

p=1;

$70 \% 2 == 0 \Rightarrow$

$x = 70 / 2 = 45$

$p = 2$

$45 \% 2 != 0 \Rightarrow$ iesim din while

$s = s + 2 = 2$

$f = f + 1 = 3$

$35 > 1 \Rightarrow$

$p = 0$

$35 \% 3 != 0 \Rightarrow s = 2 ; f = 4$

$35 > 1$

$p = 0$

$35 \% 4 != 0 \Rightarrow s = 2 ; f = 5$

$35 > 1$

$p = 5$

$35 \% 5 == 0 \Rightarrow p = 1 ; x = 35 / 5 = 7$

$7 \% 5 != 0 \Rightarrow s = 3 ; f = 6 ;$

$7 > 1$

$7 \% 6 != 0 \Rightarrow s = 3 ; f = 6 ;$

$7 > 1$

$7 \% 7 == 0 \Rightarrow p = 1 ; x = 1 ;$

$1 \% 7 != 0 \Rightarrow s = 4 ; f = 8$

$1 > 1$ fals

iesim din while

se afiseaza 4

b)

- b)** Scrieți o valoare de 3 cifre care poate fi citită pentru x , astfel încât valoarea afișată să fie 6. **(4p.)**

Raspuns: **400**

c)

- c) Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int x;
6      cin >> x;
7      int s = 0;
8      int f = 2;
9      while(x>1){
10         int p=0;
11         while(x%f==0){
12             x=x/f;
13             p=p+1;
14         }
15         s=s+p;
16         f=f+1;
17     }
18     cout << s;
19 }
20
```

d)

- d) Scrieți valorile naturale din intervalul [7,28] care pot fi introduse pentru variabila x , astfel încât, după executarea programului, valoarea afișată să fie 1. (6p.)

Raspuns: 7 11 13 17 19 23 (Numerele prime din intervalul [7,28])

Subiectul II

ex.1

1. Se consideră graful neorientat cu mulțimea nodurilor $\{1, 2, 3, 4, 5, 6, 7, 8\}$ și mulțimea muchiilor $\{[1, 2], [2, 3], [2, 4], [4, 7], [2, 6], [1, 5], [5, 6], [6, 8], [7, 8]\}$. Pentru a transforma graful într-un arbore, putem elimina: (4p.)
- a. muchiile $[1, 5]$ și $[1, 2]$
 - b. muchia $[5, 6]$
 - c. nodul 3
 - d. muchiile $[2, 6]$ și $[4, 7]$

Raspuns: d)

Explicatie:

Subiectul II, ex.1

Un arbore este un graf aciclic și complet conex (adică e o un singur element)

Se observa că graful are porțiuni ciclice ($L1 = [1, 2, 6, 5]$) și ($L2 = [2, 4, 7, 8, 6]$)

În primul rând, pentru ca un graf să fie arbore, nr. de muchii trebuie să fie egal cu nr. de noduri - 1 (8)

Astfel, variantele b) și c) pica.

Verificăm varianta a):

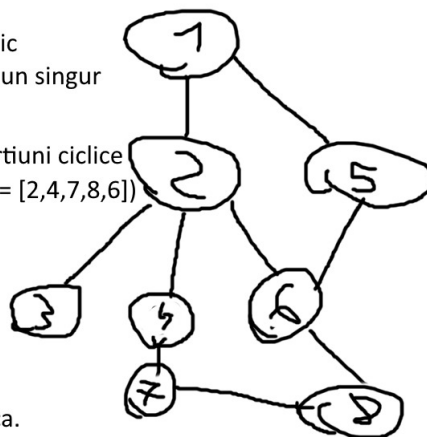
Dacă eliminăm muchiile $[1, 5]$ și $[1, 2]$, vom avea 2 elemente conexe (nodul 1 devine element conex cu el însuși deoarece nu mai este legat de alte noduri prin cel puțin o muchie!!!)

Deci singurul răspuns viabil este d).

Hai să vedem de ce acest răspuns este corect:

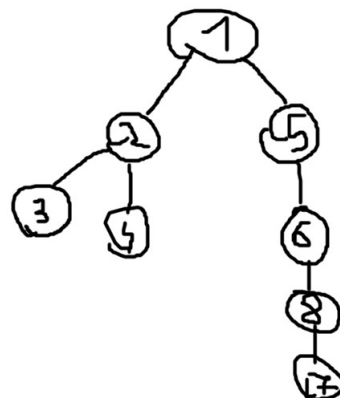
Răspunsul d) este corect deoarece, prin stergerea muchiei $[2, 6]$, scăpăm de porțiunea ciclică $L1$.

De asemenea, prin stergerea muchiei $[4, 7]$ scăpăm de porțiunea ciclică $L2$! Observați graful după ce ștergem aceste muchii:



Varianta 35

**raspuns:
d)**



Nicio porțiune ciclică, așa este?

De asemenea, graful are un singur element conex (el însuși lol), nu ca în cazul variantelor b) și c).

După cum am spus, varianta a) pica deoarece rămânem cu o porțiune ciclică!!

ex.2

2. Se consideră o listă alocată dinamic care are cel puțin 10 elemente și fiecare element al listei memorează în câmpul `info` un număr întreg, iar în câmpul `next` adresa elementului următor în listă sau `NULL` dacă nu există un element următor.

Variabila `p` memorează adresa de început a listei, iar variabila `aux` este de același tip cu `p`. Dacă în urma executării secvenței alăturate de program, variabila `p` are valoarea `NULL`, atunci:

(4p.)

