

**Examenul de bacalaureat național 2020**  
**Proba E. d)**  
**Informatică**  
**Limbajul C/C++**

Testul 13

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

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 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 la 5 feladatok esetében, csak a helyes válasz betűjelét kell a vizsgalpra írni. Minden helyes válasz 4 pontot ér.

- Adja meg azt a C/C++ kifejezést, amelynek az értéke 1 akkor és csakis akkor, ha az  $n$  egész típusú változóban tárolt természetes szám osztható 4-el és 5-el is.
  - $!(n \% 4 != 0 \ || \ n \% 5 != 0)$
  - $!(n / 4 == 1 \ || \ n / 5 != 0)$
  - $n \% 4 == 0 \ \&\& \ !(n \% 5 == 0)$
  - $n / 4 == 0 \ \&\& \ !(n / 5 == 0)$
- Mellékelten definiáltuk az  $f$  alprogramot. Az  $f(200200)$  értéke:

```
int f (int x)
{ if (x>20) return 2*f(x/10);
  return 20;
}
```

  - 160
  - 202
  - 210
  - 320
- A backtracking módszert alkalmazva az  $\{5, 6, 7, 8\}$  halmazból generáljuk az összes legtöbb három számjegyből álló páratlan számot. Az első 8 generált szám sorrendben: 5, 55, 555, 557, 565, 567, 57, 575. Melyik lesz a 12. generált szám.
  - 65
  - 67
  - 587
  - 655
- Egy 6 csomóponttal rendelkező gyökeres fának, amelynek a csomópontjait 1-től 6-ig számoztuk, az ősvektora, az "apák" tömbje:  $(2, 8, 2, 9, 8, 9, 0, 7, 7, 9)$ . Határozzátok meg a fa „levél” csomópontjainak a számát.
  - 4
  - 5
  - 6
  - 7
- Adott egy 7 csomópontot és 20 élel tartalmazó irányítás nélküli gráf. Határozzátok meg legkevesebb hány csomópontot törölhetünk, ahhoz, hogy a létrejött részgráf teljes legyen.
  - 0
  - 1
  - 2
  - 3

**II. TÉTEL** **(40 pont)**

- Adott a mellékelt algoritmus pszeudokódban.**  
Az  $a \% b$  jelöli az  $a$  természetes szám  $b$  nem nulla természetes számmal való osztási maradékát, és  $[c]$  a  $c$  valós szám egész részét.
  - Írjátok le az algoritmus végrehajtását követően, milyen szám jelenik meg, ha a beolvasott számok rendre: 12345, 25, 2070, 49, 270135, 21. **(6p.)**
  - Ha  $n$  beolvasott értéke 100, akkor adjatok meg egy számsort a  $[10^3, 10^4)$  intervallumból, amelyet ha beolvasunk, az algoritmus végrehajtása után a kiírt érték 10 lesz. **(6p.)**
  - Írjátok az adott algoritmusnak megfelelő C/C++ programot. **(10p.)**
  - Írjátok meg azt az eredetivel egyenértékű algoritmust pszeudokódban, amelyik csak egy ismétlődő struktúrát tartalmaz **(6p.)**

```
beolvas n (természetes szám)
p ← 1; m ← 0; k ← 0
amíg n ≠ 0 végezd el
  beolvas x (természetes szám)
  i ← k
  amíg i ≠ 0 végezd el
    x ← [x/10]; i ← i - 1
  ha x = 0 akkor c ← n % 10
  különben c ← x % 10
  m ← c * p + m; n ← [n/10]
  p ← p * 10; k ← k + 1
kiír m
```

2. A `c` változó egy kör adatait tárolja: a kör sugarának hosszát illetve az `xOy` koordináta rendszerben a kör középpontjának valós koordinátáit (abszcissza és ordináta). Tudva, hogy az alábbi C/C++ kifejezések értékei olyan valós számok, amelyek a kör adatait tárolják, írjátok meg a `cerc` címkeű struktúra definícióját, úgyhogy, abban tárolhassuk a rögzített adatokat, és deklaráljátok helyesen a `c` változót.
- `c.raza`      `c.centru.x`      `c.centru.y`      (6p.)

