



Lecții de pregătire la informatică Admitere 2019

Tema: Discutarea problemelor date
la ultimele sesiuni de admitere

Bogdan Alexe
bogdan.alexe@fmi.unibuc.ro

Cuprinsul lecției de azi

Enunțuri și rezolvări pentru subiectul de informatică de la:

1. Admiterea la INFO, MATE 2018 (fără CTI)
2. Admiterea la INFO, MATE 2017 (fără CTI)
3. Admiterea la INFO, MATE 2016 (fără CTI)
4. Admiterea la INFO, MATE 2015 (fără CTI)
5. Admiterea la INFO, MATE 2014 (fără CTI)
6. Admiterea la INFO, MATE 2013 (fără CTI)
7. Admiterea la INFO, MATE 2012 (fără CTI)

Subject INFO 2018

Enunț subiect INFO 2018

IV. Informatică.

- (a) Se citește un număr natural L ($20 \leq L \leq 1000$) și un sir de cel mult 10000 de caractere ce conține cuvinte despărțite între ele prin câte un spațiu. Fiecare cuvânt din sirul de caractere citit este format din cel mult L litere mari ale alfabetului englez. Să se scrie un program care afișează aceste cuvinte, în ordinea în care se citesc, pe linii de cel mult L caractere, astfel încât orice linie începe și se termină cu un cuvânt și oricare două cuvinte de pe aceeași linie sunt separate printr-un singur spațiu. Oricare linie este folosită la maxim, adică dacă un cuvânt are loc pe acea linie va fi pus acolo și nu va fi trecut pe linia următoare sau spart pe 2 linii.

Exemplu: se citește $L = 22$ și sirul de caractere PROBLEMA DE LA EXAMEN NU MI SE PARE FOARTE GREU DE REZOLVAT IN TEMPUL ACORDAT. Programul va afișa:

```
PROBLEMA DE LA EXAMEN
NU MI SE PARE FOARTE
GREU DE REZOLVAT IN
TEMPUL ACORDAT
```

- (b) Într-un text formatat pe linii ca la punctul (a), două spații sunt *conecțate* dacă se învecinează pe verticală sau pe diagonală. Pentru textul formatat mai sus avem mai multe exemple de spații *conecțate*: spațiul de pe poziția 9, linia 1, ce separă literele A și D este *conecțat* cu cel de pe poziția 9, linia 2; spațiul de pe poziția 15, linia 1, este *conecțat* cu cel de pe poziția 14, linia 2. Spațiul de pe poziția 3, linia 2, nu este *conecțat* cu niciun alt spațiu. Să se scrie un program care citește numerele naturale L, N și apoi un text formatat pe N linii de cel mult L caractere ca la punctul (a) și afișează mesajul DA dacă în tot textul există cel puțin o pereche de spații *conecțate*, altfel afișează mesajul NU.

Enunț subiect INFO 2018

- (c) În arta tipografică un *râu* este o însiruire de spații care se întinde pe verticală, pe liniile consecutive ale unui text. Mai precis, un *râu* este definit ca o secvență de cel puțin 2 spații în care oricare 2 spații de pe linii consecutive sunt conectate. Spre exemplu, pentru textul de la punctul (a), avem un *râu* de lungime 4 format din: spațiul de pe poziția 9, linia 1; spațiul de pe poziția 9, linia 2; spațiul de pe poziția 8, linia 3 și spațiul de pe poziția 7, linia 4. De remarcat, faptul că de pe poziția 17, linia 3, nu pornește nici un *râu* întrucât linia 4 se termină pe poziția 14. Să se scrie un program, cu o complexitate de timp cât mai bună, care citește numerele naturale L, N și apoi un text formatat pe N linii de cel mult L caractere ca la punctul (a) și afișează lungimea celui mai lung *râu* posibil, dacă acesta există sau mesajul NU, dacă nu există niciun *râu*.

Note:

1. Programele vor fi scrise într-unul dintre limbajele de programare studiate în liceu (Pascal, C, C++). Pentru fiecare soluție se vor descrie informal detaliile algoritmului folosit și ale implementării sub formă de program: semnificația variabilelor, a structurilor de date, a structurilor repetitive și a instrucțiunilor condiționale.
2. Programele vor folosi instrucțiunile de bază ale limbajului de programare ales, funcții din biblioteci de bază (inclusiv cele de intrare/ieșire), dar nu și alte funcții din biblioteci specializate.
3. Citirea datelor se poate face de la tastatură sau dintr-un fișier text. Afisarea se va face numai la monitor. Cele 3 subpuncte se pot rezolva independent, dar funcțiile descrise la un subpunkt pot fi folosite și la subpunctele următoare.

Barem subiect INFO 2018

IV. Informatică.	Oficiu	1 p
(a)	Citirea sirului de caractere	0,25 p
	Găsirea cuvintelor separate prin spațiu	0,75 p
	Distribuirea cuvintelor pe o linie cu cel mult L caractere	0,5 p
	Încărcarea liniei la maxim	0,5 p
	Cuvintele sunt conținute integral pe o linie	0,25 p
	Afișarea corectă a liniilor	0,25 p
	Corectitudinea limbajului	0,5 p
	Explicații	0,25 p
(b)	Utilizarea unei structuri de date pentru menținerea informației necesare	0,5 p
	Verificarea conectivității pe cazul general (cu 3 vecini)	0,5 p
	Verificarea conectivității pe cazurile particulare (cu 0, 1, 2 vecini)	0,5 p
	Tratarea cazului pentru DA	0,5 p
	Tratarea cazului pentru NU	0,5 p
	Corectitudinea limbajului	0,5 p
	Explicații	0,25 p
(c)	Corectitudinea soluției	1 p
	Optimalitatea soluției corecte $\mathcal{O}(L \cdot N)$	1 p
	Deducerea complexității soluției prezentate	0,25 p
	Explicații	0,25 p

Rezolvare subiect INFO 2018 – punctul a

Exemplu: se citește $L = 22$ și sirul de caractere PROBLEMA DE LA EXAMEN NU MI SE PARE FOARTE GREU DE REZOLVAT IN TIMPUL ACORDAT. Programul va afișa:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
1

Când încep să completez o nouă linie, ea este **vidă**, nu are nimic.

Rezolvare subiect INFO 2018 – punctul a

Exemplu: se citește $L = 22$ și sirul de caractere PROBLEMA DE LA EXAMEN NU MI SE PARE FOARTE GREU DE REZOLVAT IN TIMPUL ACORDAT. Programul va afișa:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	P	R	O	B	L	E	M	A													

Când încep să completez o nouă linie, ea este **vidă**, nu are nimic.

Primul cuvânt pe linia curentă întotdeauna are loc (are cel mult L litere și începe pe linie).
NU PUN SPAȚIU DUPĂ EL (vezi enunț).

afișează aceste cuvinte, în ordinea în care se citesc, pe linii de cel mult L caractere, astfel încât orice linie începe și se termină cu un cuvânt și oricare două cuvinte de pe aceeași linie sunt separate printr-un singur spațiu. Oricare linie este folosită la maxim, adică dacă un cuvânt

Dacă mai vreau să adaug un cuvânt la linia curentă ce condiție trebuie să pun?

CONDIȚIE: lungimea curentă a liniei + 1 + lungimea cuvântului de adăugat $\leq L$

Rezolvare subiect INFO 2018 – punctul a

Exemplu: se citește $L = 22$ și sirul de caractere PROBLEMA DE LA EXAMEN NU MI SE PARE FOARTE GREU DE REZOLVAT IN TIMPUL ACORDAT. Programul va afișa:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	P	R	O	B	L	E	M	A	D	E		L	A		E	X	A	M	E	N		

Rezolvare subiect INFO 2018 – punctul a

Exemplu: se citește $L = 22$ și sirul de caractere PROBLEMA DE LA EXAMEN NU MI SE PARE FOARTE GREU DE REZOLVAT IN TIMPUL ACORDAT. Programul va afișa:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	P	R	O	B	L	E	M	A	D	E		L	A		E	X	A	M	E	N		
2																						

Când încep să completez o nouă linie, ea este **vidă**, nu are nimic.

Primul cuvânt pe linia curentă întotdeauna are loc (are cel mult L litere și începe pe linie).
NU PUN SPAȚIU DUPĂ EL (vezi enunț).

Dacă mai vreau să adaug un cuvânt la linia curentă pun condiția:

CONDIȚIE: lungimea curentă a liniei + 1 + lungimea cuvântului de adăugat $\leq L$

Rezolvare subiect INFO 2018 – punctul a

Exemplu: se citește $L = 22$ și sirul de caractere PROBLEMA DE LA EXAMEN NU MI SE PARE FOARTE GREU DE REZOLVAT IN TIMPUL ACORDAT. Programul va afișa:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	P	R	O	B	L	E	M	A		D	E		L	A		E	X	A	M	E	N	
2	N	U		M	I		S	E		P	A	R	E		F	O	A	R	T	E		
3	G	R	E	U		D	E		R	E	Z	O	L	V	A	T		I	N			
4	T	I	M	P	U	L		A	C	O	R	D	A	T								

- (a) Citirea sirului de caractere 0,25 p
Găsirea cuvintelor separate prin spațiu 0,75 p
Distribuirea cuvintelor pe o linie cu cel mult L caractere 0,5 p
Încărcarea liniei la maxim 0,5 p
Cuvintele sunt conținute integral pe o linie 0,25 p
Afișarea corectă a liniilor 0,25 p
Corectitudinea limbajului 0,5 p
Explicații 0,25 p

```
solutieInfo.c x
1 #include <stdio.h>
2 #include <string.h>
3
4 int main(){
5
6     int L;
7     char sir[10000];
8
9     //citire date
10    printf("dati sirul: "); fgets(sir,10000,stdin);
11    printf("L = "); scanf("%d",&L);
12
13    int lungime_linie = 0;
14    char separatori[] = " ";
15    char* p = strtok(sir,separatori);
16    while(p)
17    {
18        if (lungime_linie==0)
19        {
20            printf("%s",p);//printeaza cuvant
21            lungime_linie = strlen(p);
22        }
23        else
24            if(lungime_linie + 1 + strlen(p) <= L) //conditie
25            {
26                printf(" %s",p); //printeaza spatiu + cuvant
27                lungime_linie = lungime_linie + 1 + strlen(p);
28            }
29        else
30        {
31            printf("\n"); //mergi pe o alta linie
32            printf("%s",p); //printeaza cuvant
33            lungime_linie = strlen(p);
34        }
35        p = strtok(NULL,separatori);
36    }
37    return 0;
38 }
```

Rezolvare subiect INFO 2018 – punctul b

- (b) Într-un text formatat pe linii ca la punctul (a), două spații sunt *conectate* dacă se învecinează pe verticală sau pe diagonală. Pentru textul formatat mai sus avem mai multe exemple de spații *conectate*: spațiul de pe poziția 9, linia 1, ce separă literele A și D este *conectat* cu cel de pe poziția 9, linia 2; spațiul de pe poziția 15, linia 1, este *conectat* cu cel de pe poziția 14, linia 2. Spațiul de pe poziția 3, linia 2, nu este *conectat* cu niciun alt spațiu. Să se scrie un program care citește numerele naturale L, N și apoi un text formatat pe N linii de cel mult L caractere ca la punctul (a) și afișează mesajul DA dacă în tot textul există cel puțin o pereche de spații *conectate*, altfel afișează mesajul NU.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	P	R	O	B	L	E	M	A		D	E		L	A		E	X	A	M	E	N	
2	N	U	M	I		S	E		P	A	R	E		F	O	A	R	T	E			
3	G	R	E	U		D	E	R	E	Z	O	L	V	A	T		I	N				
4	T	I	M	P	U	L		A	C	O	R	D	A	T								

Rezolvare subiect INFO 2018 – punctul b

- (b) Într-un text formatat pe linii ca la punctul (a), două spații sunt *conectate* dacă se învecinează pe verticală sau pe diagonală. Pentru textul formatat mai sus avem mai multe exemple de spații *conectate*: spațiul de pe poziția 9, linia 1, ce separă literele A și D este *conectat* cu cel de pe poziția 9, linia 2; spațiul de pe poziția 15, linia 1, este *conectat* cu cel de pe poziția 14, linia 2. Spațiul de pe poziția 3, linia 2, nu este *conectat* cu niciun alt spațiu. Să se scrie un program care citește numerele naturale L, N și apoi un text formatat pe N linii de cel mult L caractere ca la punctul (a) și afișează mesajul DA dacă în tot textul există cel puțin o pereche de spații *conectate*, altfel afișează mesajul NU.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	P	R	O	B	L	E	M	A		D	E		L	A		E	X	A	M	E	N	
2	N	U	M	I		S	E		P	A	R	E		F	O	A	R	T	E			
3	G	R	E	U		D	E	R	E	Z	O	L	V	A	T		I	N				
4	T	I	M	P	U	L		A	C	O	R	D	A	T								

Folosește o structură de date corespunzătoare pentru stocarea liniilor: matrice cu N linii sau matrice cu 2 linii (țin minte ultimele două linii citite) sau altceva.

Rezolvare subiect INFO 2018 – punctul b

- (b) Într-un text formatat pe linii ca la punctul (a), două spații sunt *conectate* dacă se învecinează pe verticală sau pe diagonală. Pentru textul formatat mai sus avem mai multe exemple de spații *conectate*: spațiul de pe poziția 9, linia 1, ce separă literele A și D este *conectat* cu cel de pe poziția 9, linia 2; spațiul de pe poziția 15, linia 1, este *conectat* cu cel de pe poziția 14, linia 2. Spațiul de pe poziția 3, linia 2, nu este *conectat* cu niciun alt spațiu. Să se scrie un program care citește numerele naturale L, N și apoi un text formatat pe N linii de cel mult L caractere ca la punctul (a) și afișează mesajul DA dacă în tot textul există cel puțin o pereche de spații *conectate*, altfel afișează mesajul NU.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	P	R	O	B	L	E	M	A		D	E		L	A		E	X	A	M	E	N	X
2	N	U	M	I		S	E		P	A	R	E		F	O	A	R	T	E	X	X	
3	G	R	E	U		D	E	R	E	Z	O	L	V	A	T		I	N	X	X	X	
4	T	I	M	P	U	L		A	C	O	R	D	A	T	X	X	X	X	X	X	X	

Folosește o structură de date corespunzătoare pentru stocarea liniilor: matrice cu N linii sau matrice cu 2 linii (țin minte ultimele două linii citite) sau altceva.

Marchez elementele din matrice care nu fac parte din textul meu (să nu le confund cu spațiile)

Rezolvare subiect INFO 2018 – punctul b

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	P	R	O	B	L	E	M	A		D	E		L	A		E	X	A	M	E	N	X
2	N	U		M	I	S	E		P	A	R	E		F	O	A	R	T	E	X	X	
3	G	R	E	U		D	E		R	E	Z	O	L	V	A	T		I	N	X	X	X
4	T	I	M	P	U	L		A	C	O	R	D	A	T	X	X	X	X	X	X	X	X

Foloseşte o structură de date corespunzătoare pentru stocarea liniilor: matrice cu N linii sau matrice cu 2 linii (țin minte ultimele două linii citite) sau altceva.

Marchez elementele din matrice care nu fac parte din textul meu (să nu le confund cu spaţiile)

Parcurg liniile de sus in jos (sau invers) si verific pentru fiecare spaţiu dacă are printre vecinii de pe linia următoare un spaţiu (3 posibili vecini – cazuri limită)

Rezolvare subiect INFO 2018 – punctul b

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	P	R	O	B	L	E	M	A		D	E		L	A		E	X	A	M	E	N	X
2	N	U		M	I	S	E		P	A	R	E		F	O	A	R	T	E	X	X	
3	G	R	E	U	D	E		R	E	Z	O	L	V	A	T		I	N	X	X	X	
4	T	I	M	P	U	L		A	C	O	R	D	A	T	X	X	X	X	X	X	X	

Foloseşte o structură de date corespunzătoare pentru stocarea liniilor: matrice cu N linii sau matrice cu 2 linii (țin minte ultimele două linii citite) sau altceva.

Marchez elementele din matrice care nu fac parte din textul meu (să nu le confund cu spaţiile)

Parcurg liniile de sus in jos (sau invers) si verific pentru fiecare spaţiu dacă are printre vecinii de pe linia următoare un spaţiu (**3 posibili vecini – cazuri limită**)

Rezolvare subiect INFO 2018 – punctul b

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	P	R	O	B	L	E	M	A		D	E		L	A		E	X	A	M	E	N	X
2	N	U		M	I	S	E		P	A	R	E		F	O	A	R	T	E	X	X	
3	G	R	E	U		D	E		R	E	Z	O	L	V	A	T		I	N	X	X	X
4	T	I	M	P	U	L		A	C	O	R	D	A	T	X	X	X	X	X	X	X	X

Foloseşte o structură de date corespunzătoare pentru stocarea liniilor: matrice cu N linii sau matrice cu 2 linii (țin minte ultimele două linii citite) sau altceva.

