

Examenul național de bacalaureat 2021

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

Testul 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 muchie are extremități distincte și oricare două muchii diferă prin cel puțin una dintre extremități.

SUBIECTUL I

(20 de puncte)

Az 1-től 5-ig számozott itemek esetén írja a vizsgalapra a helyes válasznak megfelelő betűt! Minden helyes válasz 4 pontot ér.

- Adja meg azt a C/C++ kifejezést, amelynek értéke **1** akkor és csakis akkor, ha az **x** egész típusú változóban tárolt érték eleme a $[-21, -2] \cup [2, 21]$ intervallumegyesítésnek!

a. <code>x>=-21 && x<=-2 && x>=2 && x<=21</code>
b. <code>!(x<-21 x>-2) !(x<2 x>21)</code>
c. <code>x>=-21 x<=-2 x>=2 x<=21</code>
d. <code>!(x<-21 && x>21 && x>-2 x<2)</code>
- Az **f** alprogram a mellékelt módon van értelmezve.
Határozza meg, mit ír ki az `f(54321)`; függvénymeghívás hatására!

<pre>void f (int x) { cout<<"*"; printf("*"); if(x>0) { cout<<x; printf("%d",x); f(x/100); } cout<<"/"; printf("/"); }</pre>	
a. <code>****554354321</code>	b. <code>****/5/543/54321/</code>
c. <code>*54321*543*5*////</code>	d. <code>/5/543/54321/</code>
- A backtracking módszert használva generáljuk az összes lehetőséget, ahogyan kiválaszthatunk négy bukaresti múzeumot az alábbi halmazból {Muzeul de Artă Veche Apuseană (**MAVA**), Muzeul Colecțiilor de Artă (**MCA**), Muzeul Căilor Ferate Române (**MCFR**), Muzeul Național al Hărtiilor și Cărții Vechi (**MNHCV**), Muzeul Național al Literaturii Române (**MNLR**), Muzeul Național Tehnic (**MNT**)}. Két megoldás különböző, ha legalább egy múzeumban különbözik. Az első öt megoldás a generálás sorrendjében: (**MAVA**, **MCA**, **MCFR**, **MNHCV**), (**MAVA**, **MCA**, **MCFR**, **MNLR**), (**MAVA**, **MCA**, **MCFR**, **MNT**), (**MAVA**, **MCA**, **MNHCV**, **MNLR**), (**MAVA**, **MCA**, **MNHCV**, **MNT**). Adj meg egy múzeumfelsorolást, amely megoldása lehet a generálásnak az adott sorrendben!

a. (MNHCV , MNLR , MCFR , MNT)	b. (MCA , MNHCV , MNLR , MNT)
c. (MCA , MCFR , MNHCV , MNLR , MNT)	d. (MAVA , MCA , MNHCV , MCFR)
- Egy 5 csomópontból álló nem irányított gráf szomszédsági mátrixa mellékelve van. Határozza meg, hogy hány élet kell törölnünk, hogy az adott részgráf fa legyen.

	0	1	1	0	1
	1	0	1	0	0
	1	1	0	1	1
	0	0	1	0	1
	1	0	1	1	0

a. 3	b. 4	c. 5	d. 6
------	------	------	------
- Adott egy 7 csomópontból álló nem irányított gráf, csomópontjai 1-től 7-ig sorszámozva a következő élekkel: {**[2,3]**, **[2,5]**, **[2,6]**, **[3,6]**, **[4,7]**, **[5,1]**, **[6,5]**}. Határozza meg azon algráfjainak számát, amelyben az élek {**[2,3]**, **[2,6]**, **[3,6]**} halmazból lehetnek.

a. 5	b. 6	c. 7	d. 8
------	------	------	------

SUBIECTUL al II-lea

(40 de puncte)

- A mellékelt algoritmus pszeudokódban van ábrázolva.**
Az $a \% b$ az a természetes számnak a b nullától különböző természetes számmal való osztási maradékát és a $[c]$ a c valós szám egész részét jelöli.

 - Írja le az algoritmus által kiírt értéket, ha a **250887** számot olvassuk be! **(6p.)**
 - Írjon két számot, amelyeket beolvasva az algoritmus végrehajtása után a kiírt érték megegyezik a beolvasottal mindkét esetben! **(6p.)**
 - Írjon az adott algoritmusnak megfelelő C/C++ programot! **(10p.)**
 - Írjon az adott algoritmussal egyenértékű pszeudokód algoritmust helyettesítve a második ismétlődő struktúrát egy más típusú ismétlődő struktúrával! **(6p.)**
- ```

beolvas n
(nem nulla természetes szám)
x ← 0; m ← 0; p ← 1
amíg x < 10 végezd el
 cn ← n
 amíg cn ≠ 0 végezd el
 c ← cn % 10; cn ← [cn / 10]
 ha c = x akkor
 m ← c * p + m; p ← p * 10
 x ← x + 2
kiír m