```
while (p!=NULL && p->info%5!=0)
{ aux=p;
  p=aux->next;
  delete p; | free(p);
}
```

- a. toate numerele din listă sunt divizibile cu 5 b. doar primul element din listă este divizibil cu 5
- c. în listă nu s-a memorat niciun număr divizibil cu 5 d. doar ultimul element memorat în listă este divizibil cu 5

Raspuns: c)

ex.3

3. Ce se va afișa în urma executării secvenței alăturate, în care variabila `c` memorează un șir cu cel mult 20 de caractere, iar variabila `i` este de tip întreg?

(6p.)

```
char c[]="tamara";
for(i=0;i<3;i++)
  c[i]=c[i+1];
cout<<c; | printf("%s",c)
```

`c[]="tamara"`

`i = 0`

`c[0] = c[1] => t se transforma in a => c[]="aamara"`

`i = 1`

`c[1] = c[2] => a se transforma in m => c[]="ammara"`

`i = 2`

`c[2] = c[3] => m se transforma in a => c[]="amaara"`

La final se afiseaza amaara

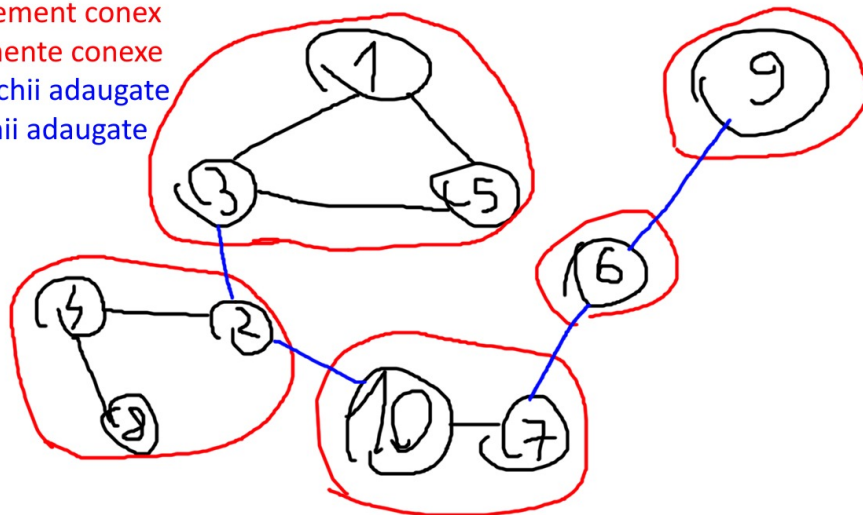
ex.4

4. Un graf neorientat cu 10 noduri, numerotate de la 1 la 10, este reprezentat cu ajutorul listelor de adiacență alăturate. Câte componente conexe are graful și care este numărul minim de muchii ce trebuie adăugate pentru ca graful să fie conex? (6p.)

1:3,5	6:
2:4	7:10
3:1,5	8:4
4:2,8	9:
5:1,3	10:7

Raspuns:

incercuit = element conex
total = 5 elemente conexe
albastru = muchii adaugate
total = 4 muchii adaugate



ex.5

5. Scrieți programul C/C++ care citește de la tastatură un număr natural n ($n \leq 50$) și construiește în memorie o matrice cu n linii și n coloane, ale cărei elemente sunt numere întregi citite de la tastatură. Pentru fiecare coloană a matricei, în ordine, programul afișează pe ecran cel mai mic număr de pe respectiva coloană. Numerele afișate vor fi separate prin câte un spațiu. (10p.)

Exemplu: pentru $n=4$ și matricea alăturată, se vor afișa pe ecran valorile: -7 18 -10 2.

122	103	5	10
-7	18	-10	2
107	999	59	4
1	200	100	7