Marchez elementele din matrice care nu fac parte din textul meu (să nu le confund cu spaţiile)

Parcurg liniile de sus in jos (sau invers) si verific pentru fiecare spaţiu dacă are printre vecinii de pe linia următoare un spaţiu (**3 posibili vecini – cazuri limită**)

Mă pot opri după primul spaţiu conectat: afişez DA

Dacă am ajuns la linia N: afişez NU

Rezolvare subiect INFO 2018 – punctul b

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	P	R	O	B	L	E	M	A		D	E		L	A		E	X	A	M	E	N	X
2	N	U		M	I	S	E		P	A	R	E		F	O	A	R	T	E	X	X	
3	G	R	E	U	D	E		R	E	Z	O	L	V	A	T		I	N	X	X	X	
4	T	I	M	P	U	L		A	C	O	R	D	A	T		X	X	X	X	X	X	

- (b) Utilizarea unei structuri de date pentru menținerea informației necesare 0,5 p
Verificarea conectivității pe cazul general (cu 3 vecini) 0,5 p
Verificarea conectivității pe cazurile particulare (cu 0, 1, 2 vecini) 0,5 p
Tratarea cazului pentru DA 0,5 p
Tratarea cazului pentru NU 0,5 p
Corectitudinea limbajului 0,5 p
Explicații 0,25 p

```
9 //citire date
10 printf("L = "); scanf("%d",&L);
11 printf("N = "); scanf("%d",&N);
12
13 setbuf(stdin, NULL);
14 char matrice[N][L+1];
15 ▼ for (i = 0; i < N; i++)
16 ▼ {
17     printf("Dati linia %d formatata ",i);
18     fgets(matrice[i],L+1,stdin);
19     if(matrice[i][strlen(matrice[i]) - 1] == '\n')
20         matrice[i][strlen(matrice[i]) - 1] = '\0';
21     fflush(stdin);
22 }
23
24 //cautare spatii conectate
25 int gasit = 0;
26 for (i=0; i < N-1; i++)
27 ▼ {
28     for(j = 1; j < strlen(matrice[i])-1 && j <= strlen(matrice[i+1]); j++)
29 ▼ {
30         if (matrice[i][j] == ' ') //daca e spatiu
31         {
32             if(matrice[i+1][j-1] == ' ') //vecinul din stanga jos
33                 gasit = 1;
34
35             if (matrice[i+1][j] == ' ') //vecinul de jos
36                 gasit = 1;
37
38             if (j+1 <= strlen(matrice[i+1]) && matrice[i+1][j+1] == ' ') // vecinul din dreapta jos
39                 gasit = 1;
40
41             if (gasit)
42             {
43                 printf("DA \n");
44                 return 0;
45             }
46         }
47     }
48 }
49 printf("NU \n");
```

Rulare

```
L = 22
N = 4
Dati linia 0 formatata PROBLEMA DE LA EXAMEN
Dati linia 1 formatata NU MI SE PARE FOARTE
Dati linia 2 formatata GREU DE REZOLVAT IN
Dati linia 3 formatata TIMPUL ACORDAT
DA
```

```
L = 22
N = 2
Dati linia 0 formatata PROBLEMA DE LA EXAMEN
Dati linia 1 formatata NU PARE
NU
```

Rezolvare subiect INFO 2018 – punctul c

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	P	R	O	B	L	E	M	A		D	E		L	A		E	X	A	M	E	N	X
2	N	U		M	I	S	E		P	A	R	E		F	O	A	R	T	E	X	X	
3	G	R	E	U	D	E		R	E	Z	O	L	V	A	T		I	N	X	X	X	
4	T	I	M	P	U	L		A	C	O	R	D	A	T		X	X	X	X	X	X	

(c) În arta tipografică un *râu* este o înșiruire de spații care se întinde pe verticală, pe liniile consecutive ale unui text. Mai precis, un *râu* este definit ca o secvență de cel puțin 2 spații în care oricare 2 spații de pe linii consecutive sunt conectate. Spre exemplu, pentru textul de la punctul (a), avem un *râu* de lungime 4 format din: spațiul de pe poziția 9, linia 1; spațiul de pe poziția 9, linia 2; spațiul de pe poziția 8, linia 3 și spațiul de pe poziția 7, linia 4. De remarcat, faptul că de pe poziția 17, linia 3, nu pornește nici un *râu* întrucât linia 4 se termină pe poziția 14. Să se scrie un program, cu o complexitate de timp cât mai bună, care citește numerele naturale L, N și apoi un text formatat pe N linii de cel mult L caractere ca la punctul (a) și afișează lungimea celui mai lung *râu* posibil, dacă acesta există sau mesajul NU, dacă nu există niciun *râu*.

Rezolvare subiect INFO 2018 – punctul c

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	P	R	O	B	L	E	M	A		D	E		L	A		E	X	A	M	E	N	X
2	N	U		M	I	S	E		P	A	R	E		F	O	A	R	T	E	X	X	
3	G	R	E	U	D	E		R	E	Z	O	L	V	A	T		I	N	X	X	X	
4	T	I	M	P	U	L		A	C	O	R	D	A	T		X	X	X	X	X	X	

(c) În arta tipografică un *râu* este o înșiruire de spații care se întinde pe verticală, pe liniile consecutive ale unui text. Mai precis, un *râu* este definit ca o secvență de cel puțin 2 spații în care oricare 2 spații de pe linii consecutive sunt conectate. Spre exemplu, pentru textul de la punctul (a), avem un *râu* de lungime 4 format din: spațiul de pe poziția 9, linia 1; spațiul de pe poziția 9, linia 2; spațiul de pe poziția 8, linia 3 și spațiul de pe poziția 7, linia 4. De remarcat, faptul că de pe poziția 17, linia 3, nu pornește nici un *râu* întrucât linia 4 se termină pe poziția 14. Să se scrie un program, cu o complexitate de timp cât mai bună, care citește numerele naturale L, N și apoi un text formatat pe N linii de cel mult L caractere ca la punctul (a) și afișează lungimea celui mai lung *râu* posibil, dacă acesta există sau mesajul NU, dacă nu există niciun *râu*.

Rezolvare subiect INFO 2018 – punctul c

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	P	R	O	B	L	E	M	A		D	E		L	A		E	X	A	M	E	N	X
2	N	U		M	I	S	E		P	A	R	E		F	O	A	R	T	E	X	X	
3	G	R	E	U		D	E		R	E	Z	O	L	V	A	T		I	N	X	X	X
4	T	I	M	P	U	L		A	C	O	R	D	A	T	X	X	X	X	X	X	X	X

Calculez lungimea celei mai lungi însiruiriri de spații (care respectă regula pentru formarea unui râu) de sus în jos și de la stânga la dreapta.

Memorez pentru fiecare pozitie (i,j) această lungime – matricea rau.

Dacă la poziția (i,j) am caracterul spațiu:

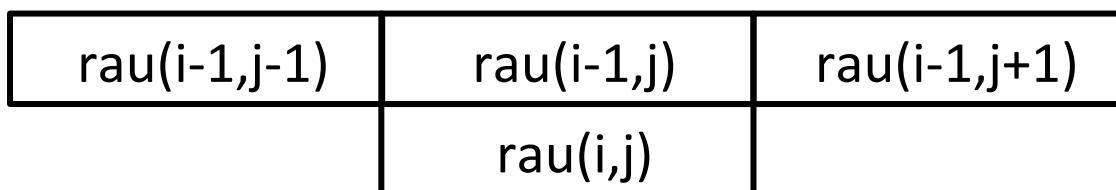
- posibil să am un râu (însiruire de spații de lungime cel puțin 2)

rau($i-1, j-1$)	rau($i-1, j$)	rau($i-1, j+1$)
rau(i, j)		

- $\text{rau}(i,j) = 1 + \max(\text{rau}(i-1, j-1), \text{rau}(i-1, j), \text{rau}(i-1, j+1))$
- trebuie să mă asigur că cei trei vecini există

Rezolvare subiect INFO 2018 – punctul c

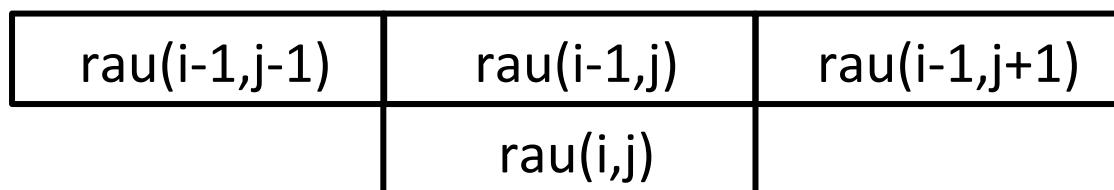
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	P	R	O	B	L	E	M	A		D	E		L	A		E	X	A	M	E	N	X
2	N	U		M	I		S	E		P	A	R	E		F	O	A	R	T	E	X	X
3	G	R	E	U		D	E		R	E	Z	O	L	V	A	T		I	N	X	X	X
4	T	I	M	P	U	L		A	C	O	R	D	A	T	X	X	X	X	X	X	X	X



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	X

Rezolvare subiect INFO 2018 – punctul c

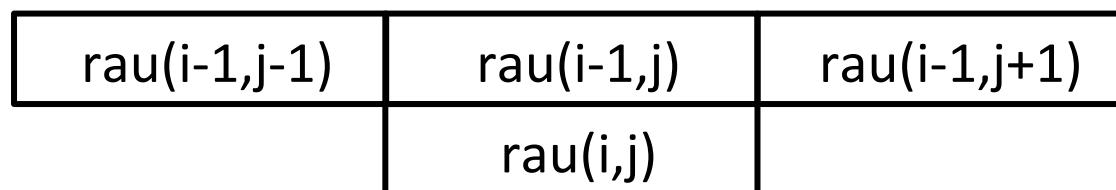
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	P	R	O	B	L	E	M	A		D	E		L	A		E	X	A	M	E	N	X
2	N	U		M	I		S	E		P	A	R	E		F	O	A	R	T	E	X	X
3	G	R	E	U		D	E		R	E	Z	O	L	V	A	T		I	N	X	X	X
4	T	I	M	P	U	L		A	C	O	R	D	A	T	X	X	X	X	X	X	X	X



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	X
2	0	0	1	0	0	1	0	0	2	0	0	0	0	0	2	0	0	0	0	0	0	X

Rezolvare subiect INFO 2018 – punctul c

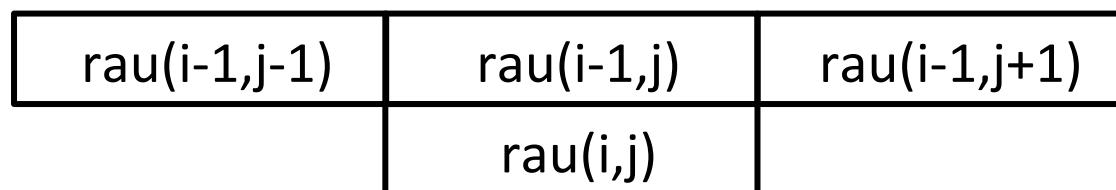
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	P	R	O	B	L	E	M	A		D	E		L	A		E	X	A	M	E	N	X
2	N	U		M	I		S	E		P	A	R	E		F	O	A	R	T	E	X	X
3	G	R	E	U		D	E		R	E	Z	O	L	V	A	T		I	N	X	X	X
4	T	I	M	P	U	L		A	C	O	R	D	A	T	X	X	X	X	X	X	X	X



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	X	
2	0	0	1	0	0	1	0	0	2	0	0	0	0	0	2	0	0	0	0	0	0	X	X
3	0	0	0	0	2	0	0	3	0	0	0	0	0	0	0	0	0	1	0	0	X	X	X

Rezolvare subiect INFO 2018 – punctul c

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	P	R	O	B	L	E	M	A		D	E		L	A		E	X	A	M	E	N	X
2	N	U		M	I		S	E		P	A	R	E		F	O	A	R	T	E	X	X
3	G	R	E	U		D	E		R	E	Z	O	L	V	A	T		I	N	X	X	X
4	T	I	M	P	U	L		A	C	O	R	D	A	T	X	X	X	X	X	X	X	X



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	X	
2	0	0	1	0	0	1	0	0	2	0	0	0	0	0	2	0	0	0	0	0	0	X	X
3	0	0	0	0	2	0	0	3	0	0	0	0	0	0	0	0	0	1	0	0	X	X	X
4	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	

◀ ▶ solutieInfo_c.c ×

```
1 #include <stdio.h>
2 #include <string.h>
3
4
5 int main(){
6
7     int L, N, i, j;
8
9     //citire date
10    printf("L = "); scanf("%d",&L);
11    printf("N = "); scanf("%d",&N);
12
13    setbuf(stdin, NULL);
14    char matrice[N][L+1], rau[N][L];
15    for (i = 0; i< N; i++)
16    {
17        printf("Dati linia %d formatata ",i);
18        fgets(matrice[i],L+1,stdin);
19        if(matrice[i][strlen(matrice[i]) - 1] == '\n')
20            matrice[i][strlen(matrice[i]) - 1] = '\0';
21        fflush(stdin);
22
23        //pune 1 unde ai spatiu
24        for(j=0;j<strlen(matrice[i]);j++)
25            rau[i][j] = matrice[i][j] == ' ';
26    }
27}
```

```
29 int lungime_maxima_rau = 0;
30
31 //cautare rau
32 for (i = 1; i < N; i++)
33     for(j = 1; j <= strlen(matrice[i])-1 && j <= strlen(matrice[i-1]) + 1; j++)
34     {
35         if (matrice[i][j] == ' ') //daca e spatiu
36         {
37             //vecinul din stanga sus
38             rau[i][j] = 1 + rau[i-1][j-1];
39
40             //vecinul de sus
41             if(j <= strlen(matrice[i-1]))
42             {
43                 if(rau[i][j] < 1 + rau[i-1][j])
44                     rau[i][j] = 1 + rau[i-1][j];
45             }
46
47             //vecinul din dreapta
48             if(j+1 <= strlen(matrice[i-1]))
49             {
50                 if(rau[i][j] < 1 + rau[i-1][j+1])
51                     rau[i][j] = 1 + rau[i-1][j+1];
52             }
53
54             if (lungime_maxima_rau < rau[i][j])
55                 lungime_maxima_rau = rau[i][j];
56         }
57     }
58
59 if(lungime_maxima_rau >=2 )
60     printf("DA, raul maxim are lungimea %d \n",lungime_maxima_rau);
61 else
62     printf("NU");
```

Rulare

```
Bogdan-Alexes-MacBook-Pro:solADMITERE2018 bogdan$ gcc solutieInfo_c.c
Bogdan-Alexes-MacBook-Pro:solADMITERE2018 bogdan$ ./a.out
L = 22
N = 4
Dati linia 0 formatata PROBLEMA DE LA EXAMEN
Dati linia 1 formatata NU MI SE PARE FOARTE
Dati linia 2 formatata GREU DE REZOLVAT IN
Dati linia 3 formatata TIMPUL ACORDAT
DA, raul maxim are lungimea 4
0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0
0 0 1 0 0 1 0 0 2 0 0 0 0 2 0 0 0 0 0 0 0 0
0 0 0 0 2 0 0 3 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

- (c) Corectitudinea soluției 1 p
Optimalitatea soluției corecte $\mathcal{O}(L \cdot N)$ 1 p
Deducerea complexității soluției prezentate 0,25 p
Explicații 0,25 p

Subject MATE 2018

Enunț subiect MATE 2018

IV. Informatică. Ionuț a fost admis la FMI și tatăl lui i-a făcut cadou o mașină la mână a doua ca să călătorească cu ea în vacanță, împreună cu prietena sa Măriuca. Din păcate, mașina este veche și la scurt timp, acul vitezometrului se rupe. Descurcăreț de mic, Ionuț îl lipește cu adeziv, dar își dă seama că în urma reparației, acul nu mai indică viteza corectă. Făcând mai multe experimente, Ionuț realizează că viteza pe care o indică acul vitezometrului diferă de viteza reală a mașinii cu un număr real constant c , a cărui valoare absolută e mai mică decât 100 km/oră.

Pentru a calcula constanta c , Ionuț merge pe 3 segmente de drum succesive, pe fiecare dintre ele cu viteză constantă și își notează viteza indicată de acul defect al vitezometrului pe fiecare segment de drum:

- pe primul segment de drum, de 40 km, acul îi indică o viteză de 50 km/oră;
- pe al doilea segment de drum, tot de 40 km, parcurs la o altă viteză, acul îi indică 60 km/oră;
- pe al treilea segment de drum, de 100 km, acul îi indică 90 km/oră.

În acest timp, Măriuca înregistrează timpii în care sunt parcurse cele 3 segmente de drum, dar pentru că vrea să îl pună la încercare pe Ionuț, la final nu îi spune decât timpul total t de 5 ore în care a parcurs toate cele 3 segmente de drum cumulate. După ce își aduce aminte din liceu că viteza se calculează ca raportul dintre distanță și timp, Ionuț ajunge la concluzia că acul vitezometrului indică întotdeauna cu 30 km/oră mai mult decât viteza reală a mașinii.

Scrieți un program care citește numărul natural n , reprezentând numărul segmentelor succesive de drum pe care Ionuț le parurge, numărul real t , reprezentând timpul total (în ore) în care acestea sunt parcurse, precum și n perechi de numere reale, fiecare reprezentând lungimea unui segment de drum (în km) și valoarea indicată de acul vitezometrului pe segmentul de drum respectiv (în km/oră), iar apoi afișează valoarea numerică a constantei reale c (în km/oră), calculată cu o precizie de 2 zecimale, cu care viteza indicată de acul defect diferă de viteza reală a mașinii.

Barem subiect MATE 2018

IV. Informatică. Oficiu	1 p
Găsirea relației care permite aflarea constantei c	2 p
Considerarea unui interval de căutare pentru constanta c care conține și valori negative	1 p
Găsirea constantei c prin căutare	3 p
Respectarea aproximării de 2 zecimale	1 p
Corectitudinea limbajului	1 p
Explicații	1 p

Rezolvare subiect MATE 2018

Găsirea relației care permite aflarea constantei c

- viteza afisata de acul vitezometrului = viteza reală + c
- pentru fiecare segment de drum i avem: $v_i + c = \frac{d_i}{t_i}$ v_i – viteza reală, $v_i + c$ – viteza afişată
 d_i – distanța parcursă
 t_i – timpul pe segmentul de drum i
- știu timpul total $t = t_1 + t_2 + \dots + t_n$
- dar fiecare t_i verifică relația: $t_i = \frac{d_i}{v_i + c}$
- în final , avem că: $\sum_{i=1}^n \frac{d_i}{v_i + c} = t$
- verificare cu datele problemei: $\frac{40}{50+c} + \frac{40}{60+c} + \frac{100}{90+c} = 5$
- soluția $c = -30$: $\frac{40}{20} + \frac{40}{30} + \frac{100}{60} = \frac{120 + 80 + 100}{60} = \frac{300}{60} = 5$

Rezolvare subiect MATE 2018

Cum găsesc constanta c?

- consider funcția: $f(c) = \sum_{i=1}^n \frac{d_i}{v_i + c}$
- observație: dacă $c_1 < c_2$ atunci avem că $\frac{d_i}{v_i + c_1} > \frac{d_i}{v_i + c_2}$
- de aici rezultă că funcția f este monoton descrescătoare
- caut constanta c în intervalul (-100, +100) cu pasul 0.01 astfel încât f(c) este cea mai apropiată valoare de t
- pot face căutare secvențială (complexitate liniară) sau căutare binară
- funcția f(c) se va apropiă de t apoi se va îndepărta

Cod soluție subiect MATE 2018