```
- Az **m** változó egyidejűleg tárolja egy múzeum kiállításainak számát (**numar**), ( $[3, 10^2]$  intervallumban levő természetes szám) valamint minden kiállítás nevét (**nume**) és gyűjteményét (**colectie**). A **nume** és **colectie** mezők maximum 20 karaktert tartalmazó karaktersorok. Az alábbi C/C++ kifejezések értékei sorrendben: a kiállítások száma, a harmadik kiállítás neve valamint a gyűjtemény, amelyhez tartozik. Írja meg a **muzeu** címkével rendelkező struktúra értelmezését, amelyben tárolhatjuk a múzeum adatait, valamint deklarálja megfelelő képpen az **m** változót. **(6p.)**

**m.numar      m.exponat[2].nume      m.exponat[2].colectie**
  - Az **s** változó maximum 20 karakterből álló karaktersort tárol.

Írja le mi íródik ki az alábbi programrész végrehajtása során! **(6p.)**

```

strcpy(s, "muzeu");
s[0] = s[0] + 1;
cout << s[1] << s[0] << endl; | printf("%c%c\n", s[1], s[0]);
strcpy(s, "muzeu" + 2);
cout << s; | printf("%s", s);

```

**SUBIECTUL al III-lea**

**(30 de puncte)**

- A **suma** alprogramnak egyetlen **n** paramétere van, amelyen keresztül egy természetes számot kap ( $n \in [1, 10^6]$ ). Az alprogram visszatéríti az **n** szám pozitív osztóinak összegét, amelyek nem prímek. Írja meg az alprogram teljes definícióját!  
**Például:**  $n=12$ -re az alprogram visszatérítési értéke **23** ( $23=1+4+6+12$ ). **(10p.)**
- Egy előadásteremnek több sora van és minden sorban ugyanannyi hely van. Egy személy több helyet is lefoglalhat, de legtöbb egy helyet egy sorban. A terem konfigurációját egy kétdimenziós tömbbel szemléltetjük, ahol az első **oszlopnak** az első sor szék felel meg, valamint az utolsó **oszlopnak** az utolsó sor szék. Az üres helyeket 0-val jelöltük, a lefoglalt helyeket pedig nem nulla természetes számokkal. Egy személy által lefoglalt helyeket ugyanazzal a számmal jelöltünk, különböző személyek által lefoglalt helyeket pedig különböző számokkal.  
Írjon C/C++ programot, amely beolvassa a billentyűzetről az **m** és **n** természetes számokat a  $[2, 20]$  intervallumból, valamint az **m · n** természetes számot a  $[0, 400]$  intervallumból! A **m** sorral és **n** oszloppal rendelkező kétdimenziós tömb a terem konfigurációját jelöli. A program ellenőrzi, hogy van-e olyan személy, aki az első illetve az utolsó sorban is foglalt helyet és kiírja a képernyőre azokat a számokat, amelyekkel számoztuk ezeket a helyeket, mint a példában.

A számokat bármilyen sorrendben kiírhatjuk egy-egy szóközzel elválasztva. Ha nincsenek ilyen helyek, akkor a képernyőn a **nu exista** üzenet jelenjen meg!  
**Példa:** ha **m=5**, **n=4**, akkor a mellékelt tömb esetén a kiírt értékek nem föltétlen ebben a sorrendben **6 8** **(10p.)**

|   |   |   |   |
|---|---|---|---|
| 3 | 7 | 0 | 0 |
| 0 | 0 | 5 | 8 |
| 6 | 2 | 0 | 1 |
| 0 | 3 | 0 | 6 |
| 8 | 0 | 6 | 4 |
- Két természetes számot **z-barátoknak** nevezünk, ha ugyanaz a tízeseinek számjegye.  
A **bac.in** szöveges állomány legtöbb  $10^6$  természetes számot tartalmaz a  $[10, 10^9]$  intervallumból, szóközzel elválasztva. A kérés az, hogy írja ki a sorozat azon pozícióit, amelyeken levő számok a legtöbb z-barát számot alkotnak az előttük levő értékekkel. A kiírt számok szóközzel legyenek elválasztva! Tervezzon egy hatékony algoritmust futási idő szempontjából!  
**Például:** ha a szövegállomány tartalma: **726 358 98 157 20 49 128 879 659 271**

a képernyőre kiíródik a 7 9 szám (a 128, illetve a 659 megfelelnek a feltételnek).

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

**(2p.)**

**b.** Írja le a tervezett algoritmusnak megfelelő C/C++ programot!

**(8p.)**