

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

Testul 15

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.

**SUBIECTUL I**

(20 punct)

Írja a vizsgalpra az 1 és 5 közötti feladatokra adott helyes válaszoknak megfelelő betűt. Mindegyik helyes válasz 4 pontot ér.

1. A következő C/C++  
`(x>=16) && !( x<17 || x>19) && (x<=20)`  
kifejezés értéke akkor és csakis akkor 1, ha az **x** változó a következő intervallumból kap értéket:  
a. [16,18]                      b. [17,19]                      c. [18,20]                      d. [19,20]
2. A backtracking módszerét használva generáljuk a fajtiszta galambok összes lehetséges elrendezési módozatait a madárházakba a következő halmazból {**creți, iacobini, jucători, rotați, toboșari**}. Két megoldás különböző, ha a fajták sorrendje különböző. Az első négy megoldást a következő sorrendbe kapjuk: (**creți, iacobini, jucători, rotați, toboșari**), (**creți, iacobini, jucători, toboșari, rotați**), (**creți, iacobini, rotați, jucători, toboșari**), (**creți, iacobini, rotați, toboșari, jucători**). Jelöljük be az utolsóelőtti megoldást.  
a. (**toboșari, rotați, creți, iacobini, jucători**)  
b. (**toboșari, rotați, creți, jucători, iacobini**)  
c. (**toboșari, rotați, jucători, creți, iacobini**)  
d. (**toboșari, rotați, jucători, iacobini, creți**)
3. Az **A** és **B** változók mindegyike egy pozitív koordinátát tárol az xOy koordinátarendszerben (**x** abszcissza és **y** ordináta), egy szakasz két végpontja. Jelöljük be azt a C/C++ kifejezést, amelyik értéke akkor és csakis akkor 1, ha az egyik végpontja a szakasznak az xOy koordinátarendszer origójában van.  

	<pre>struct punct { int x,y; } A,B;</pre>
--	---

  
a. **(A.x+A.y) \* (B.x+B.y) == 0**                      b. **(A(x)+A(y)) \* (B(x)+B(y)) == 0**  
c. **(x.A+y.A) \* (x.B+y.B) == 0**                      d. **punct.A(x+y) \* punct.B(x+y) == 0**
4. Egy gyökeres fában egy csúcspont az **x** szinten található, ha a csúcspont és a gyökér közti elemi lánc hosszúsága **x**. A 0. szinten egy csúcspont (gyökér) található. Adott egy 8 csúcspontból áll gyökeres fa, 1-től 8-ig sorszámozva, a következő élekkel [1,3], [1,7], [1,8], [2,4], [3,5], [3,6], [4,5]. Tudva azt, hogy az adott fa gyökere a 7-es sorszámu csúcspont, jelöljük be a fa szintjeinek a számát.  
a. 3                      b. 4                      c. 6                      d. 7
5. Adott egy 5 csúcspontból álló irányított gráf, a csúcspontok 1-től 5-ig vannak sorszámozva, a következő élekkel (1,4), (3,5), (5,1), (5,2). Jelöljük be, hogy hány élt kell a gráfhoz adni, hogy a kapott gráf erősen összefüggő legyen.  
a. 1                      b. 2                      c. 3                      d. 4

**SUBIECTUL al II-lea**

**(40 punct)**

**1. Adott a mellékelt pszeudokód.**

- a. Mit fog kiírni az algoritmus a végrehajtás során, ha a beolvasott  $n$  érték 5. **(6p.)**
- b. Adjon meg két értéket a  $[10, 10^2)$  intervallumból amelyeket, ha beolvas az algoritmus az  $n$  változóba, akkor 14-t fog kiírni. **(6p.)**
- c. Írja meg az adott algoritmusnak megfelelő C/C++ programot. **(10p.)**
- d. Írjon pszeudokódban az adott algoritmussal egyenértékű algoritmust, amelyben a **minden...végezd el** ismétlődő struktúrát egy másik ismétlődő struktúrával helyettesíti. **(6p.)**

```

beolvas n (természetes szám)
nr ← 0
minden i ← n, 1, -1 végezd el
  x ← 0; y ← 1
  minden j ← 1, i végezd el
    r ← 2*x - y; x ← y; y ← r
  ha y > 0 akkor
    nr ← nr + 1
kiír nr

```

2. Adott a mellékelt  $f$  alprogram. Adjon meg két természetes számot az  $[1, 10]$  intervallumból, amelyeket ha az  $x_1$  és  $x_2$  változók megkapnak, akkor az  $f(10, x_1)$  5-t térít vissza az  $f(x_2, 10)$  pedig 1-et. **(6p.)**

```

int f(int x, int y)
{
  if (x > y) return x*y + f(x-y, y);
  if (x < y) return y*x + f(x, y-x);
  return 1;
}

```

3. Az  $i$  és  $j$  egész típusú változók, az  $a$  egy kétdimenziós tömb 4 sorral és 5 oszloppal, 0-tól sorszámozva, az elemei egész számok, kezdeti értékük nulla. Anélkül, hogy más változókat használna az adott változókon kívül, írjon egy utasítássorozatot, amelynek a végrehajtásának eredményeként az  $a$  változó a mellékelt táblázatot fogja tárolni. **(6p.)**

1	5	9	13	17
2	6	10	14	18
3	7	11	15	19
4	8	12	16	20

**SUBIECTUL al III-lea**

**(30 punct)**

1. A `divPrimMax` alprogramnak két paramétere van:

- $n$ , egy természetes szám ( $n \in [2, 10^9]$ );
- $p$ , amelyen keresztül visszaadja a legnagyobb prím osztóját az  $n$ -nek.

Írjuk meg a teljes alprogramot.

**Példa:** ha  $n=2000$ , akkor a hívás eredményeképpen  $p=5$  lesz, mert  $2000=2^4 \cdot 5^3$ .

**(10p.)**

2. Egy maximum 100 karakterből álló szövegben a szavak az angol ábc kisbetűiből állnak és egy szóközzel vannak elválasztva egymástól. Írjunk egy C/C++ programot, amely beolvassa a billentyűzetről a fent említett szöveget és kiírja a képernyőre azoknak a szavaknak a számát, amelyekben a magánhangzók és a mássalhangzók száma egyenlő. Magánhangzóknak tekintjük a következő betűket:  $a, e, i, o, u$ .

**Példa:** a következő szövegre

cuvantul consoane are un numar de patru vocale si patru consoane

kiírja, hogy: 6.

**(10p.)**

3. Beolvassunk a billentyűzetről két természetes számot  $p_1$ -t és  $p_2$ -t az  $[1, 81]$  intervallumból, írja ki a **bac.out** állományba az összes olyan számot, amely pontosan 7 számjegyű és az első két számjegy szorzata  $p_1$ , a következő 3 számjegy egyenlő egymással és az utolsó két számjegy szorzata  $p_2$ . A állományban a számok csökkenő sorrendbe kell, szerepeljenek, minden sorba egy szám kell, kerüljön. Tervezzon meg egy hatékony algoritmust a memória felhasználás és a futási idő szempontjából.

**Példa:** ha  $p_1=12$  és  $p_2=8$ , akkor 2633324 és 3400018 két szám a 160 számból, amelyek a kért tulajdonsággal rendelkeznek. ( $2 \cdot 6=3 \cdot 4=12$  és  $2 \cdot 4=1 \cdot 8=8$ ).

a. Írjon egy C/C++ programot a megtervezett algoritmusnak megfelelően.

**(8p.)**

b. Írja le saját szavaival a megtervezett algoritmust és indokolja meg ennek a hatékonyságát.

**(2p.)**