```
solutieMate.c      x
1 #include <stdio.h>
2 #include <math.h>
3
4 float calculeazaValoareFunctieC(float c, int nr_segmente, float v[], float d[])
5 {
6     int i;
7     float valFunctie = 0;
8     for(i = 0; i < nr_segmente; i++)
9     {
10         if (v[i] + c != 0)
11             valFunctie += d[i]/(v[i]+c);
12         else
13             return -1;
14     }
15     return valFunctie;
16 }
```

```
17
18 int main(){
19     int i, n;
20     float c, t, viteza_ac[100], d[100];
21
22     //citesc datele
23     printf("n = "); scanf("%d",&n);
24     for (i = 0; i < n; i++)
25     {
26         printf("Citesc datele pentru segmentul de drum %d \n", i+1);
27
28         printf("viteza afisata de ac v[%d] = ",i);
29         scanf("%f",&viteza_ac[i]);
30
31         printf("distanta segmentului de drum %d d[%d] = ", i+1 ,i);
32         scanf("%f",&d[i]);
33     }
34     printf("Timpul total t = ");
35     scanf("%f",&t);
36
37     //cautare secentiala
38     float minim = fabs(calculeazaValoareFunctieC(-100,n,viteza_ac,d) - t);
39     float c_optim = -100;
40
41     for (c = -100; c < 100; c = c + 0.01)
42     {
43         float valFunctie = calculeazaValoareFunctieC(c,n,viteza_ac,d);
44         if (fabs(valFunctie - t) < minim)
45         {
46             minim = fabs(valFunctie - t);
47             c_optim = c;
48         }
49     }
50
51     printf("c = %f \n", c_optim);
52
53     return 0;
54 }
```

Rulare

```
Bogdan-Alexes-MacBook-Pro:solADMITERE2018 bogdan$ gcc -o solutieMate solutieMate.c
Bogdan-Alexes-MacBook-Pro:solADMITERE2018 bogdan$ ./solutieMate
n = 3
Citesc datele pentru segmentul de drum 1
viteza afisata de ac v[0] = 50
distanta segmentului de drum 1 d[0] = 40
Citesc datele pentru segmentul de drum 2
viteza afisata de ac v[1] = 60
distanta segmentului de drum 2 d[1] = 40
Citesc datele pentru segmentul de drum 3
viteza afisata de ac v[2] = 90
distanta segmentului de drum 3 d[2] = 100
Timpul total t = 5
c = -29.997637
```

Subject INFO 2017

Enunț subiect INFO 2017

IV. Informatică.

Fie n un număr natural nenul. Fie v un vector cu n poziții numerotate de la 1 la n și elemente numere naturale diferite, de la 1 la n , într-o ordine oarecare. Pentru i și j numere naturale între 1 și n , numim $\text{FLIP}(n, v, i, j)$ operația care inversează ordinea elementelor din v situate pe pozițiile de la i la j .

- Să se scrie în limbaj de programare o procedură (sau funcție) care implementează operația $\text{FLIP}(n, v, i, j)$.
- Să se scrie un program care sortează crescător vectorul v , folosind pentru schimbarea ordinii elementelor în v doar operația $\text{FLIP}(n, v, 1, k)$, cu k de la 2 la n .
- Considerăm că n este o putere a lui 2 ($n = 2^m$, cu m număr natural nenul) și vectorul v are proprietatea că pentru orice i de la 1 la m și orice j de la 1 la 2^{m-i} , există k de la 1 la 2^{m-i} , astfel încât pe pozițiile din v de la $2^i(j-1)+1$ la $2^i j$ se află numerele naturale de la $2^i(k-1)+1$ la $2^i k$, într-o ordine oarecare. Să se scrie un program care sortează crescător vectorul v , folosind pentru schimbarea ordinii elementelor în v doar operația $\text{FLIP}(n, v, 2^i(j-1)+1, 2^i j)$, cu i de la 1 la m și j de la 1 la 2^{m-i} , printr-un algoritm mai eficient decât cel implementat la punctul b), care se bazează pe proprietatea vectorului v .

Enunț subiect INFO 2017

Exemple:

	Date de intrare	Date de ieșire
a)	FLIP(9, [3 2 6 8 5 9 1 7 4], 1, 6)	$v = [9 5 8 6 2 3 1 7 4]$
	FLIP(4, [2 1 4 3], 1, 4)	$v = [3 4 1 2]$
	FLIP(16, [14 13 15 16 11 12 9 10 2 1 4 3 8 7 6 5], 5, 8)	$v = [14 13 15 16 10 9 12 11 2 1 4 3 8 7 6 5]$
b)	$n = 9$ $v = [3 2 6 8 5 9 1 7 4]$	$v = [1 2 3 4 5 6 7 8 9]$
c)	$n = 4$ $v = [2 1 4 3]$	$v = [1 2 3 4]$
	$n = 16$ $v = [14 13 15 16 11 12 9 10 2 1 4 3 8 7 6 5]$	$v = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]$

Note:

1. Programele vor fi scrise într-unul dintre limbajele de programare studiate în liceu (Pascal,C,C++). La fiecare subpunct a), b), c), se va preciza complexitatea timp, în funcție de n , a soluției implementate și se vor descrie informal detaliile algoritmului folosit și ale implementării sub formă de program: semnificația variabilelor, a structurilor de date, a structurilor repetitive, a instrucțiunilor condiționale.
2. Toate operațiile de tip FLIP se vor face în vectorul v , fără a se folosi alți vectori auxiliari.
3. La subpunctul a), datele se transmit ca parametri ai procedurii/funcției $FLIP(n, v, i, j)$. La subpunctele b) și c), se citesc de la tastatură n și v , fiecare pe un rând separat și se afișează vectorul v sortat crescător, pe un singur rând. Se va considera că datele de intrare ale programelor sunt oricât de mari, dar fără a pune probleme de reprezentare în memorie cu ajutorul tipurilor de date standard.
4. Programele vor folosi doar instrucțiunile de bază ale limbajului de programare ales, inclusiv cele de intrare/ieșire, dar nu și alte funcții din biblioteci specialize.

Barem subiect INFO 2017

IV. Informatică.	Oficiu	1 p
(a)	Folosirea corectă a noțiunii de procedură / funcție	1 p
	Implementarea fără vector auxiliar	1 p
	Corectitudinea soluției	1 p
	Corectitudinea limbajului	0,5 p
	Explicații	0,25 p
	Complexitate	0,25 p
(b)	Determinarea maximului dintr-un vector	0,5 p
	Utilizarea FLIP conform cerinței	1 p
	Corectitudinea soluției	1 p
	Corectitudinea limbajului	0,5 p
	Explicații	0,25 p
	Complexitate	0,25 p
(c)	Corectitudinea soluției	1 p
	Explicații	0,25 p
	Complexitate	0,25 p

Rezolvare subiect INFO 2017 – punctul a

- a) Să se scrie în limbaj de programare o procedură (sau funcție) care implementează operația $\text{FLIP}(n, v, i, j)$.

	Date de intrare	Date de ieșire
a)	$\text{FLIP}(9, [3\ 2\ 6\ 8\ 5\ 9\ 1\ 7\ 4], 1, 6)$	$v = [9\ 5\ 8\ 6\ 2\ 3\ 1\ 7\ 4]$
	$\text{FLIP}(4, [2\ 1\ 4\ 3], 1, 4)$	$v = [3\ 4\ 1\ 2]$
	$\text{FLIP}(16, [14\ 13\ 15\ 16\ 11\ 12\ 9\ 10\ 2\ 1\ 4\ 3\ 8\ 7\ 6\ 5], 5, 8)$	$v = [14\ 13\ 15\ 16\ 10\ 9\ 12\ 11\ 2\ 1\ 4\ 3\ 8\ 7\ 6\ 5]$

```
void flip(int n, int v[], int i, int j)
{
    i=i-1;//pornim de la 0 in C
    j=j-1;//pornim de la 0 in C
    int kk,aux;
    //nu mergem cu contorul pana la capat pentru ca altfel nu facem nimic
    for(kk=i;kk<=(i+j)/2;kk++)
    {
        aux = v[kk];
        v[kk] = v[j-kk+i];
        v[j-kk+i] = aux;
    }
}
```

Rezolvare subiect INFO 2017 – punctul a

- a) Să se scrie în limbaj de programare o procedură (sau funcție) care implementează operația $\text{FLIP}(n, v, i, j)$.

	Date de intrare	Date de ieșire
a)	$\text{FLIP}(9, [3\ 2\ 6\ 8\ 5\ 9\ 1\ 7\ 4], 1, 6)$	$v = [9\ 5\ 8\ 6\ 2\ 3\ 1\ 7\ 4]$
	$\text{FLIP}(4, [2\ 1\ 4\ 3], 1, 4)$	$v = [3\ 4\ 1\ 2]$
	$\text{FLIP}(16, [14\ 13\ 15\ 16\ 11\ 12\ 9\ 10\ 2\ 1\ 4\ 3\ 8\ 7\ 6\ 5], 5, 8)$	$v = [14\ 13\ 15\ 16\ 10\ 9\ 12\ 11\ 2\ 1\ 4\ 3\ 8\ 7\ 6\ 5]$

Altă soluție:

