Examenul naţional de bacalaureat 2022
Proba E. d)
Informatică
Limbajul C/C++

Varianta 4

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)

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Indicați expresia C/C++ cu valoarea `2022`.

a. `4044/4/2` **b.** `4044/(4*2)` **c.** `1011*1+1` **d.** `1011*(1+1)`

2. Subprogramul `f` este definit alăturat. Indicați valoarea parametrului `x` pentru care `f(x,3)` are valoarea `2022`.

```
int f(int n,int p)
{ if(n==0) return 0;
  else if(p==0) return n*10+2;
       else return f(n/10,p-1);
}
```

a. `120234` **b.** `122023` **c.** `202345` **d.** `220223`

3. Utilizând metoda backtracking, se generează toate posibilitățile de a expune câte trei mașini produse în România, din gama Dacia, din mulţimea {`Dokker` (negru), `Duster` (negru), `Lodgy` (gri), `Logan` (alb), `Sandero` (roșu)}, astfel încât să nu fie expuse simultan două mașini de aceeași culoare, Sandero să NU apară pe prima poziție, iar Logan să NU fie înainte de Dokker sau Duster. Două soluții sunt distincte dacă au cel puțin o mașină diferită sau dacă ordinea mașinilor este diferită. Primele cinci soluţii generate sunt, în această ordine (`Dokker`, `Lodgy`, `Logan`), (`Dokker`, `Lodgy`, `Sandero`), (`Dokker`, `Logan`, `Lodgy`), (`Dokker`, `Logan`, `Sandero`), (`Dokker`, `Sandero`, `Lodgy`). Indicaţi penultima soluție, în ordinea generării acestora.

a. `Logan`, `Lodgy`, `Sandero` **b.** `Logan`, `Sandero`, `Lodgy`

c. `Lodgy`, `Dokker`, `Logan` **d.** `Lodgy`, `Logan`, `Dokker`

4. Variabilele `s` și `d` permit memorarea câte unui șir cu maximum `50` de caractere. Indicați șirul memorat prin intermediul variabilei `d` în urma executării secvenței alăturate.

```
strcpy(s,"info@rocnee.eu");
s[strlen(s)-3]='\0';
strcpy(d,strchr(s,'@')+1);
```

a. `inf@r` **b.** `info@` **c.** `rocnee` **d.** `rocnu`

5. Într-un arbore cu `7` noduri, numerotate de la `1` la `7`, nodul `1` este rădăcină și pentru orice nod numerotat cu `i` (`i∈[1,7]`), dacă se notează cu `F(i)` numărul de descendenți direcți („fii”) ai săi, atunci `F(i)=0`, dacă `i` este „frunză”, altfel `F(i)=i+1`. Indicați numărul maxim de „frați” ai nodului `7`.

a. `4` **b.** `3` **c.** `2` **d.** `1`

SUBIECTUL al II-lea (40 de puncte)

1. **Algoritmul alăturat este reprezentat în pseudocod.**
S-a notat cu **a%b** restul împărţirii numărului natural **a** la numărul natural nenul **b** şi cu **[c]** partea întreagă a numărului real **c**.

```
citeşte n (număr natural)
repetă
  x←n%100
  dacă x>1 atunci
    p←(x%2)*(x%3)*(x%5)*(x%7)
    dacă p≠0 atunci
      scrie x,' '
    ■
  ■
  n←[n/10]
până când n<10
```

a. Scrieţi ce se afişează în urma executării algoritmului dacă se citeşte numărul **1237518**. **(6p.)**

b. Scrieţi două numere din intervalul **[100,999]** care pot fi citite, astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, să se afişeze două numere. **(6p.)**

c. Scrieţi programul C/C++ corespunzător algoritmului dat. **(10p.)**

d. Scrieţi în pseudocod un algoritm, echivalent cu cel dat, înlocuind structura **repetă...până când** cu o structură repetitivă cu test inițial. **(6p.)**

