

Examenul de bacalaureat național 2018
Proba E. d)
Informatică
Limbajul C/C++

Varianța 1

Filiera teoretică, profilul real, specializările: matematică-informatică
matematică-informatică intensiv informatică
Filiera vocațională, profilul militar, specializarea matematică-informatică

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- În rezolvările cerute, identificatorii utilizați 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).
- În programele cerute, datele de intrare se consideră corecte, validarea acestora nefiind necesară.
- În grafurile din cerințe oricare arc/muchie are extremități distincte și oricare două arce/muchii distincte diferă prin cel puțin una dintre extremități.

I. TÊTEL

(30 pont)

Az 1-es item esetén írja a vizsgalapra a helyes válasznak megfelelő betűt.

1. Az **x**, **y** és **z** változók egész típusúak és egy-egy nem nulla természetes számot tárolnak. Ha a mellékelt C/C++ kifejezés értéke 1, adja meg a növekvő sort a változók értékeiből, az alább megadott sorrendek közül. **(4p.)**
- a.** **x, y, z** **b.** **y, z, x** **c.** **z, x, y** **d.** **z, y, x**

2. Adott a mellékelt algoritmus pszeudokódban.

Jelölje **a%b** az **a** természetes szám **b** nem nulla természetes számmal való osztási maradékát és **[a]** az **a** valós szám egész részét.

- a)** Adja meg a kiírt számot, ha a beolvasott szám 10523. **(6p.)**
- b)** Írjon négy egész számot, amelyeket ha beolvasunk, az algoritmus elvégzése után a kiírt érték minden esetben 722. **(4p.)**
- c)** Írjon az adott algoritmussal egyenértékű pszeudokód algoritmust, amelyben az **ismeteld...ameddig** szerkezetet megfelelő módon helyettesít egy elől tesztelő ismétlődő szerkezettel. **(6p.)**
- d)** Írja meg az adott algoritmusnak megfelelő C/C++ programot. **(10p.)**

```
olvas n
    (egész szám)
m ← 0
p ← 1
x ← 0
    ha n < 0 akkor
        n ← -n
    ■
    ismételd
        c ← n % 10
        n ← [n / 10]
        ha c > m akkor
            m ← c
        ■
        x ← m * p + x
        p ← p * 10
    ameddig n ≠ 0
kiír x
```

II. TÉTEL

(30 pont)

Az 1-es és a 2-es itemek esetén írja a vizsgalapra a helyes válasznak megfelelő betűt.

1. Az `strlen("2018")` kifejezés értéke: (4p.)
a. 4 b. 5 c. 6 d. 7
2. Egy irányított gráf teljes, ha bármely i és j csúcsa között létezik vagy mindkét él, az (i, j) és (j, i) , vagy csak az egyik ezek közül.
Egy irányított gráfnak 5 csúcsa és 20 éle van. Ahhoz, hogy megkapjuk egy olyan részgráfját, amely két erősen összefüggő komponensből áll, ahol mindkét komponens teljes, és az egyiknek 3, a másiknak 2 csúcsa van, a minimálisan eltávolítható élek száma: (4p.)
a. 2 b. 3 c. 6 d. 10

Írja a vizsgalapra a következő feladatok megoldásait.

3. A mellékelt értelmezésben a `cat` és a `rest` mezők két természetes, nem nulla szám osztási hányadosát illetve maradékát tárolják.
- ```
struct impartire
{
 int cat;
 int rest;
}rezultat;
int x;
```

Írjon egy utasítássort, amelynek eredményeként a `rezultat` változóba kerül 2018-nak az `x` változóban tárolt értékkel való egész osztásából származó hányados és maradék, amennyiben az `x` nem nulla, ellenkező esetben a képernyőn jelenjen meg az `impartire nepermisa` üzenet. (6p.)

4. A mellékelt szomszédsági mátrix egy 8 csúcsból álló fát ír le, a csúcsok 1-től 8-ig vannak számozva. Soroljon fel három csúcsot, amelyet gyökérnek választhatunk úgy, hogy minden csúcsnak legfeljebb két közvetlen leszármazottja (gyereke) legyen. (6p.)