```
void flip2(int n, int v[], int i, int j)
{
    i = i-1;
    j = j-1;
    int aux;
    while(i < j)
    {
        aux = v[j];
        v[j] = v[i];
        v[i] = aux;
        i++;
        j--;
    }
}
```

Complexitate liniară $O(n)$

Intuiție – punctul b

- b) Să se scrie un program care sortează crescător vectorul v , folosind pentru schimbarea ordinii elementelor în v doar operația $\text{FLIP}(n, v, 1, k)$, cu k de la 2 la n .

$n = 9$

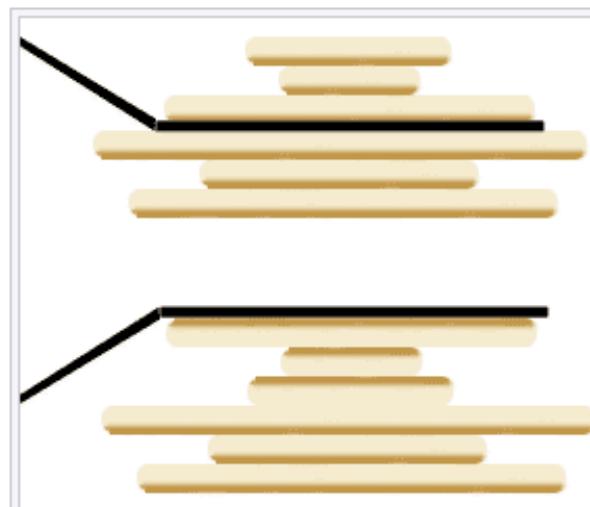
$v = [3 \ 2 \ 6 \ 8 \ 5 \ 9 \ 1 \ 7 \ 4]$

$v = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9]$

Pancake sorting

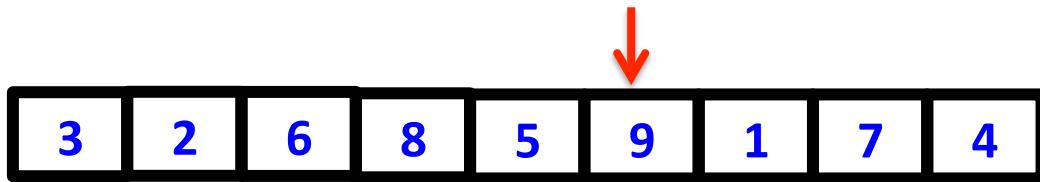
From Wikipedia, the free encyclopedia

Pancake sorting is the colloquial term for the mathematical problem of sorting a disordered stack of pancakes in order of size when a [spatula](#) can be inserted at any point in the stack and used to flip all pancakes above it. A *pancake number* is the minimum number of flips required for a given number of pancakes. In this form, the problem was first discussed by [American geometer Jacob E. Goodman](#).^[1] It is a variation of the [sorting](#) problem in which the only allowed operation is to reverse the elements of some [prefix](#) of the sequence. Unlike a traditional

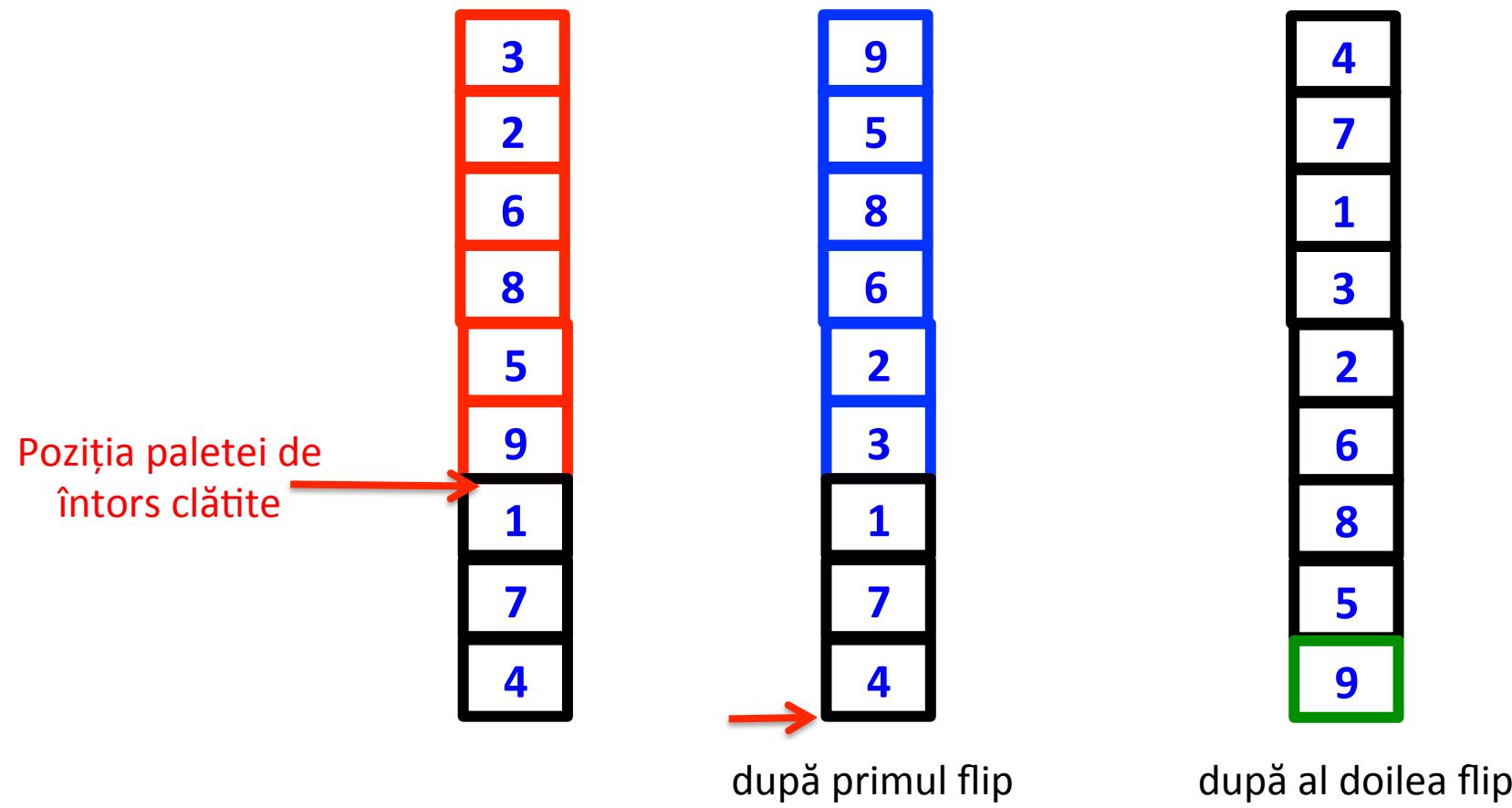


Demonstration of the primary operation. The spatula is flipping over the top three pancakes, with the result seen below. In the burnt pancake problem, their top sides would now be burnt instead of their bottom sides.

Intuiție – punctul b



Vector cu diametrele clătitelor



Rezolvare subiect INFO 2017 – punctul b

- b) Să se scrie un program care sortează crescător vectorul v , folosind pentru schimbarea ordinii elementelor în v doar operația $\text{FLIP}(n, v, 1, k)$, cu k de la 2 la n .

$n = 9$
 $v = [3 \ 2 \ 6 \ 8 \ 5 \ 9 \ 1 \ 7 \ 4]$

$v = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9]$

Idee:

Pasul 1: găsesc poziția ***pozMaxim*** pe care se află maximul ($= n$) din vectorul $v[1\dots n]$ și duc maximul pe ultima poziție (n) cu două flip-uri:

pozMaxim
↓



Rezolvare subiect INFO 2017 – punctul b

- b) Să se scrie un program care sortează crescător vectorul v , folosind pentru schimbarea ordinii elementelor în v doar operația $\text{FLIP}(n, v, 1, k)$, cu k de la 2 la n .

$n = 9$
 $v = [3 \ 2 \ 6 \ 8 \ 5 \ 9 \ 1 \ 7 \ 4]$

$v = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9]$

Idee:

Pasul 1: găsesc poziția ***pozMaxim*** pe care se află maximul ($= n$) din vectorul $v[1\dots n]$ și duc maximul pe ultima poziție (n) cu două flip-uri:

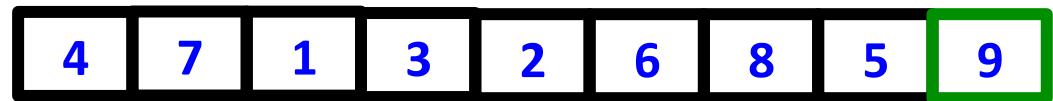
pozMaxim



$\text{flip}(n, v, 1, \text{pozMaxim})$



$\text{flip}(n, v, 1, n)$



Rezolvare subiect INFO 2017 – punctul b

- b) Să se scrie un program care sortează crescător vectorul v , folosind pentru schimbarea ordinii elementelor în v doar operația $\text{FLIP}(n, v, 1, k)$, cu k de la 2 la n .

$n = 9$

$v = [3 \ 2 \ 6 \ 8 \ 5 \ 9 \ 1 \ 7 \ 4]$

$v = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9]$

Idee:

Pasul 2: găsesc poziția ***pozMaxim*** pe care se află maximul ($= n-1$) din vectorul $v[1\dots n-1]$ și duc maximul pe poziția $n-1$ cu două flip-uri:

pozMaxim



Rezolvare subiect INFO 2017 – punctul b

- b) Să se scrie un program care sortează crescător vectorul v , folosind pentru schimbarea ordinii elementelor în v doar operația $\text{FLIP}(n, v, 1, k)$, cu k de la 2 la n .

$n = 9$
 $v = [3 \ 2 \ 6 \ 8 \ 5 \ 9 \ 1 \ 7 \ 4]$

$v = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9]$

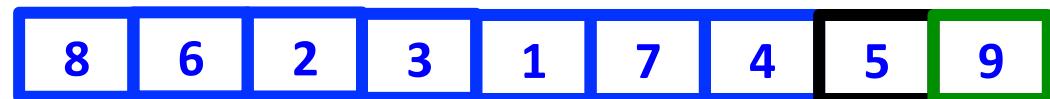
Idee:

Pasul 2: găsesc poziția ***pozMaxim*** pe care se află maximul ($= n-1$) din vectorul $v[1\dots n-1]$ și duc maximul pe poziția $n-1$ cu două flip-uri:

pozMaxim



$\text{flip}(n, v, 1, \text{pozMaxim})$



$\text{flip}(n, v, 1, n-1)$



Rezolvare subiect INFO 2017 – punctul b

- b) Să se scrie un program care sortează crescător vectorul v , folosind pentru schimbarea ordinii elementelor în v doar operația $\text{FLIP}(n, v, 1, k)$, cu k de la 2 la n .

$n = 9$
 $v = [3 \ 2 \ 6 \ 8 \ 5 \ 9 \ 1 \ 7 \ 4]$

$v = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9]$

Idee:

Pasul $i+1$: găsesc poziția ***pozMaxim*** pe care se află maximul ($= n-i$) din vectorul $v[1\dots n-i]$ și duc maximul pe poziția $n-i$ cu două flip-uri:

pozMaxim



$\text{flip}(n, v, 1, \text{pozMaxim})$



$\text{flip}(n, v, 1, n-i)$



Complexitate pătratică $O(n^2)$

Rezolvare subiect INFO 2017 – punctul b

- b) Să se scrie un program care sortează crescător vectorul v , folosind pentru schimbarea ordinii elementelor în v doar operația $\text{FLIP}(n, v, 1, k)$, cu k de la 2 la n .

```
void sorteazaFlip(int n,int v[])
{
    //alternativa: pot sa pornesc si de la coada
    //->cand gasesc maximul verific daca e ultimul element si il las neschimbat
    //solutia propusa: pornesc de la inceput si caut maximul, apoi fac 2 flip-uri
    int i,j;
    for(i=0;i<n;i++)
    {
        //gaseste pozitia pe care se afla n-i = maximul din vectorul ramas
        int maxim = v[0];
        int pozMaxim = 0;
        for(j=1;j<n-i;j++)
            if(maxim<v[j])
            {
                maxim = v[j];
                pozMaxim = j;
            }
        //stiu acum ca maximul se afla pe pozitia pozMaxim
        //PRIMUL FLIP
        flip(n,v,1,pozMaxim+1); //apelez flip cu +1 pt ca numar de la 0
        //AL DOILEA FLIP
        flip(n,v,1,n-i);
    }
}
```

Intuiție – punctul c

- c) Considerăm că n este o putere a lui 2 ($n = 2^m$, cu m număr natural nenul) și vectorul v are proprietatea că pentru orice i de la 1 la m și orice j de la 1 la 2^{m-i} , există k de la 1 la 2^{m-i} , astfel încât pe pozițiile din v de la $2^i(j-1)+1$ la $2^i j$ se află numerele naturale de la $2^i(k-1)+1$ la $2^i k$, într-o ordine oarecare. Să se scrie un program care sortează crescător vectorul v , folosind pentru schimbarea ordinii elementelor în v doar operația $\text{FLIP}(n, v, 2^i(j-1)+1, 2^i j)$, cu i de la 1 la m și j de la 1 la 2^{m-i} , printr-un algoritm mai eficient decât cel implementat la punctul b), care se bazează pe proprietatea vectorului v .

c)	$n = 4$ $v = [2 \ 1 \ 4 \ 3]$	$v = [1 \ 2 \ 3 \ 4]$
	$n = 16$ $v = [14 \ 13 \ 15 \ 16 \ 11 \ 12 \ 9 \ 10 \ 2 \ 1 \ 4 \ 3 \ 8 \ 7 \ 6 \ 5]$	$v = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16]$

Cum se traduce proprietatea vectorului v ?

Dacă lungimea vectorului e putere a lui 2, $n = 2^m$, ATUNCI ORICUM am împărți vectorul v în intervale de lungime 2^i (există 2^{m-i} astfel de intervale) în al j -lea interval de lungime 2^i din v se regăsesc numere naturale consecutive dintr-un interval de forma $[2^i(k-1)+1 \ 2^i k]$ (începe cu număr impar).

Intuiție – punctul c

Cum se traduce proprietatea vectorului v?

Dacă lungimea vectorului e putere a lui 2: $n = 2^m$ ATUNCI ORICUM am împărți vectorul v în intervale de lungime 2^i (există 2^{m-i} astfel de intervale) în al j-lea interval de lungime 2^i din v se regăsesc numere naturale consecutive dintr-un interval de forma $[2^i(k-1)+1, 2^i k]$ (începe cu număr impar).

Să generăm aleator un asemenea vector de lungime $n = 2^3$, $m=3$



Pentru a respecta proprietatea trebuie să pun 6



Pentru a respecta proprietatea trebuie să am pe următoarele 2 poziții fie 7, 8 fie 8, 7

Intuiție – punctul c

Cum se traduce proprietatea vectorului v?

Dacă lungimea vectorului e putere a lui 2: $n = 2^m$ ATUNCI ORICUM am împărți vectorul v în intervale de lungime 2^i (există 2^{m-i} astfel de intervale) în al j-lea interval de lungime 2^i din v se regăsesc numere naturale consecutive dintr-un interval de forma $[2^i(k-1)+1, 2^ik]$ (începe cu număr impar).

Să generăm aleator un asemenea vector de lungime $n = 2^3$, $m=3$

5	6	8	7	2	?		
---	---	---	---	---	---	--	--

5	6	8	7	2	1		
---	---	---	---	---	---	--	--

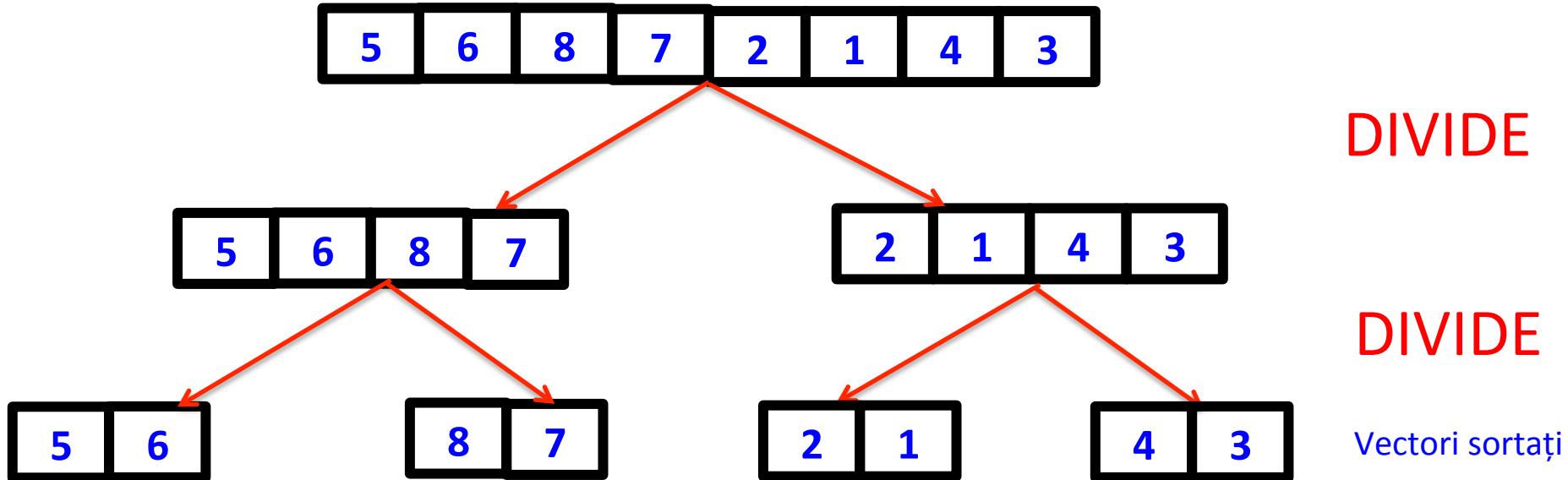
Pentru a respecta proprietatea trebuie să pun 1

5	6	8	7	2	1	4	3
---	---	---	---	---	---	---	---

Pentru a respecta proprietatea trebuie să am pe următoarele 2 poziții fie 3, 4 fie 4, 3

Soluție – punctul c

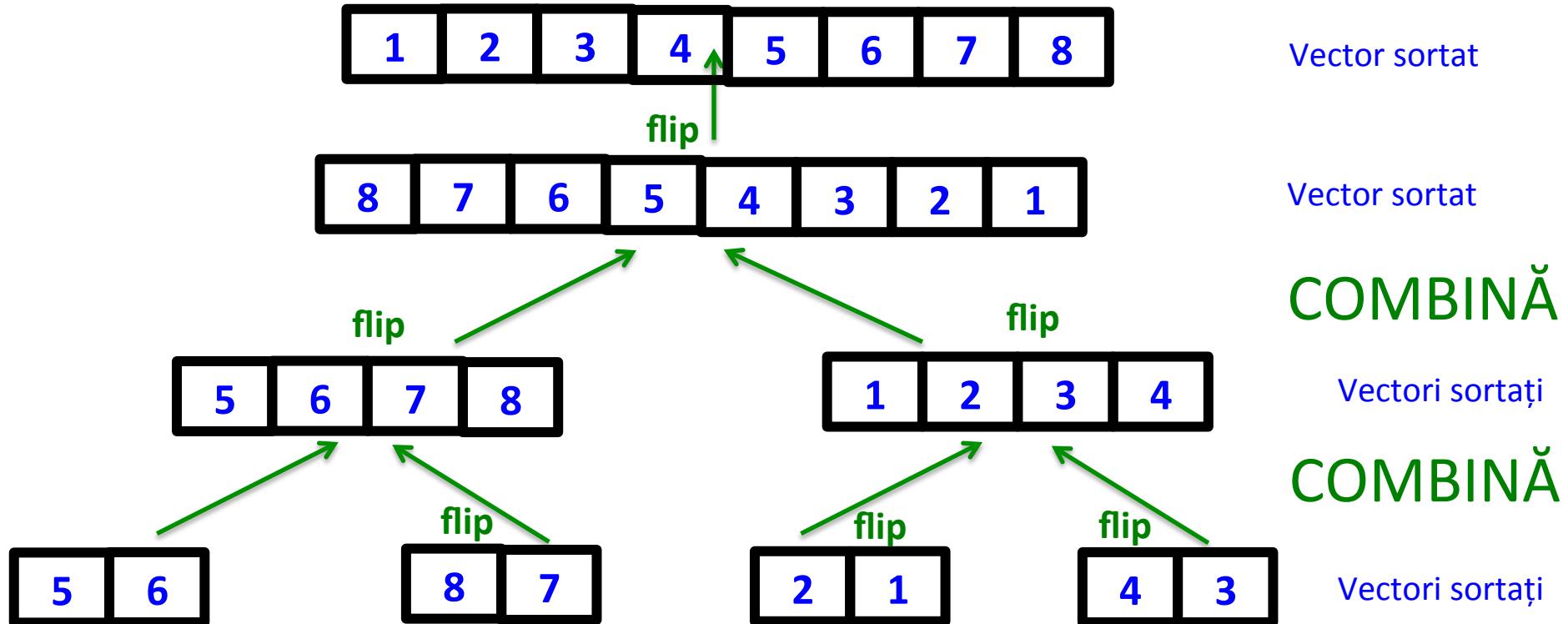
Divide et impera pe intervale de forma 2^i folosind flip-uri pe subintervale



COMBINĂ SOLUȚIA (prin flip-uri):

- dacă elementele din subvectorul din stânga sunt mai mici decât cele din dreapta trebuie să sortezi subvectorul din stânga crescător (cu flip-uri) și cel din dreapta crescător (cu flip-uri)
- dacă elementele din subvectorul din stânga sunt mai mari decât cele din dreapta trebuie să sortezi subvectorul din stânga descrescător (cu flip-uri) și cel din dreapta descrescător (cu flip-uri)

Soluție – punctul c



COMBINĂ SOLUȚIA (prin flip-uri):

- dacă elementele din subvectorul din stânga sunt mai mici decât cele din dreapta trebuie să sortezi subvectorul din stânga crescător (cu flip-uri) și cel din dreapta crescător (cu flip-uri)
- dacă elementele din subvectorul din stânga sunt mai mari decât cele din dreapta trebuie să sortezi subvectorul din stânga descrescător (cu flip-uri) și cel din dreapta descrescător (cu flip-uri)

Complexitate $O(n \log(n))$

Soluție – punctul c

