

Examenul național de bacalaureat 2024

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

Simulare

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 TETEL (20 pont)

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

1. Jelölje meg a mellékelt C/C++ kifejezés értékét. 6.3/20+24

- a. 0 b. 24 c. 24.315 d. 24.9

2. Az alábbi utasítássorozatban az összes változó egész típusú.

```
for(i=0; i<5; i++)  
{ for(j=0; j<5; j++)  
    if(j<i) cout<<.....<<" "; | printf("%d ", .....);  
    else cout<<2*i<<" "; | printf("%d ", 2*i);  
    cout<<endl; | printf("\n");  
}
```

0 0 0 0 0
0 2 2 2 2
0 2 4 4 4
0 2 4 6 6
0 2 4 6 8

Jelölje meg azt a kifejezést, amelyet a pontozott rész helyére írhatunk úgy, hogy a kapott utasítássorozat végrehajtás után a fenti ábrán szereplő értékeket írja ki, ebben a sorrendben.

- a. 2+j b. 2*j c. 2+i+j d. 2*i*j

3. Egy egydimenziós tömb elemei ebben a sorrendben, (3, 6, 10, 16, 20, 24, 49). Annak ellenőrzésére, hogy a tömbben létezik-e az **x=22** értékű elem, a bináris keresés módszerét alkalmazzuk. Adja meg sorban azon elemeket, amelyekkel össze lesz hasonlítva az **x** értéke a megadott módszer alkalmazása során.

- a. 16, 6, 24 b. 16, 24, 20 c. 49, 24, 20 d. 3, 10, 20, 49

4. Jelölje meg a mellékelt C/C++ kifejezés értékét.

floor(sqrt(24))

- a. 25 b. 24 c. 5 d. 4

5. Az alábbi utasítássorozatban az összes változó egész típusú, és természetes számokat tárolnak.

Jelölje meg azt a kifejezést, amelyet a pontozott rész helyére írhatunk úgy, hogy a kapott utasítássorozat végrehajtás után az **y** változó értéke 2^n ($n \in [0, 15]$) legyen.

```
y=1; x=2;  
while(n>0)  
    if(.....){ x=x*x; n=n/2; }  
    else { y=y*x; n=n-1; }
```

- a. **n%2==0** b. **n/2>=1** c. **n%4>2** d. **n/4<2**

II TETEL (40 pont)

1. 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. Adja meg az algoritmus végrehajtása során kiírt értéket, ha a beolvasott szám **32625**. (6p.)

b. Írjon két számot a **[10, 100]** intervallumból, amelyeket az **n** változóba beolvashatunk úgy, hogy mindkettőre az algoritmus végrehajtása után a kiírt érték **2** legyen. (6p.)

c. Írja le az algoritmusnak megfelelő C/C++ programot. (10p.)

d. Írjon pszeudokódban a megadottal egyenértékű algoritmust, amelyben a **második amíg ... végezd el** struktúrát megfelelőképpen helyettesíti egy hátul tesztelő ismétlődő struktúrával. (6p.)

```
beolvas n  
(nem nulla természetes szám)  
s←0; d←2  
amíg d*d≤n végezd el  
    p←0  
    amíg n%d=0 végezd el  
        n←[n/d]; p←1  
    s←s+d*p; d←d+1  
ha n≠1 akkor  
    s←s+n  
kiír s
```

2. Egy sorozatot úgy kapunk, hogy növekvő sorrendben összefésüljük az **A**=(2,3,7,11,20) és **B**=(3,5,6,10,16,24) egydimenziós tömbök elemeit balról jobbra haladva, és minden lépésben egy **(x,y)** értékpárt hasonlítunk össze, amelyben **x** egy elem az **A**-ból, és **y** egy elem a **B**-ből. Írja le azokat a számpárokat, amelyeket az ötödik és hatodik lépésben hasonlítunk össze, a feldolgozás sorrendjében. (6p.)
3. Egy videojáték esetén tároljuk a kategóriát, amelyhez tartozik (**A** betű, ha akciójáték, valamint **S** betű, ha stratégia játék) és a nehézségi szintek számát (természetes szám az **[1,10]** intervallumból). A **c1** és **n1** változók tárolják egy videojáték kategóriáját, valamint a nehézségi szintek számát, és a **c2** és **n2** változók tárolják egy másik videojáték kategóriáját és a nehézségi szintjeinek számát. Deklarálja megfelelően a **c1** és **c2** változókat, és tudva azt, hogy a két játéknak különböző számú nehézségi szintjük van, írjon egy C/C++ utasítássorozatot, amely végrehajtása után kiírja a képernyőre a több nehézségi szinttel rendelkező játék adatait: a kategória teljes megnevezését (**ACTIUNE** vagy **STRATEGIE**), amelyet egy szóköz követ, majd a játék nehézségi szintjeinek számát.
Példa: ha **c1** értéke az **A** betű, és **n1=2**, valamint **c2** értéke az **S** betű és **n2=4**, akkor a képernyőre a következő lesz kiírva:
STRATEGIE 4 (6p.)