2. Un graf neorientat cu **6** noduri, numerotate de la **1** la **6**, are muchiile **[1,2]**, **[1,3]**, **[1,4]**, **[1,5]**, **[2,3]**, **[2,4]**, **[2,5]**, **[3,5]**, **[4,6]**, **[5,6]**. Scrieți mulțimea muchiilor unui subgraf al acestui graf, care să fie complet și să aibă un număr maxim de noduri. **(6p.)**

3. Pentru o sală de spectacole se memorează următoarele informații specifice: numărul de locuri disponibile și prețul biletului pentru un loc. Variabilele **i** și **s** sunt întregi, iar variabila **t** memorează informațiile specifice pentru fiecare dintre cele **15** săli de spectacole dintr-un oraș.

```
struct sala
{ int nrLocuri;
  int pret;
} t[15];
```

Scrieți o secvență de instrucțiuni în urma executării căreia variabila **s** să memoreze suma totală obținută în urma vânzării biletelor pentru un festival de teatru care s-ar desfășura simultan în toate sălile, iar acestea ar fi ocupate complet. **(6p.)**

SUBIECTUL al III-lea (30 de puncte)

1. Subprogramul **patrate** are trei parametri:
• **n**, prin care primeşte un număr natural (**n∈[2,10⁹]**);
• **x** și **y**, prin care furnizează câte un număr natural cu proprietatea că **x²·y²=n** și **2≤x<y** sau valoarea **0**, prin fiecare dintre aceștia, dacă nu există două astfel de numere. Dacă sunt mai multe astfel de valori, se furnizează cele corespunzătoare unei valori minime a lui **x**.
Scrieţi definiţia completă a subprogramului.
Exemplu: pentru **n=400**, după apel, **x=2** şi **y=10**, iar pentru **n=16** sau **n=24**, după apel, **x=0** şi **y=0**. **(10p.)**

2. Suprafața unei scene a fost împărțită în zone dispuse ca elementele unui tablou bidimensional cu număr egal de linii și coloane. Fiecare instrument are alocată o zonă și este identificat printr-un număr natural. Dirijorul hotărăște eliminarea tuturor instrumentelor plasate pe diagonala principală a tabloului și rearanjarea celor păstrate, prin deplasarea unora dintre acestea cu câte o poziție spre stânga, astfel încât să nu existe zone intermediare libere, ca în exemplu.
Scrieţi un program C/C++ care citeşte de la tastatură numere naturale din intervalul **[2,10²]**, și anume **n** şi elementele unui tablou bidimensional cu **n** linii şi **n** coloane, reprezentând numerele de identificare ale instrumentelor, în ordinea plasării lor inițiale pe scenă. Programul modifică apoi tabloul în memorie corespunzător hotărârii dirijorului și afișează pe ecran tabloul obținut, fiecare linie a tabloului pe câte o linie a ecranului, elementele de pe aceeași linie fiind separate prin câte un spațiu.

Exemplu: pentru **n=4** şi tabloul

20	30	80	52
71	93	64	51
62	81	55	49
90	65	32	22

se obține tabloul

30	80	52
71	64	51
62	81	49
90	65	32

(10p.)

3. Numim **secvență progresivă** a unui șir crescător de numere naturale un subșir al acestuia, format din termeni aflați pe poziții consecutive în șirul dat, cu proprietatea că fiecare termen apare în subșir de un număr de ori egal cu valoarea sa. Lungimea secvenței este egală cu numărul de termeni ai acesteia.
Fișierul **bac.txt** conține un șir crescător de cel mult **10⁶** numere naturale din intervalul [**1**,**10⁶**], astfel încât orice termen al șirului apare de un număr de ori cel mult egal cu valoarea sa. Numerele sunt separate prin câte un spațiu.
Se cere să se afișeze pe ecran lungimea maximă a unei secvențe progresive din șirul aflat în fișier. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al memoriei utilizate.
Exemplu: dacă fișierul conține numerele **1 2 2 3 4 4 4 4 6 6 6 6 6 6 7 7 7 8 8 8 8 8 8 8 8** atunci pe ecran se afișează valoarea **10**.
a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
b. Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**