```
void sorteazaFlipOptimDivideEtImpera(int n, int v[],int i,int j)
{
    //conditia de oprire
    if (j-i==1)
    {
        // nu faci nimic daca ai doua elemente, pt ca nu stii la acest moment cum sa le sortezi
        return;
    }
    //divide problema in 2 subprobleme
    sorteazaFlipOptimDivideEtImpera(n,v,i,(i+j)/2);
    sorteazaFlipOptimDivideEtImpera(n,v,(i+j)/2+1,j);
    //combina solutia
    // daca elementele din subvectorul din stanga sunt mai mici decat cele din dreapta
    // trebuie sa sortezi subvectorul din stanga crescator si cel din dreapta crescator
    if(v[i-1]<v[j-1])
    {
        if (v[i-1] > v[(i+j)/2-1])
            flip(n,v,i,(i+j)/2);
        if(v[(i+j)/2] > v[j-1])
            flip(n,v,(i+j)/2+1,j);
    }
    else
    {
        // elementele din subvectorul din stanga sunt mai mari decat cele din dreapta
        // trebuie sa sortezi subvectorul din stanga descrescator si cel din dreapta descrescator
        if (v[i-1] < v[(i+j)/2-1])
            flip(n,v,i,(i+j)/2);
        if(v[(i+j)/2] < v[j-1])
            flip(n,v,(i+j)/2+1,j);
    }
}
```

Soluție – punctul c

În main:

- citește datele din vector
- sortează vectorul cu funcția anterioară
- dacă e sortat descrescător aplică flip

```
int v6[] = {14, 13, 15, 16, 11, 12, 9, 10, 2, 1, 4, 3, 8, 7, 6, 5};  
n=16;  
sorteazaFlipOptimDivideEtImpera(n,v6,1,n);  
//verifica ca nu e sortat descrescator  
if(v6[0] > v6[n-1])  
    flip(n,v6,1,n);  
afiseaza(n,v6);
```

Subject MATE 2017

Enunț subiect MATE 2017

IV. Informatică.

Considerăm triunghiul infinit de mai jos, format din numere naturale:

			1			
		2	3			
	4	5	6			
	7	8	9	10		
11	12	13	14	15		
16	17	18	19	20	21	
.....						

Spunem că perechea de numere (x, y) este adiacență dacă x și y sunt vecini pe aceeași linie sau pe diagonală, pe linii consecutive. Spre exemplu, $(8,9)$, $(12,8)$ și $(8,13)$ sunt adiacențe, dar $(8,14)$ sau $(18,8)$ nu sunt adiacențe.

Numim drum de la x la y de lungime $p - 1$, cu $p \geq 1$, o secvență de numere $x_1 x_2 x_3 \dots x_p$, cu $x = x_1$ și $y = x_p$ și cu proprietatea că toate perechile (x_i, x_{i+1}) , cu i de la 1 la $p - 1$, sunt adiacențe.

Scrieți un program, într-unul dintre limbajele de programare studiate în liceu (Pascal,C,C++), care primește ca date de intrare 2 numere naturale nenule x și y și afișează un drum de la x la y de lungime minimă. Spre exemplu, 1 2 5 8 13 este un drum de lungime minimă de la 1 la 13.

Notă: Se vor descrie informal detaliile algoritmului folosit și ale implementării sub formă de program: semnificația variabilelor, a structurilor de date, a structurilor repetitive, a instrucțiunilor condiționale.

Barem subiect MATE 2017

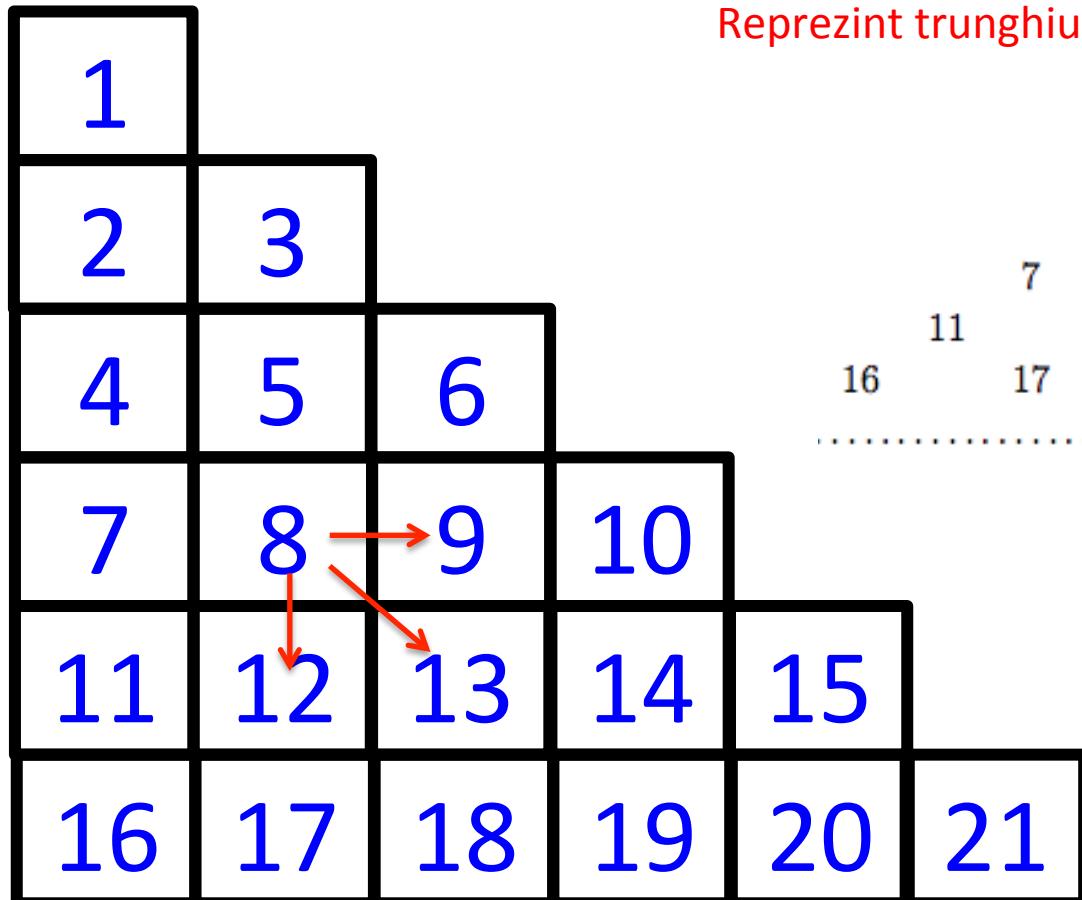
IV. Informatică. Oficiu	1 p
Tratarea celor trei cazuri $x < y, x = y, x > y$	1 p
Determinarea nivelului pe care se află un număr	1 p
Determinarea corectă a tuturor vecinilor (adiacențelor) unui număr	1 p
Afișarea unui drum corect pentru orice x și y	1 p
Afișarea unui drum corect minim pentru orice x și y	3 p
Corectitudinea limbajului	1 p
Explicații	1 p

Reprezentare triunghiuri ca matrice

Reprezentă triunghiul infinit ca o matrice triunghiulară

Reprezent trunghiul					
1					
2	3				
4	5	6			
7	8	9	10		
11	12	13	14	15	
16	17	18	19	20	21

Adiacențe în matrice



Reprezint trunchiul infinit ca o matrice triunghiulară

1						
2	3					
4	5	6				
7	8	9	10			
11	12	13	14	15		
16	17	18	19	20	21	
.....						

Adiacențe în matrice:

- aceeași linie
- aceeași coloană
- diagonală dreapta jos

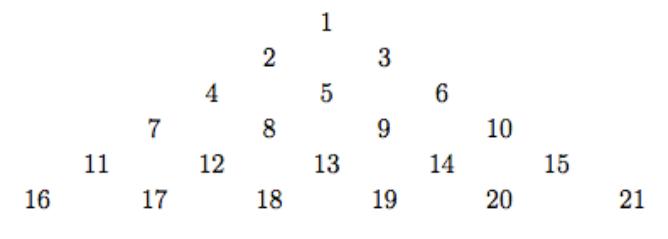
Puncte pe aceeași coloană

1					
2	3				
4	5	6			
7	8	9	10		
11	12	13	14	15	
16	17	18	19	20	21

linia 2, coloana 2

linia 6, coloana 2

Drumul de lungime minimă:
-coboară pe coloană: 3, 5, 8, 12, 17



Puncte pe coloane diferite coloanaX > coloanaY

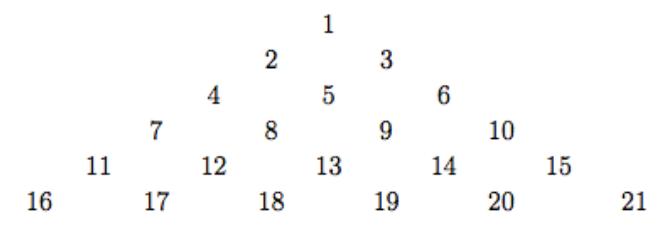
1					
2	3				
4	5	6			
7	8	9	10		
11	12	13	14	15	
16	17	18	19	20	21

linia 3, coloana 3

linia 6, coloana 1

Drumul de lungime minimă:

- mergi în stânga până când coloanaX = coloanaY, apoi cobori drept: 6, 5, 4, 7, 11, 16 (nu pot să cobor pe diagonală în stânga)



Puncte pe coloane diferite coloanaX < coloanaY

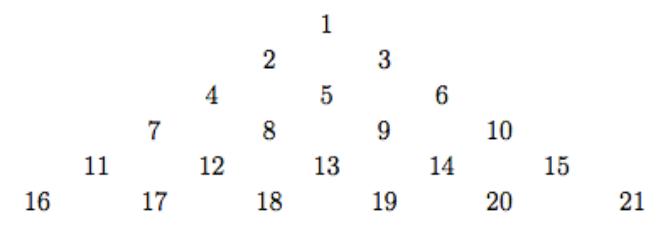
1					
2	3				
4	5	6			
7	8	9	10		
11	12	13	14	15	
16	17	18	19	20	21

linia 3, coloana 1

linia 6, coloana 3

Drumul de lungime minimă:

- mergi în diagonală până când coloanaX = coloanaY sau linieX = linieY apoi fie cobori pe coloană fie o iezi la dreapta: 4, 8, 13, 18



Puncte pe coloane diferite
coloanaX < coloanaY

1					
2	3				
4	5	6			
7	8	9	10		
11	12	13	14	15	
16	17	18	19	20	21

linia 3, coloana 1

linia 6, coloana 4

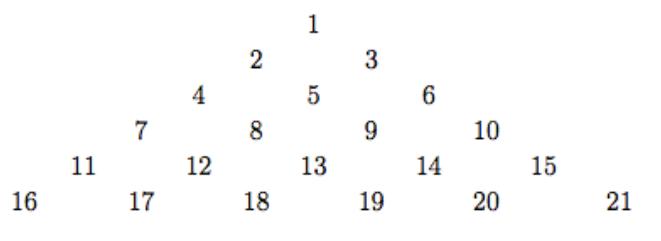
Drumul de lumeni
- mergi în diagonala
coloana Y sau X
coloană fie o încercare

linia 3, coloana 1

linia 6, coloana 6

Drumul de lungime minimă:

- mergi în diagonală până când coloanaX = coloanaY sau linieX = linieY apoi fie cobori pe coloană fie o iezi la dreapta: 4, 8, 13, 19, 20, 21



Soluție

```
int main()
{
    int x, y;

    printf("x = ");scanf("%d",&x);
    printf("y = ");scanf("%d",&y);
    if(x==y)
    {
        printf("Drumul de lungime minima este: %d \n",x);
        return 0 ;
    }

    if (x>y)
        afiseazaDrumLungimeMinima(y,x);
    else
        afiseazaDrumLungimeMinima(x,y);

    return 0;
}
```

Soluție

```
void afiseazaDrumLungimeMinima(int x, int y)
{
    int linieX,coloanaX,linieY,coloanaY,i;
    linieX = determinaLinie(x);
    coloanaX = determinaColoana(x);
    linieY = determinaLinie(y);
    coloanaY = determinaColoana(y);

    int a[100][100];
    determinaMatrice(a,y);

    //determina drumul de lungime minima astfel:
    //daca coloanaX = coloanaY cobori direct pe coloana
    if(coloanaX == coloanaY)
    {
        printf("Drumul de lungime minima este: ");
        for (i=0;i<=linieY-linieX;i++)
            printf("%d ", a[linieX+i][coloanaX]);
    }
    //daca coloanaX > coloana Y mergi in stanga pana cand ajungi pe aceeasi coloana cu y
    // apoi cobori direct
    if(coloanaX > coloanaY)
    {
        printf("Drumul de lungime minima este: ");
        for (i=0;i<=coloanaX-coloanaY;i++)
            printf("%d ", a[linieX][coloanaX-i]);
        for (i=1;i<=linieY-linieX;i++)
            printf("%d ", a[linieX+i][coloanaY]);
    }
    //daca coloanaX < coloana Y cobori pe diagonala pana cand ajungi pe aceeasi coloana sau linie cu y
    if(coloanaX < coloanaY)
    {
```

Soluție

```
//daca coloanaX < coloana Y cobori pe diagonala pana cand ajungi pe aceeasi coloana sau linie cu y
if(coloanaX < coloanaY)
{
    printf("Drumul de lungime minima este: ");
    for (i=0;(i<=linieY-linieX) && (coloanaX+i <= coloanaY);i++)
        printf("%d ", a[linieX+i][coloanaX+i]);
    i--;
    int j;
    //daca sunt pe aceeasi linie cu Y o iau la dreapta
    if(linieX + i ==linieY)
        for (j=1;coloanaX+i+j <= coloanaY;j++)
            printf("%d ", a[linieX+i][coloanaX+i+j]);

    //daca sunt pe aceeasi coloana cu Y o iau in jos
    if(coloanaX + i ==coloanaY)
        for (j=1;linieX+i+j <= linieY;j++)
            printf("%d ", a[linieX+i+j][coloanaX+i]);
}
}
```

Soluție

```
int determinaLinie(int a)
{
    int linie = 0;
    int totalNumere = 0;
    int numereLiniaCurenta = 0;
    while(totalNumere < a)
    {
        linie++;
        numereLiniaCurenta++;
        totalNumere += numereLiniaCurenta;
    }
    return linie;
}
```

```
int determinaColoana(int a)
{
    int linie = 0;
    int totalNumere = 0;
    int numereLiniaCurenta = 0;
    while(totalNumere < a)
    {
        linie++;
        numereLiniaCurenta++;
        totalNumere += numereLiniaCurenta;
    }
    int coloana = linie - (totalNumere - a);
    return coloana;
}
```