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

5. Írjon egy C/C++ programot, amely egy  $n$  ( $n \in [2, 10^2]$ ) természetes számot és egy  $n$  elemű, természetes, az  $[0, 10^4]$  intervallumban található számokból álló sorozatot olvas be a billentyűzetről. A program építsen fel a memóriában egy kétdimenziós tömböt,  $n$  sorral és  $n$  oszloppal, 0-tól sorszámozva ezeket úgy, hogy bármely páros sorszámú oszlopot lentől felfele, vagy a páratlan sorszámúakat fentről lefele bejárva, a beolvasott sorozatot kapjuk vissza, akárcsak a példában.  
A program írja ki a képernyőre a kért tömböt, a tömb minden sora a képernyő külön sorában jelenjen meg és a soron belüli elemeket egy-egy szóközzel elválasztva.

**Példa:** ha  $n=4$ , és a beolvasott számsor 7 2 5 3, a mellékelt tömböt kapjuk.

(10p.)

|   |   |   |   |
|---|---|---|---|
| 3 | 7 | 3 | 7 |
| 5 | 2 | 5 | 2 |
| 2 | 5 | 2 | 5 |
| 7 | 3 | 7 | 3 |

III. Tétel

(30 pont)

Az 1-es item esetén írja a vizsgalapra a helyes válasznak megfelelő betűt.

1. Adott a mellékelt  $f$  alprogram.  
Jelölje meg mi íródik ki az  
alábbi meghívás esetén.  
 $f(7)$ ;

(4p.)

```
void f(int n)
{ cout<<n%2; | printf("%d",n%2);
 if(n>=3) f(n-3);
}
```

a. 10

b. 010

c. 101

d. 1010

Írja a vizsgalapra a következő feladatok megoldásait.

2. A backtracking algoritmust használva előállítjuk az összes lehetséges megoldást, ahogy 5 különböző írószerről álló csomagot kialakíthatunk a {*stilou*, *pană*, *toc*, *creion*, *pensulă*}, halmazból, úgy hogy minden csomagban a *creion* megelőzi a *stilou*-t és a *pană*-t. Két csomag különböző, ha az elemek más sorrendbe vannak helyezve. Az első öt előállított megoldás, az előállítás sorrendjében (*toc*, *creion*, *stilou*, *pană*, *pensulă*), (*toc*, *creion*, *stilou*, *pensulă*, *pană*), (*toc*, *creion*, *pană*, *stilou*, *pensulă*), (*toc*, *creion*, *pană*, *pensulă*, *stilou*), (*toc*, *creion*, *pensulă*, *stilou*, *pană*). Adja meg a hatodik és a hetedik megoldást, az előállítás sorrendjében. (6p.)

3. Az *interval* alprogramnak egyetlen paramétere van, az *n*, amelyen keresztül egy természetes számot kap ( $n \in [3, 10^6]$ ). Az alprogram visszatéríti a legkisebb olyan *x* ( $n < x$ ) természetes számot, amely **NEM** prím, úgy hogy az  $[n, x]$  intervallumban egyetlen prímszám létezzen.

Írja le a teljes alprogramot.

**Példa:** ha  $n=8$ , az alprogram által visszatérített szám 12.

(10p.)

4. A mellékeltlen meghatározott sorozat első tagjai a következők (az *n* egy nem nulla természetes szám):

0, 3, 8, 15, 24, 35, 48, 63, 80 ....

$$f_n = \begin{cases} 0 & \text{ha } n=1 \\ 3 & \text{ha } n=2 \\ 2 \cdot f_{n-1} - f_{n-2} + 2 & \text{különb} \end{cases}$$

ha  $n=1$

ha  $n=2$

különb

A billentyűzetről két természetes számot olvasunk be, az *x*-et és *y*-t a  $[0, 10^9]$  intervallumból, amelyek a sorozatból két **egymás utáni** tag értékének felelnek meg ( $x < y$ ). Írassd ki a *bac.txt* szövegállományba, szigorúan csökkenő sorrendbe, egy-egy szóközzel elválasztva, a sorozat minden *y*-nél kisebb vagy vele egyenlő tagját.

Tervezzon a futási idő és a felhasznált memória szempontjából hatékony algoritmust.

**Példa:** ha a beolvasott számok

48 63

a *bac.txt* a következő számokat fogja tartalmazni:

63 48 35 24 15 8 3 0

a) Írja le saját szavaival a használt algoritmust, indokolva annak hatékonyságát

(2p.)

b) Írja meg a megtervezett algoritmusnak megfelelő C/C++ programot.

(8p.)