

Examenul de bacalaureat național 2020

Proba E. d)

Informatică

Limbajul C/C++

Testul 10

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.
- Identifierii utilizati 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

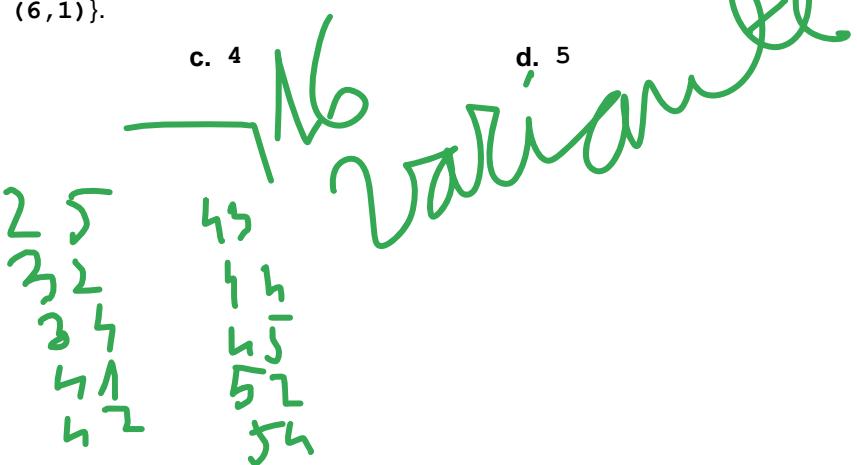
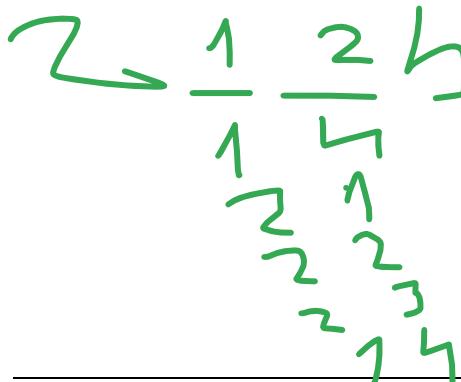
(20 de puncte)

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.

- Variabila **x** este de tip întreg și memorează un număr nenul. Indicați expresia C/C++ cu valoarea 1 dacă și numai dacă 20 este multiplu al lui **x**.
a. $20 / (20/x) == 0$ b. $20 / (20\%x) == 0$ c. $20 \% (20\%x) == 0$ d. $20 \% (20/x) == 0$
- Subprogramul **f** este definit alăturat.
Valoarea lui **f(19, 20)** este:

```
int f(int x, int y)
{
    if(x<=0) return y;
    if(y<=0) return x;
    return f(2*x-y, y+1);
}
```


a. 38 b. 24 c. -5 d. -7
- Utilizând metoda backtracking se generează toate posibilitățile de a forma siruri de către 4 animale din mulțimea {cal, câine, papagal, porumbel, ponei}, astfel încât pe oricare două poziții alăturate să nu se afle animale din submulțimea {cal, papagal, ponei}. Primele opt soluții generate sunt, în această ordine: (cal, câine, cal, câine), (cal, câine, cal, porumbel), (cal, câine, câine, cal), (cal, câine, câine, câine), (cal, câine, câine, papagal), (cal, câine, câine, porumbel), (cal, câine, câine, ponei), (cal, câine, papagal, câine). Indicați numărul de soluții generate care au pe prima poziție un câine și pe ultima poziție un porumbel.
a. 20 b. 18 c. 16 d. 14
- Un arbore cu rădăcină are 8 noduri, numerotate de la 1 la 8, și este reprezentat prin vectorul de „tați” (4, 4, 2, 7, 4, 8, 0, 7). Indicați frunzele arborelui.
a. 1, 3, 5, 6 b. 1, 2, 5 c. 7, 8 d. 3
- Un graf orientat cu 6 vârfuri, numerotate de la 1 la 6 are mulțimea arcelor {(1, 4), (1, 6), (3, 1), (3, 4), (3, 6), (6, 1)}. Indicați numărul de subgrafuri ale sale, pentru fiecare dintre acestea mulțimea arcelor fiind {(1, 6), (3, 1), (3, 6), (6, 1)}.
a. 2 b. 3 c. 4 d. 5



SUBIECTUL al II-lea

(40 de puncte)