```
void determinaMatrice(int a[][100],int y)
{
    int linie = 0, totalNumere = 0, numereLiniaCurenta = 0, j;
    while(totalNumere < y)
    {
        linie++, numereLiniaCurenta++;
        for(j=1;j<=numereLiniaCurenta;j++)
            a[linie][j] = totalNumere+j;
        for(j=1;j<=numereLiniaCurenta;j++)
            printf("%d ",a[linie][j]);
        printf("\n");
        totalNumere += numereLiniaCurenta;
    }
}
```

Subject INFO 2016

Enunț subiect INFO 2016

Ionuț tocmai a terminat liceul și susține examenul de admitere la facultate. Știind că s-a pregătit foarte bine pentru examen, el dorește să își anunțe reușita după examen printr-o postare pe Facebook. Ionuț cunoaște n utilizatori reprezentați de numerele de la 1 la n , între care există m relații de prietenie de forma $i\ j$, unde i și j sunt utilizatori, iar n și m sunt numere naturale nenule. Un utilizator nu poate fi prieten cu el însuși, iar o relație de prietenie între doi utilizatori ne spune că fiecare dintre ei este prieten cu celălalt.

Întrucât dorește ca postarea lui să fie cât mai răspândită, Ionuț vrea să afle care sunt utilizatorii cei mai bine conectați din mulțimea sa de cunoscuți, pentru ca eventual să le ceară prietenia. Pentru aceasta, Ionuț trebuie să găsească cea mai mare submulțime de utilizatori cunoscuți, în care fiecare utilizator din această submulțime are cel puțin k prieteni aflați la rândul lor în submulțime, unde k este un număr natural nenul.

Fiind date la intrare numerele n , m și k , pe aceeași linie, separate prin spațiu, precum și $2m$ numere naturale cuprinse între 1 și n , pe o linie nouă, separate prin spațiu, reprezentând în ordine cele m relații de prietenie între cei n utilizatori, ajutați-l pe Ionuț să găsească o soluție la problema sa, rezolvând următoarele subpuncte:

- Să se determine și să se afișeze, în ordine de la 1 la n , numărul de prieteni al fiecărui dintre cei n utilizatori, conform relațiilor de prietenie date.
- Să se determine și să se afișeze, printr-o soluție de complexitate timp cât mai bună, în funcție de datele de intrare, membrii celei mai mari submulțimi de utilizatori, cu proprietatea că fiecare utilizator din această submulțime are cel puțin k prieteni aflați la rândul lor în submulțime. În cazul în care nu există o astfel de submulțime pentru k dat, răspunsul va fi cuvântul NU.

Enunț subiect INFO 2016

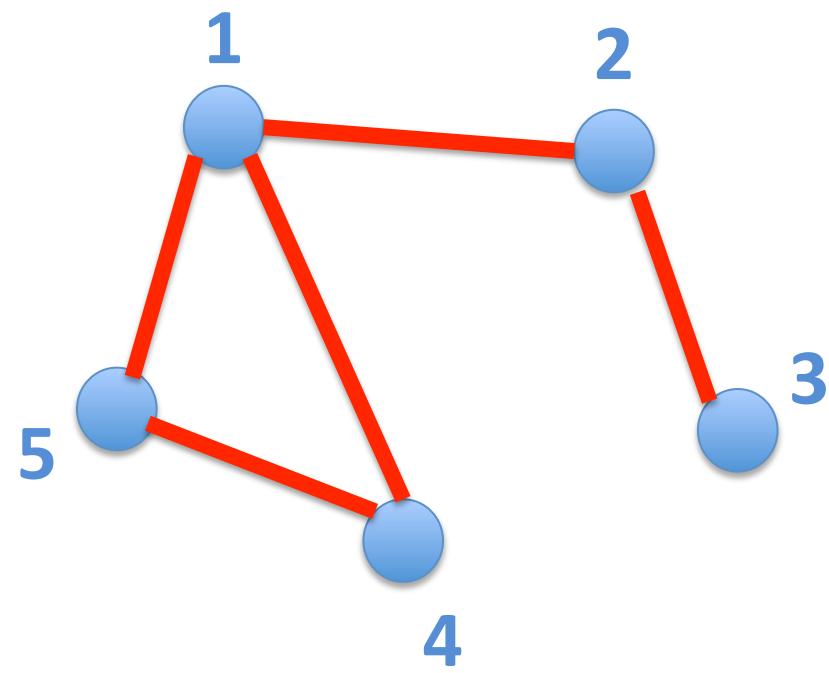
Exemplu:

Date de intrare	Date de ieșire
5 5 2 1 2 5 1 3 2 4 5 1 4	a) 3 2 1 2 2 b) 1 4 5
5 5 3 1 2 5 1 3 2 4 5 1 4	a) 3 2 1 2 2 b) NU
11 18 3 1 8 4 7 7 10 11 10 2 1 2 3 8 9 8 3 9 3 9 2 5 6 5 11 1 4 10 6 7 6 2 8 11 7 11 6	a) 3 4 3 2 2 4 4 4 4 3 3 4 b) 2 3 6 7 8 9 10 11

Rezolvare subiect INFO 2016

Exemplu:

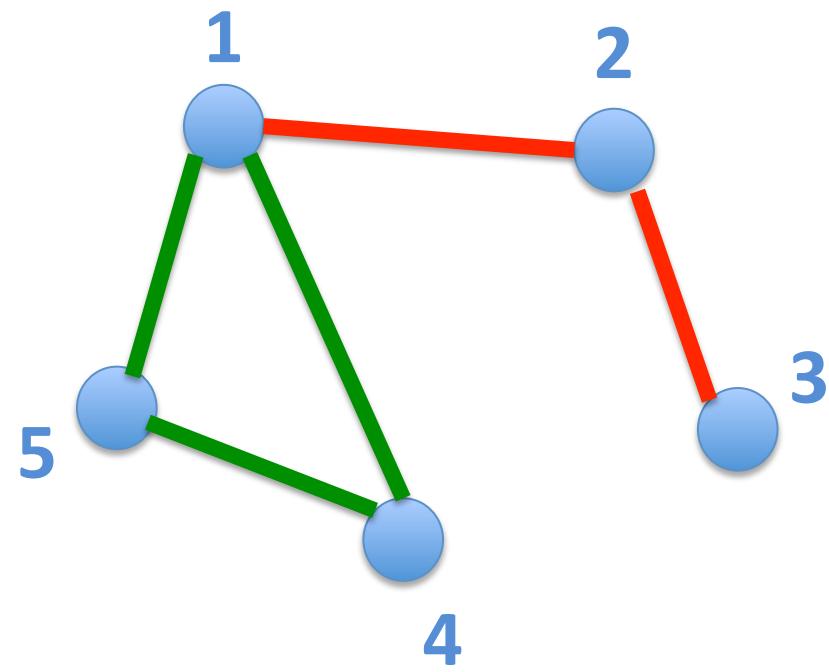
Date de intrare	Date de ieșire
5 5 2 1 2 5 1 3 2 4 5 1 4	a) 3 2 1 2 2 b) 1 4 5
5 5 3 1 2 5 1 3 2 4 5 1 4	a) 3 2 1 2 2 b) NU
11 18 3 1 8 4 7 7 10 11 10 2 1 2 3 8 9 8 3 9 3 9 2 5 6 5 11 1 4 10 6 7 6 2 8 11 7 11 6	a) 3 4 3 2 2 4 4 4 3 3 4 b) 2 3 6 7 8 9 10 11



Rezolvare subiect INFO 2016

Exemplu:

Date de intrare	Date de ieșire
5 5 2 1 2 5 1 3 2 4 5 1 4	a) 3 2 1 2 2 b) 1 4 5
5 5 3 1 2 5 1 3 2 4 5 1 4	a) 3 2 1 2 2 b) NU
11 18 3 1 8 4 7 7 10 11 10 2 1 2 3 8 9 8 3 9 3 9 2 5 6 5 11 1 4 10 6 7 6 2 8 11 7 11 6	a) 3 4 3 2 2 4 4 4 3 3 4 b) 2 3 6 7 8 9 10 11



Grad varf 1 = 3

Grad varf 2 = 2

Grad varf 3 = 1

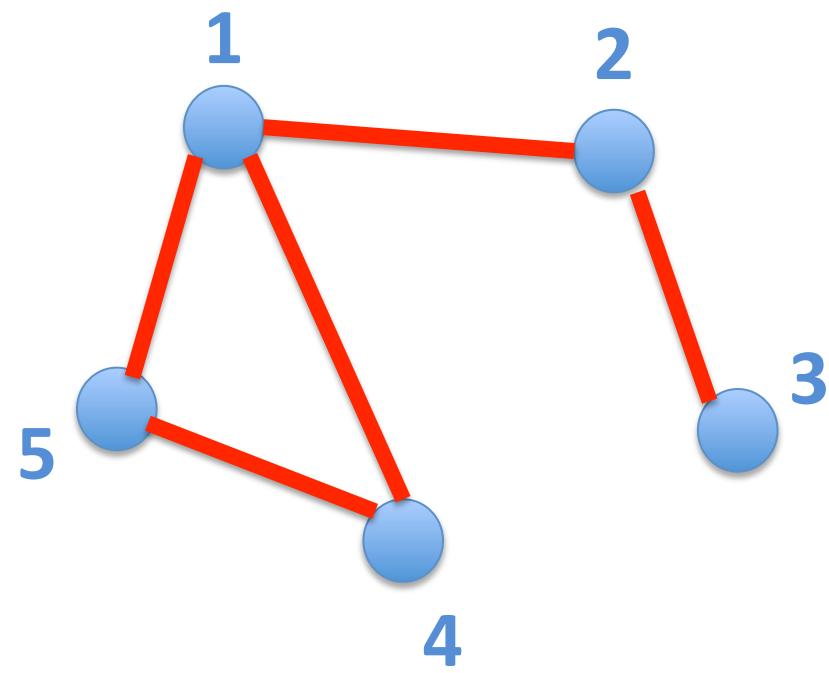
Grad varf 4 = 2

Grad varf 5 = 2

Rezolvare subiect INFO 2016

Exemplu:

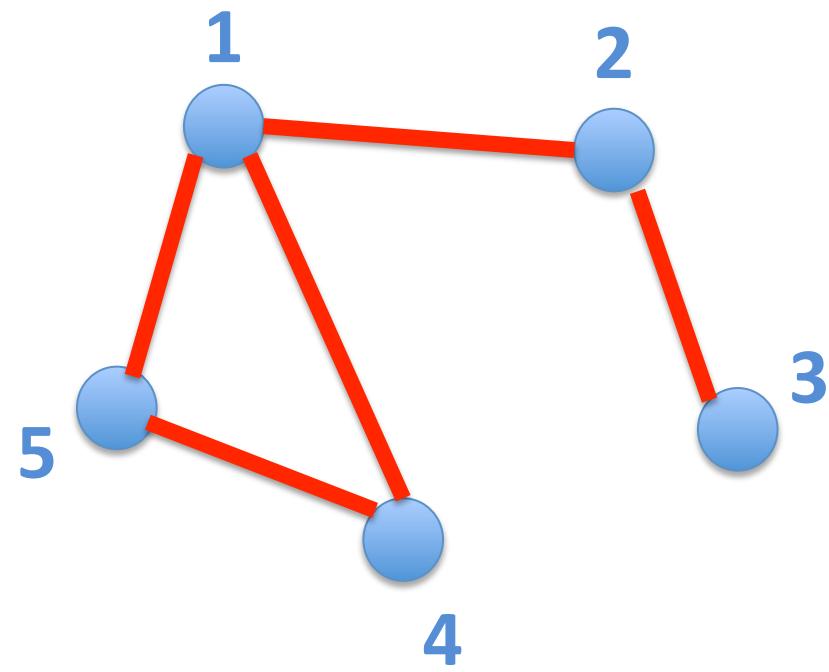
Date de intrare	Date de ieșire
5 5 2 1 2 3 1 3 2 4 5 1 4	a) 3 2 1 2 2 b) 1 4 5
5 5 3 1 2 5 1 3 2 4 5 1 4	a) 3 2 1 2 2 b) NU
1 1 1 6 3 1 8 4 7 7 10 11 10 2 1 2 3 8 9 8 3 9 3 9 2 5 6 5 1 1 1 4 1 0 6 7 6 2 8 1 1 7 1 1 6	a) 3 4 3 2 2 4 4 4 3 3 4 b) 2 3 6 7 8 9 10 11



Rezolvare subiect INFO 2016

Exemplu:

Date de intrare	Date de ieșire
5 5 2	a) 3 2 1 2 2
1 2 5 1 3 2 4 5 1 4	b) 1 4 5
5 5 3	a) 3 2 1 2 2
1 2 5 1 3 2 4 5 1 4	b) NU
11 18 3	a) 3 4 5 2 2 4 4 4 5 5 4
1 8 4 7 7 10 11 10 2 1 2 3 8 9 8 3 9 3 9 2 5 6 5 11 1 4 10 6 7 6 2 8 11 7 11 6	b) 2 3 6 7 8 9 10 11



Grad varf 1 = 3

Grad varf 2 = 3

Grad varf 3 = 2

Grad varf 4 = 2

Grad varf 5 = 2

Rezolvare subiect INFO 2016

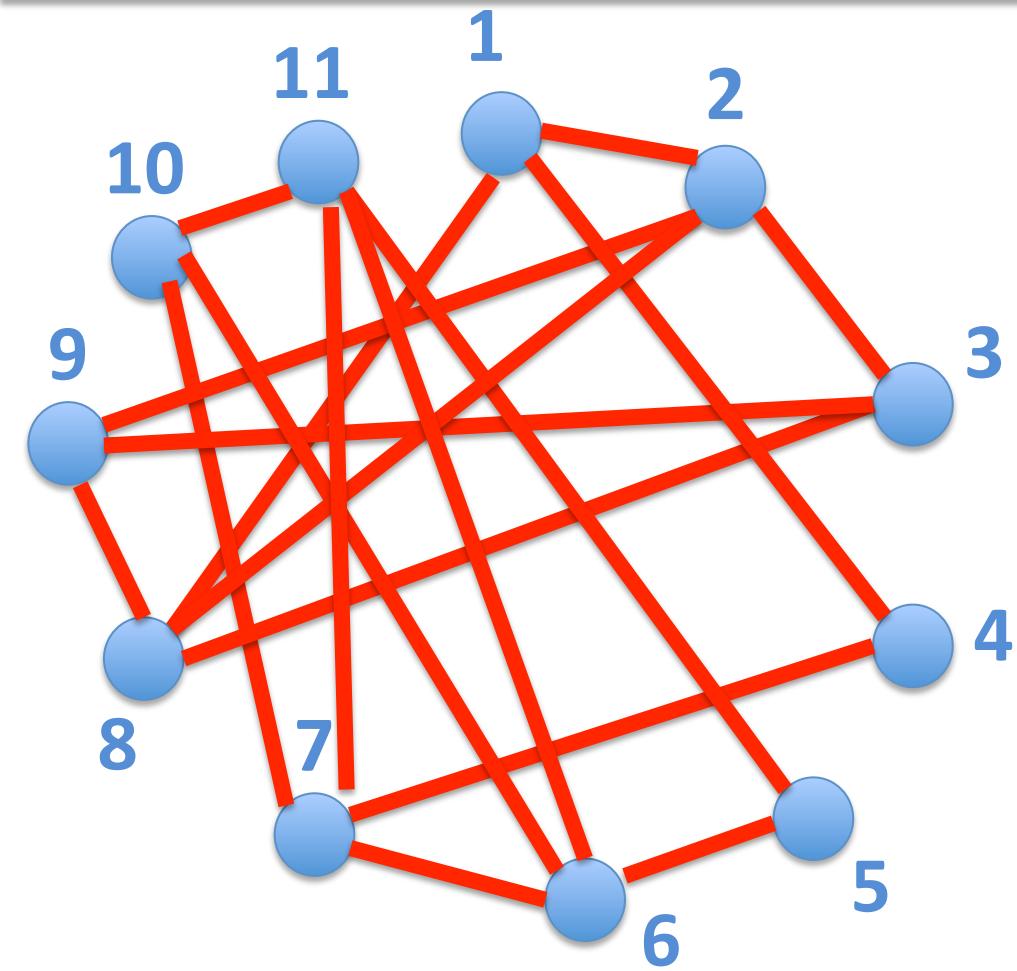
11 18 3

18 4 7 7 10 11 10 2 1 2 3 8 9 8 3 9 3 9 2 5 6 5 11 1 4 10 6 7 6 2 8 11 7 11 6

DJ NU

a) 3 4 3 2 2 4 4 4 3 3 4

b) 2 3 6 7 8 9 10 11



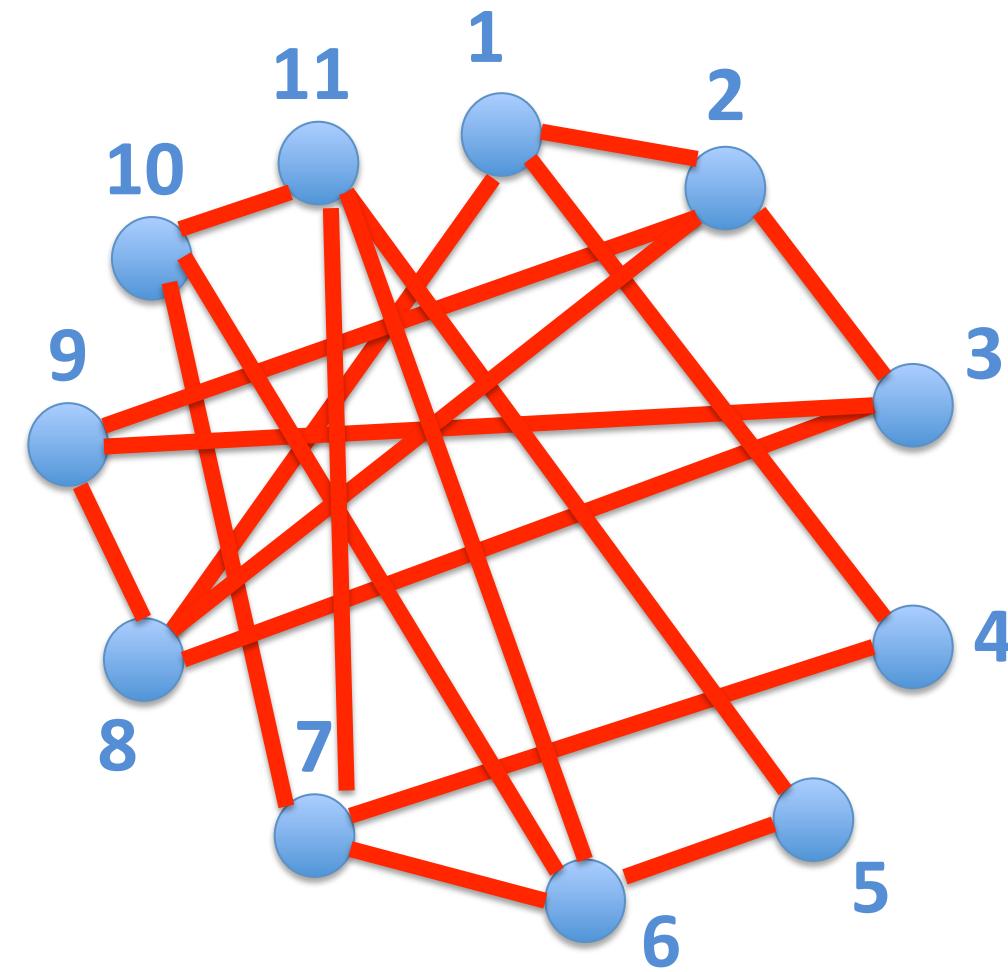
Rezolvare subiect INFO 2016

1 2 3 1 3 2 4 3 1 4

11 18 3

1 8 4 7 7 10 11 10 2 1 2 3 8 9 8 3 9 3 9 2 5 6 5 11 1 4 10 6 7 6 2 8 11 7 11 6

Varf	Grad
a)	3 4 3 2 2 4 4 4 3 3 4
b)	2 3 6 7 8 9 10 11



Grad varf 1 = 3

Grad varf 2 = 4

Grad varf 3 = 3

Grad varf 4 = 2

Grad varf 5 = 2

Grad varf 6 = 4

Grad varf 7 = 4

Grad varf 8 = 4

Grad varf 9 = 3

Grad varf 10 = 3

Grad varf 11 = 4

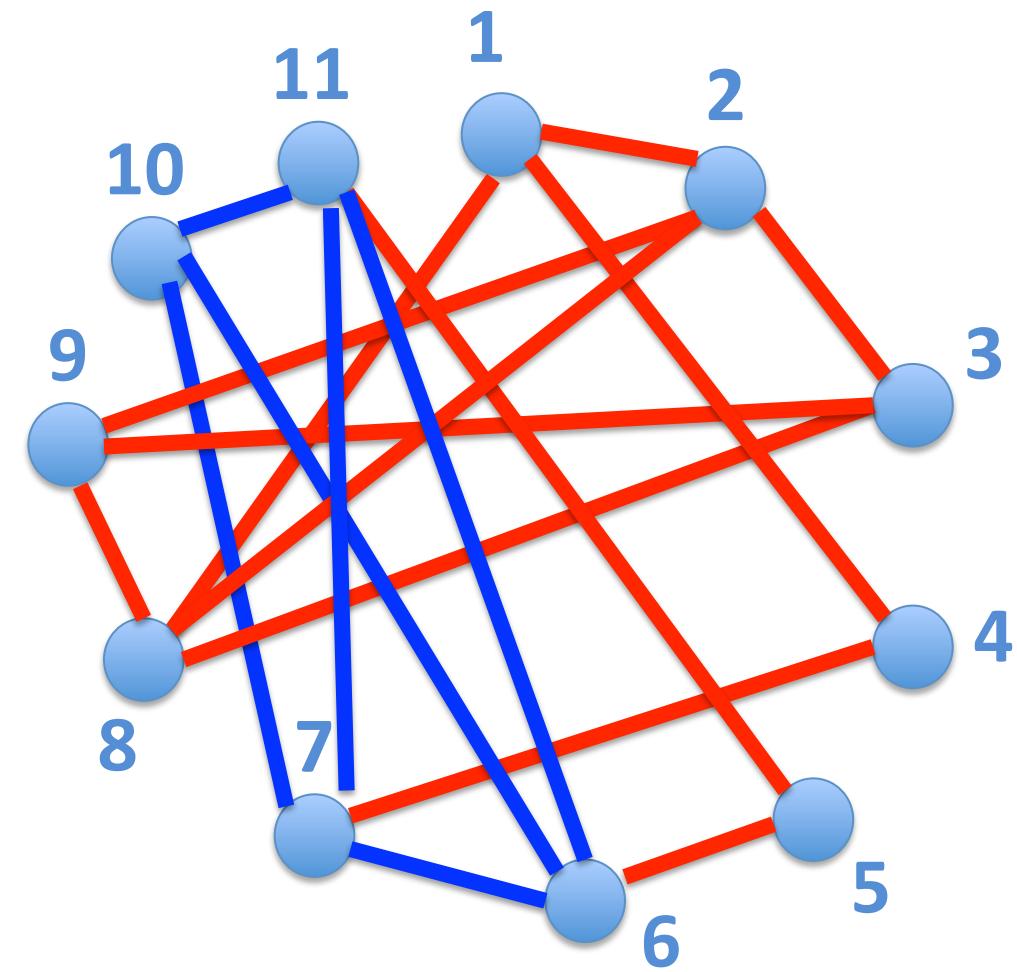
Rezolvare subiect INFO 2016

1 2 3 1 3 2 4 3 1 4

11 18 3

1 8 4 7 7 10 11 10 2 1 2 3 8 9 8 3 9 3 9 2 5 6 5 11 1 4 10 6 7 6 2 8 11 7 11 6

Set A	Set B
a) 3 4 3 2 2 4 4 4 3 3 4	
b) 2 3 6 7 8 9 10 11	



{6, 7, 10, 11}
indeplineste
conditiile.

E cea mai mare
submultime?

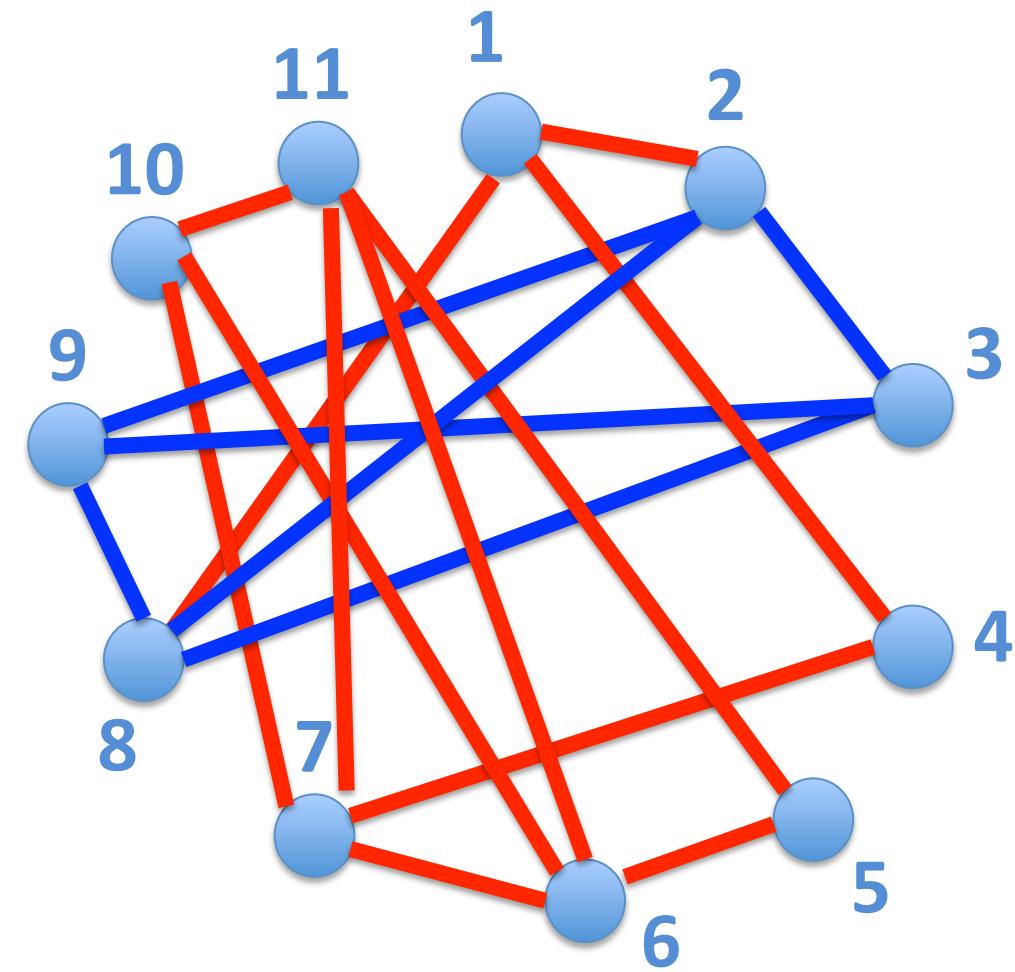
Rezolvare subiect INFO 2016

1 2 3 1 3 2 4 3 1 4

11 18 3

1 8 4 7 7 10 11 10 2 1 2 3 8 9 8 3 9 3 9 2 5 6 5 11 1 4 10 6 7 6 2 8 11 7 11 6

Număr	Rez.
a)	3 4 3 2 2 4 4 4 3 3 4
b)	2 3 6 7 8 9 10 11



{2, 3, 8, 9}
indeplineste
conditiile.

E cea mai mare
submultime?

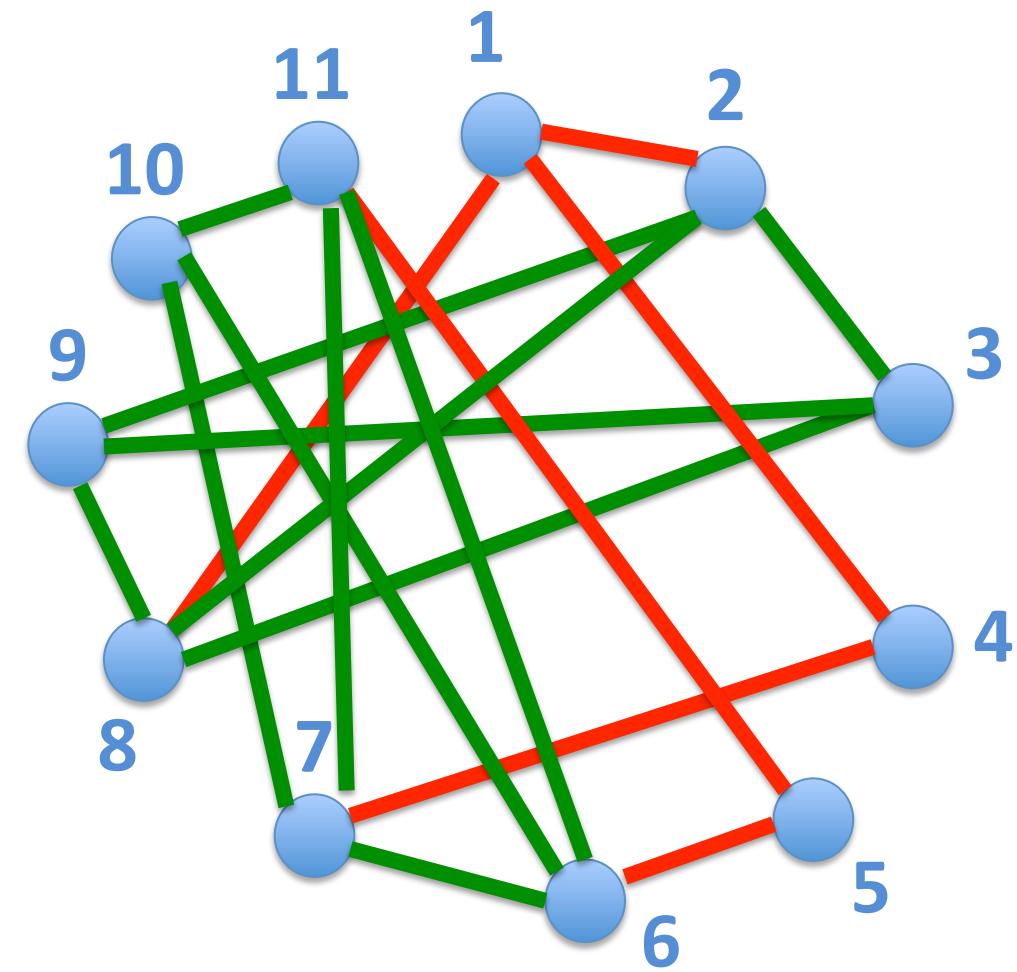
Rezolvare subiect INFO 2016

1 2 3 1 3 2 4 3 1 4

11 18 3

1 8 4 7 7 10 11 10 2 1 2 3 8 9 8 3 9 3 9 2 5 6 5 11 1 4 10 6 7 6 2 8 11 7 11 6

obiect	rezolvare
a)	3 4 3 2 2 4 4 4 3 3 4
b)	2 3 6 7 8 9 10 11



{2, 3, 6, 7, 8, 9, 10,
11} indeplineste
conditiile si e cea
mai mare
submultime

Barem subiect INFO 2016

IV. Informatică. Oficiu	1 p
Modelarea enunțului sub formă de graf	1 p
(a) Orice soluție corectă	2 p
(b) Implementarea corectă a relațiilor de prietenie cu ajutorul unei structuri de adiacență	1 p
Orice soluție corectă, indiferent de complexitate	1 p
la care se adaugă pentru o soluție de complexitate $O(n^2)$	1 p
și pentru o soluție de complexitate mai bună decât $O(n^2)$	1 p
Programele nu au greșeli de limbaj	1 p
Claritatea rezolvărilor	1 p

Rezolvare subiect INFO 2016 – punctul a

Reprezint problema ca graf si aflu gradul fiecarui varf
(gradul unui varf = cati prieteni are)

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 struct node {
5     int inf;
6     struct node *pnext;
7 };
8
9
10 int n, m, k;
11 struct node *v[100];
12 int viz[100];
13 int grad[100];
```