3. Az `i` egész típusú változó, az `s` változó legtöbb 20 karakterből álló sort tárol. Írjátok le mi jelenik meg a képernyőn az utasítás sorozat végrehajtása után. (6p.)
- ```
strcpy(s, "stilou");
cout<<s+4<<endl; | printf("%s\n", s+4);
for(i=0; i<4; i++)
    s[i]=s[0]+(i-1)*(1-i%2)+3*(2*i/3-1)*(i%2);
s[4]='\0';
cout<<s; | printf("%s", s);
```

### III. TÉTEL

(30 pont)

1. A `putere` nevű alprogramnak három paramétere van:
- `n`, amelyen keresztül kap egy természetes számot az  $[2, 10^9]$  intervallumból;
  - `d` és `p`, amelyek keresztül szolgáltatja a `d` prím osztót, amelyek a legkisebb `p` hatványon szerepel az `n` prímtényezőkre való felbontásában; amennyiben több ilyen osztó is van, akkor ezek közül a legkisebbet adja vissza.
- Írjátok le a teljes alprogramot.
- Példa:** ha `n=10780`, akkor az alprogram hívása után `d=5` és `p=1` ( $10780=2^2 \cdot 5 \cdot 7^2 \cdot 11$ ). (10p.)
2. Írjátok egy C/C++ programot, amelyik beolvassa a billentyűzetről az `n` és `k` természetes számokat a  $[2, 20]$  intervallumból, és felépít a memóriában egy `n · k` sorból és `n` oszlopból álló kétdimenziós tömböt (a sorok és oszlopok számozása 1-től történik) úgy, hogy minden `i` oszlopban ( $i \in [1, n]$ ) egy olyan növekvő sorozat legyen, melyben az első elem az `i`, mindegyik érték pontosan `k`-szor szerepel, és két egymás melletti elem értéke azonos vagy egymásutáni.
- A program írja ki a képernyőre a felépített tömböt, a tömb minden sora a képernyő külön sorába kerüljön, egy sorban az értékeket egymástól egy szóköz válassza el.
- Példa:** ha `n=4` és `k=3`, akkor a képernyőn a mellékelt tömb jelenik meg. (10p.)
- |   |   |   |   |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| 1 | 2 | 3 | 4 |
| 1 | 2 | 3 | 4 |
| 2 | 3 | 4 | 5 |
| 2 | 3 | 4 | 5 |
| 2 | 3 | 4 | 5 |
| 3 | 4 | 5 | 6 |
| 3 | 4 | 5 | 6 |
| 3 | 4 | 5 | 6 |
| 4 | 5 | 6 | 7 |
| 4 | 5 | 6 | 7 |
| 4 | 5 | 6 | 7 |
3. Adott a következő sorozat 1, 2, 4, 8, 16, 32 . . . .
- a következő módon meghatározva:  $f_1=1, f_2=2, f_n=3 \cdot f_{n-1} - 2 \cdot f_{n-2}$  (ahol `n` egy természetes szám  $n \geq 3$ ):
- A billentyűzetről beolvassuk az `x` ( $x \leq 10^9$ ) természetes számot, a sorozat egy tagjának az értékét. Írjátok be a `bac.txt`, állományba csökkenő sorrendben, egymástól egy szóközzel elválasztva a sorozat azon elemeit, amelyek kisebbek, vagy egyenlők, mint az `x`. Tervezzen memóriahasználat és végrehajtási idő szempontjából hatékony algoritmust.
- Példa:** ha a beolvasott szám 16 akkor a `bac.txt` tartalma a következő 16 8 4 2 1
- a. Írjátok le saját szavaitokkal a tervezett algoritmust, és indokoljátok annak hatékonyságát. (2p.)
- b. Írjátok a tervezett algoritmusnak megfelelő C/C++ programot. (8p.)