```

1  ex.5
2  #include <iostream>
3  using namespace std;
4  int n,m[50][50];
5  int main()
6  {
7      cin >> n;
8      for(int i=0;i<n;i++)
9          for(int j=0;j<n;j++)
10             cin >> m[i][j];
11     for(int i=0;i<n;i++){
12         int min_coloana = m[0][i];
13         for(int j=0;j<n;j++)
14             if(m[j][i]<min_coloana) min_coloana = m[j][i];
15         cout << min_coloana << " ";
16     }
17 }
18

```

Subiectul III

ex.1

1. Pentru generarea tuturor mulțimilor de câte 5 cifre, având la dispoziție cifrele de la 1 la 9, se poate utiliza un algoritm echivalent cu algoritmul de generare a: **(4p.)**
- a. permutărilor de 5 elemente b. submulțimilor mulțimii {1,2,3,4,5,6,7,8,9}
- c. combinațiilor de 9 elemente luate câte 5 d. aranjamentelor de 9 elemente luate câte 5

Raspuns: c)

ex.2 (confuzie maxima nu va stresati daca nu intelegeti nimic)

2. Subprogramul `f` este definit alăturat. Ce se va afișa în urma executării secvenței de mai jos, în care variabilele `a` și `b` sunt de tip întreg?
- ```
void f(int &a, int b)
{
 a=a-5; b=b-2;
 cout<<a<<b; | printf("%d%d",a,b);
}
```
- `a=3; b=9; f(b,a); f(b,b);`      **(6p.)**

Raspuns:

`a = 3`

`b = 9`

pentru simplitate inlocuiesc **a cu t** si **b cu k**

`f(k,t):`

`a = k-5 = 4`

`b = t - 2 = 2`

afiseaza 4 si 2

se salveaza in k variabila a din subprogramul f => `k = 4`

`f(k,k)`

`a = k - 5 = -1 ; b = -1 - 2 = -3;`

afiseaza -1 si -3

la final se afiseaza **4 2 -1 -3**

### ex.3

3. Fișierul text `numere.in` conține pe prima linie un număr natural  $n$  ( $0 < n < 1000$ ), iar pe a doua linie  $n$  numere naturale cu cel mult 9 cifre fiecare, despărțite prin câte un spațiu. Scrieți un program C/C++ care citește toate numerele din fișier și afișează pe ecran, separate prin câte un spațiu, numerele de pe a doua linie a fișierului, care încep și se termină cu aceeași cifră. (10p.)

**Exemplu:** dacă fișierul `numere.in` are  
conținutul alăturat, se numerele ce se vor afișa  
sunt: 55 3 101 7 2782

|                            |
|----------------------------|
| 9                          |
| 55 107 3 101 92 7 208 2782 |
| 80                         |

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 int n;
4 ifstream fin("numere.in");
5 int main()
6 {
7 fin >> n;
8 for(int i=0; i<n; i++){
9 int x=-1; int prima = -1; int ultima = -1;
10 fin >> x;
11 int nr = x;
12 ultima = x%10;
13 while(x/=10){
14 if(x/10==0){
15 prima = x%10;
16 break;}
17 }
18 if(nr/10==0) cout << nr << " ";
19 if(prima==ultima) cout << nr << " ";
20 }
21 }
22
```



#### ex.4

4. a) Scrieți definiția completă a unui subprogram `sum` care primește prin parametrul `x` un număr natural de cel mult 4 cifre și returnează suma divizorilor numărului `x`, diferiți de 1 și de el însuși.

**Exemplu:** dacă `x=10` se va returna valoarea 7 ( $7=2+5$ ). (4p.)

- b) Scrieți programul C/C++ care citește de la tastatură un număr natural `n` ( $0 < n < 100$ ), apoi `n` numere naturale (cu cel mult 4 cifre fiecare). Programul determină, folosind apeluri utile ale subprogramului `sum`, pentru fiecare număr natural citit, suma divizorilor săi proprii și afișează pe ecran sumele determinate, în ordinea crescătoare a valorilor lor, separate prin câte un spațiu. (6p.)

**Exemplu:** dacă `n=5` și numerele citite sunt 10 2 33 6 11

valorile afișate pe ecran vor fi: 0 0 5 7 14

deoarece suma divizorilor lui 10 este 7, suma divizorilor lui 2 este 0, suma divizorilor lui 33 este 14, suma divizorilor lui 6 este 5, suma divizorilor lui 11 este 0.

Raspuns:

a)

```
1 int sum(unsigned int x) {
2 int s = 0;
3 for(int i=2; i<=x/2; i++)
4 if(x%i==0)
5 s+=i;
6 return s;
7 }
8
```

b)

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 int n;
4 int sum(unsigned int x) {
5 int s = 0;
6 for(int i=2; i<=x/2; i++)
7 if(x%i==0)
8 s+=i;
9 return s;
10 }
11 int main()
12 {
13 cin >> n;
14 int v[100]={0};
15 for(int i=0; i<n; i++) {
16 cin >> v[i];
17 v[i]=sum(v[i]);
18 }
19 sort(v, v+n);
20 for(int i=0; i<n; i++)
21 cout << v[i] << " ";
22 }
23
```