel vannak elválasztva. A szöveg legalább két szót tartalmaz. Írjon C/C++ programot, amely a billentyűzetről beolvas egy az előbbieken meghatározott típusú szöveget, és kiírja a képernyőre egy szóközzel elválasztva a DA üzenetet és egy n természetes számot, ha a szövegben található összes szó n betűt tartalmaz, vagy a NU üzenetet, ha az összes szó nem ugyanannyi betűt tartalmaz. **Példa:** ha a beolvasott szöveg **Ana are cel mai bun mar**, a képernyőre a következő lesz kiírva: **DA 3** valamint, ha a beolvasott szöveg **Ana are cel mai dulce mar**, a kiírt üzenet **NU** (10p.)
- Egy túraútvonal mentén egy sor turisztikai jelzést használunk, amelyeket ebben a sorrendben kell követni. Minden jelzés esetén ismert a tengerszint feletti magasság (méterben kifejezve), ahol el van helyezve. Egy útvonal esetén **lépcső**nek nevezünk egy egymás utáni jelzéseket tartalmazó sorozatot, amelyek tengerszint feletti magasságaik szigorúan növekvő sorrendben levő, egymás utáni számok. Egy lépcső legalább két jelzést tartalmaz, és a hossza egyenlő az öt alkotó jelzések számával. A **bac.txt** állomány legtöbb 10⁶ természetes számot tartalmaz a [10,10⁴] intervallumból, egy-egy szóközzel elválasztva, amelyek egy túraútvonal jelzéseinek tengerszint feletti magasságait jelentik, az előfordulásuk sorrendjében. Ki kell írni a képernyőre egy-egy szóközzel elválasztva szigorúan növekvő sorrendben a leghosszabb lépcső jelzéseinek magasságait ezen az útvonalon. Ha az útvonalon több ilyen lépcső is létezik, akkor csak az egyik jelzéseinek magasságait kell kiírni, és ha nincs lépcső, a képernyőre kiírt üzenet: **nu exista**. Tervezzon a futási idő és a felhasznált memória szempontjából hatékony algoritmust. **Példa:** ha az állományban levő számok 500 600 601 405 569 570 700 701 625 626 627 520 akkor a képernyőre 625 626 627 lesz kiírva
 - Írja le saját szavaival a használt algoritmust, és indokolja annak hatékonyságát. (2p.)
 - Írja meg a leírt algoritmusnak megfelelő C/C++ programot. (8p.)