Rezolvare subiect INFO 2016 – punctul a

```
55 void main() {
56
57     int i, a, b;
58
59     printf("introduceti numarul de noduri al grafului:\n");
60     scanf("%d", &n);
61     printf("introduceti numarul de muchii al grafului:\n");
62     scanf("%d", &m);
63
64     for (i = 1; i <= n; i++) {
65         v[i] = NULL;
66         grad[i] = 0;
67         viz[i] = 0;
68     }
69
70     printf("introduceti muchiile prin perechile de numere ce reprezinta capetele muchiilor:\n");
71
72     for (i = 1; i <= m; i++) {
73         scanf("%d%d", &a, &b);
74         v[a] = add(v[a], b); // adaug in lista de adiacenta a varfului a pe varful b
75         v[b] = add(v[b], a); // adaug in lista de adiacenta a varfului b pe varful a
76         grad[a]++;
77         grad[b]++;
78     }
79
80     printf("introduceti gradul k:\n");
81     scanf("%d", &k);
82
83     for (i = 1; i <= n; i++)
84         printf("%d ", grad[i]);
85     printf("\n");
86 }
```

Rezolvare subiect INFO 2016 – punctul b

Reprezint problema ca graf si relatia de pretenie prin liste de adiacenta (alternativa: matrice de adiacenta).

Elimin toate varfurile cu grad mai mic decat k si muchiile lor din graf.

Toate varfurile care raman fac parte din submultimea dorita.