III TÉTEL

(30 pont)

1. Beolvassunk egy **n** ($n \in [1, 10^4]$) természetes számot, és ki kell írni azt a számot, amelyet úgy kapunk, hogy megduplázzuk az **n** összes páratlan számjegyét, vagy a -1 értéket, ha a számnak nincs egyetlen páratlan számjegye sem. Írja le pszeudokódban a megfelelő algoritmust.
Példa: ha **n=3361**, a kiírt érték 3333611. (10p.)
2. Egy **sablon** egy egész számokból álló sorozat, amelyben a szigorúan negatív elemeket **generikus elemeknek** nevezzük.
Egy számítógép egy ilyen sablon alapján generál egy sorozatot úgy, hogy a sablon minden generikus elemét helyettesíti egy adott természetes számokat tartalmazó **lista** egy olyan elemével, amely esetén az egyesek számjegye egyenlő.
Példa: A -20 generikus elem helyettesíthető a 0 vagy 50 értékekkel, ha a lista (24, 2, 0, 36, 201, 50).
Írjon egy C/C++ programot, amely a billentyűzetről beolvasson egy **nL** ($nL \in [1, 100]$) természetes számot, és egy **L** egydimenziós tömb **nL** elemét, majd egy **nS** ($nS \in [1, 100]$) természetes számot, és az **S** egydimenziós tömb **nS** elemét. Az **L** egy az előzőekben leírt típusú lista, és az elemei természetes számok a $[0, 10^3]$ intervallumból, valamint **S** egy az előzőekben leírt típusú sablon, és az elemei egész számok a $[-10^9, 10^9]$ intervallumból. A program a sablonnak megfelelő tömböt a memóriában átalakítja úgy, hogy ennek tartalma egy lehetséges sorozat legyen, amelyet a számítógép generálhat a megadott sablon és lista alapján. Ezek után egy-egy szóközzel elválasztva írja ki a képernyőre az így kapott tömb elemeit, vagy az **imposibil** üzenetet, ha nem lehet generálni egy ilyen sorozatot.
Példa: ha **nL=7**, és a lista **L**=(24, 2, 0, 36, 21, 104, 50), valamint **nS=8**
a következő sablon esetén **S**=(-16, 0, 3, -452, -10, -974, 10, -20)
a generált sorozat (36, 0, 3, 2, 0, 24, 10, 0)
vagy (36, 0, 3, 2, 0, 104, 10, 50)
stb., valamint az **S**=(-17, 0, 3, -452, -10, -975, 10, -20) sablon esetén
a kiírt üzenet **imposibil** (10p.)
3. Egy sort **api** típusúnak nevezünk, ha minden elem előfordulási számának paritása megegyezik az elem paritásával.
A **bac.in** állomány legfeljebb 10^6 természetes számot tartalmaz a $[1, 10^3]$ intervallumból, egy-egy szóközzel elválasztva. A képernyőre ki kell írni a **DA** üzenetet, ha a sor **api** típusú, vagy a **NU** üzenetet ellenkező esetben. Tervezzon egy a futási idő szempontjából hatékony algoritmust.
Példa: ha az állomány tartalma 2 27 8 2 27 2 2 14 14 8 27 a képernyőre kiíródik a **DA**
(a páros 2 elem 4-szer jelenik meg, és 4 szintén páros, a 8 és 14 elemek 2-szer jelennek meg, a 2 ugyancsak páros, valamint a 27-es elem 3-szor jelenik meg, és a 3 szintén páratlan szám).
a. Írja le saját szavaival a használt algoritmust, és indokolja annak hatékonyságát. (2p.)
b. Írja meg a tervezett algoritmusnak megfelelő C/C++ programot. (8p.)