1. **Algoritmul alăturat este reprezentat în pseudocod.**
 S-a notat cu **a** restul împărțirii numărului natural **a** la numărul natural nenul **b** și cu **a $\leftarrow\rightarrow b$** operația prin care se interschimbă valorile variabilelor **a** și **b**.
- a. Scrieți ce se afișează în urma executării algoritmului dacă se citesc, în această ordine, numerele **16** și **7**. **(6p.)**
- b. Dacă pentru **x** se citește numărul **10**, scrieți două valori care pot fi citite pentru variabila **y** astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, să se afișeze numărul **90**. **(6p.)**
- c. Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**
- d. Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind adekvat structura **repeta...până când** cu o structură repetitivă cu test inițial. **(6p.)**
2. Variabila **s** memorează simultan următoarele date despre fiecare dintre cele 20 de spectacole din cadrul unei stagiuni: un cod, numărul de actori și vârstele acestora. La un spectacol sunt maximum 10 actori. Expresiile C/C++ de mai jos au ca valori numere naturale și reprezintă codul primului spectacol, numărul de actori care joacă în acest spectacol, respectiv vârsta celui de al 5-lea astfel de actor. Scrieți definiția unei structuri cu eticheta **spectacol**, care permite memorarea datelor despre un spectacol, și declarați corespunzător variabila **s**.
s[0].cod s[0].nrActori s[0].varsta[4] **(6p.)**
3. Variabilele **i** și **j** sunt de tip întreg, iar variabila **a** memorează un tablou bidimensional cu **4** linii și **5** coloane, numerotate începând de la **0**, cu elemente numere întregi, inițial toate nule. Fără a utiliza alte variabile decât cele menționate, scrieți o secvență de instrucțiuni astfel încât, în urma executării acesteia, variabila **a** să memoreze tabloul alăturat. **(6p.)**
- | | | | | |
|----|----|----|----|----|
| 5 | 4 | 3 | 2 | 1 |
| 10 | 9 | 8 | 7 | 6 |
| 15 | 14 | 13 | 12 | 11 |
| 20 | 19 | 18 | 17 | 16 |

SUBIECTUL al III-lea

(30 de puncte)

1. Subprogramul **produs** are doi parametri:
- **n**, prin care primește un număr natural ($n \in [0, 10^9]$);
 - **p**, prin care furnizează produsul cifrelor pare distincte din scrierea acestuia, sau **-1** dacă nu există astfel de cifre.
- Scrieți definiția completă a subprogramului.
Exemplu: dacă **n=1622325**, după apel **p=12** ($12=6 \cdot 2$), iar dacă **n=122325**, după apel **p=2**. **(10p.)**
2. Într-un text cu cel mult 10^2 caractere cuvintele sunt formate din litere mici ale alfabetului englez și sunt separate prin câte un spațiu. Scrieți un program C/C++ care citește de la tastatură un text de tipul menționat, pe care îl modifică în memorie, inserând cuvântul **succes** între oricare două cuvinte ale sale care se termină cu aceeași literă. Cuvântul inserat este separat prin câte un spațiu de cuvintele vecine. Textul transformat este afișat pe ecran, iar dacă nu există perechi de astfel de cuvinte, se afișează pe ecran mesajul **nu există**.
Exemplu: dacă textul citit este **testez validez utilizez date corecte acum** se obține textul **testez succes validez utiliză date succes corecte acum** **(10p.)**
3. Un sir format din cel puțin trei termeni formează o progresie aritmetică de ratie **r** dacă diferența dintre oricare termen al acestuia și cel aflat pe poziția consecutivă în sir este egală cu **r**.
 Fișierul text **bac.txt** conține un sir de cel puțin trei și cel mult 10^6 numere întregi din intervalul $[-10^8, 10^8]$. Numerele sunt separate prin câte un spațiu. Se cere să se afișeze pe ecran ratia unei secvențe din sir cu număr maxim de termeni, secvență care formează o progresie aritmetică. Dacă există mai multe astfel de secvențe de lungime maximă se afișează ratia cea mai mare, iar dacă nu există nicio astfel de secvență, se afișează pe ecran mesajul **nu există**. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.
Exemplu: dacă fișierul conține numerele **4 9 14 19 18 17 8 5 3 1 -1 -3 -5 -7** pe ecran se afișează valoarea **-2** (corespunzătoare secvenței **5 3 1 -1 -3 -5 -7**).
 a. Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**
 b. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**