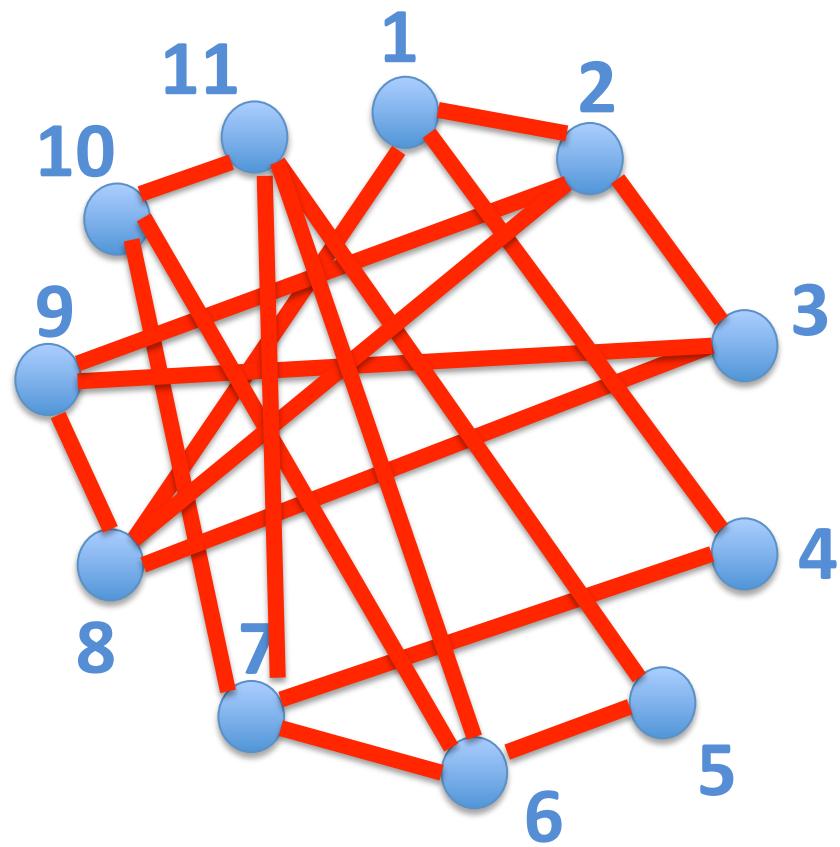
Rezolvare subiect INFO 2016 – punctul b

1 2 3 1 3 2 4 3 1 4	D)	I	U
11 18 3	a)	3 4 3 2 2 4 4 4 3 3 4	
1 8 4 7 7 10 11 10 2 1 2 3 8 9 8 3 9 3 9 2 5 6 5 11 1 4 10 6 7 6 2 8 11 7 11 6	b)	2 3 6 7 8 9 10 11	

v[1]



Lista de adiacenta
pentru varful 1



Rezolvare subiect INFO 2016 – punctul b

Reprezent problema ca graf si relatia de pretenie prin liste de adiacenta.

```
16  struct node * add(struct node *h,int x) {
17
18      struct node *p, *pc, *rez;
19
20      p = malloc(sizeof(struct node));
21      p->inf = x;
22      p->pnext = NULL;
23      if (h == NULL)
24          rez = p;
25      else {
26          pc = h;
27          while (pc->pnext != NULL) {
28
29              pc = pc->pnext;
30          }
31          pc->pnext = p;
32          rez = h;
33      }
34      return rez;
35 }
```

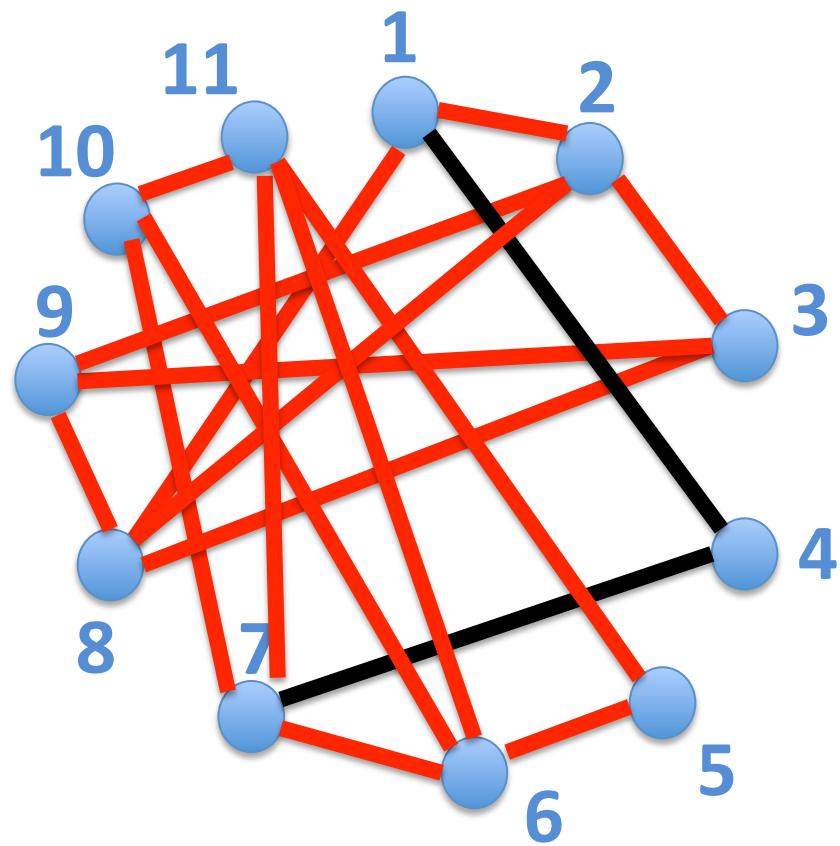
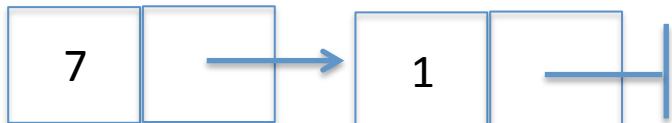
Rezolvare subiect INFO 2016 – punctul b

Elimin toate varfurile cu grad mai mic decat k si muchiile lor din graf – parcurgere in adancime a grafului si marcare ca vizitat a nodului care se elimina

```
37 void df(int x) {
38
39     struct node *pc;
40
41     if ((viz[x] == 0) && (grad[x] < k)) {
42
43         viz[x] = 1;
44         pc = v[x];
45         while (pc != NULL) {
46             grad[pc->inf]--;
47             if (grad[pc->inf] < k) df(pc->inf);
48             pc = pc->pnext;
49         }
50
51     }
52
53 }
```

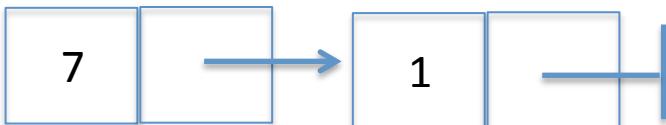
Rezolvare subiect INFO 2016 – punctul b

v[4]

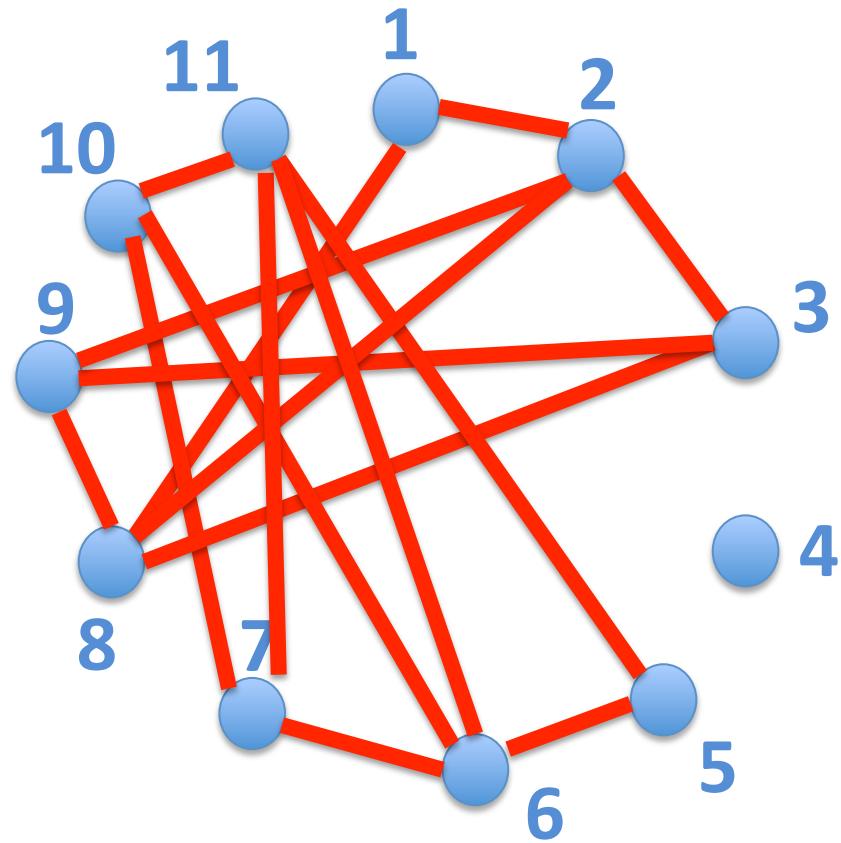


Rezolvare subiect INFO 2016 – punctul b

v[4]

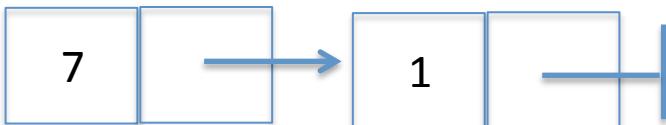


v[1]

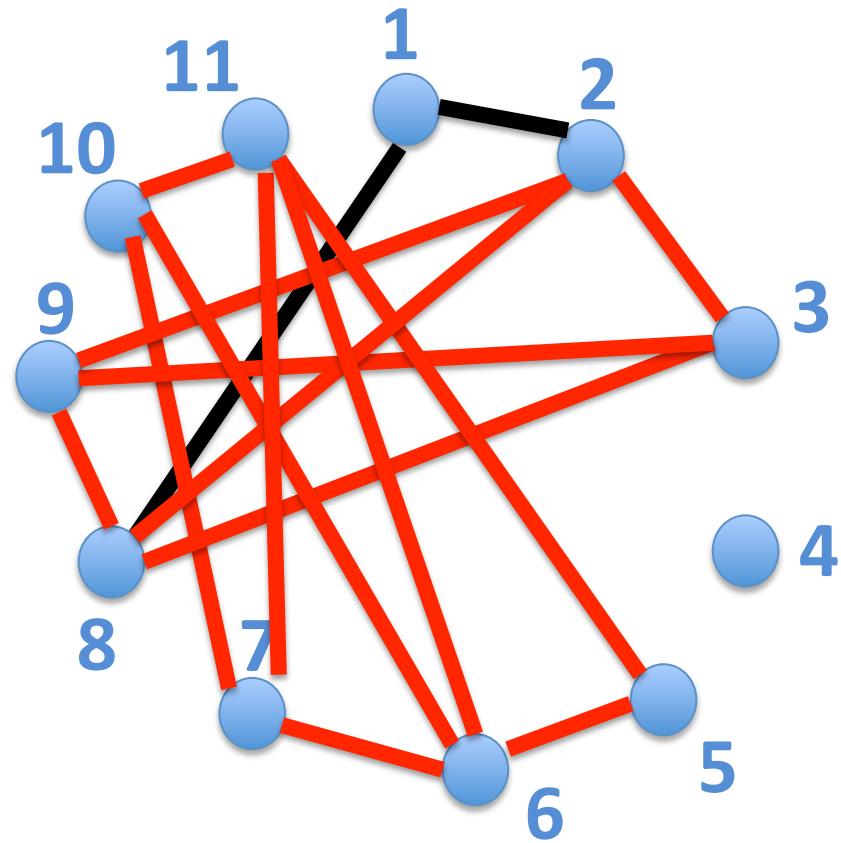


Rezolvare subiect INFO 2016 – punctul b

v[4]

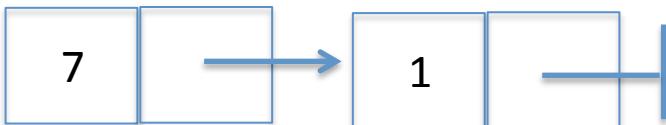


v[1]

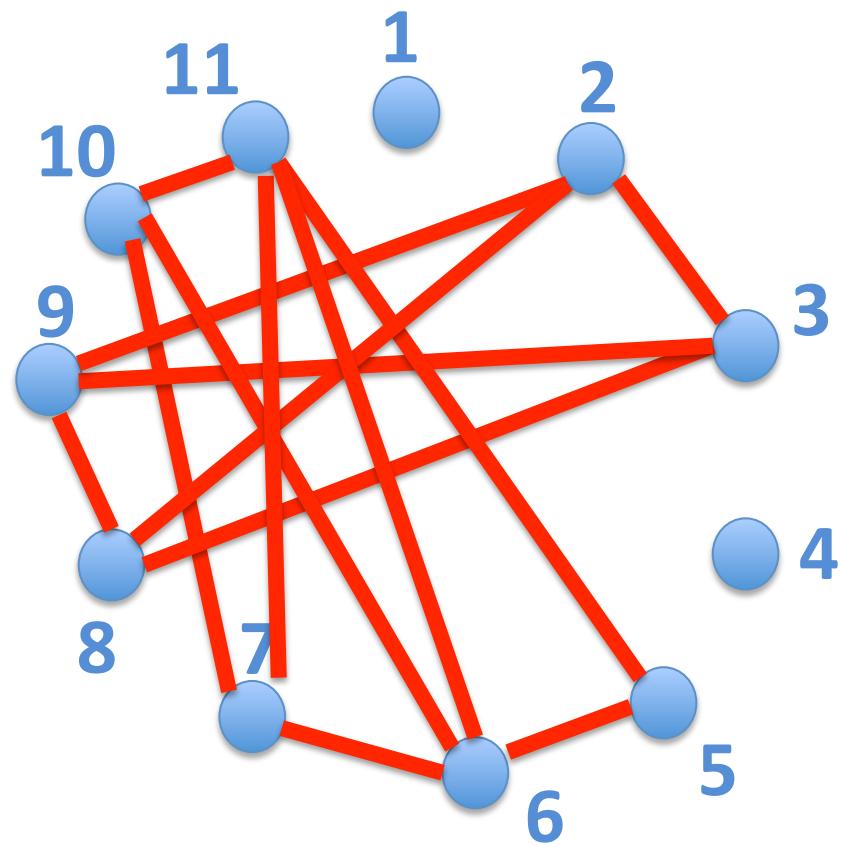


Rezolvare subiect INFO 2016 – punctul b

v[4]



v[1]



Rezolvare subiect INFO 2016 – punctul b

Reprezint problema ca graf si relatia de pretenie prin liste de adiacenta.

Elimin toate varfurile cu grad mai mic decat k si muchiile lor din graf.

Toate varfurile care raman fac parte din submultimea dorita.

```
80 printf("introduceti gradul k:\n");
81 scanf("%d", &k);
82
83 for (i = 1; i <= n; i++)
84     printf("%d ", grad[i]);
85 printf("\n");
86
87 for (i = 1; i <= n; i++) {
88     if ((viz[i] == 0) && (grad[i] < k)) {
89         df(i);
90     }
91 }
92 for (i = 1; i <= n; i++)
93     if(viz[i] == 0) printf("%d ", i);
94 printf("\n");
95
96
```

$O(n+m)$

Subject MATE 2016

Enunț subiect MATE 2016

IV. Informatică.

Se dă un sir de n numere întregi, cu n număr natural nenul, mai mic decât 32000. Se elimină primul element din sir și toate elementele sirului aflate pe poziții care reprezintă numere prime, în ordinea crescătoare a pozițiilor. Operația se repetă cu elementele rămase în sir, reposiționate după eliminarea celorlalte, până când este eliminat și ultimul element rămas. Să se scrie un program care afișează elementele sirului inițial, în ordinea în care au fost eliminate conform algoritmului descris mai sus.

Exemplul 1. Pentru $n = 10$ și sirul 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 programul va afișa 1 2 3 5 7 4 6 8 10 9.

Exemplul 2. Pentru $n = 20$ și sirul 4, 23, 16, -7, 89, 115, 23, 11, 15, 2, -8, -9, 21, 0, 75, 23, 32, -1, 4, 5 programul va afișa 4 23 16 89 23 -8 21 32 4 -7 115 11 2 0 5 15 -9 75 -1 23.

Notă: Programele vor fi scrise într-unul dintre limbajele de programare studiate în liceu (Pascal, C, C++). Pentru fiecare soluție se vor descrie informal detaliile algoritmului folosit și ale implementării sub formă de program: semnificația variabilelor, a structurilor de date, a structurilor repetitive, a instrucțiunilor condiționale.

Barem subiect MATE 2016

IV. Informatică.	Oficiu	1 p
Generarea tuturor numerelor prime până la n	2 p	
Eliminarea corectă a elementelor pentru o iterație	2 p	
Efectuarea tuturor iterățiilor pentru obținerea soluției corecte	3 p	
Programele nu au greșeli de limbaj	1 p	
Claritatea rezolvărilor	1 p	

Rezolvare subiect MATE 2016

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int ePrim(int nr)
5 {
6     if (nr<=1)
7         return 0;
8     int i;
9     for(i=2;i<=sqrt(nr);i++)
10        if(nr%i==0)
11            return 0;
12     return 1;
13 }
```

```
14 int main()
15 {
16     int n;
17     printf("n=");scanf("%d",&n); //citesc n
18     //citesc secheta de n numere, pun al- i-lea pe pozitia i
19     int v[n+1],i;
20     for(i=1;i<=n;i++)
21         scanf("%d",&v[i]);
22     int prime[n+1];
23     //generez pozitiile prime pana la n dupa enunt
24     prime[1] = 1;//prima pozitie e buna
25     for(i=2;i<=n;i++)
26         prime[i] = ePrim(i);
27     int vNrSterse[n+1],nrSterse = 0;
28     int vNrRamase[n+1];
29     int nInitial = n;
30     while(1)
31     {
32         int nrRamase = 0;
33         if (n<=3)
34         {
35             for(i=1;i<=n;i++)
36                 vNrSterse[++nrSterse] = v[i];
37             break;
38         }
39         for(i=1;i<=n;i++)
40             if(prime[i])
41                 vNrSterse[++nrSterse] = v[i];
42             else
43                 vNrRamase[++nrRamase]=v[i];
44         for(i=1;i<=nrRamase;i++)
45             v[i] = vNrRamase[i];
46         n = nrRamase;
47     }
48     //afiseaza numerele in ordinea stergerii
49     for(i=1;i<=nInitial;i++)
50         printf("%d \n",vNrSterse[i]);
```

Subject INFO 2015

Enunț subiect INFO 2015

IV. Informatică. Se citesc numerele naturale nenule a, b, c, n , urmate de o secvență de n numere naturale distințe, notată cu s .

- (a) Să se scrie un program care afișează toate perechile (x, y) cu proprietatea că x și y sunt numere diferite din secvența s , care verifică ecuația $ax^2 + by^2 = c$.

Exemplu: Dacă programul citește la intrare 1 1 25 5 3 18 5 0 4, atunci afișează perechile (3,4) (4,3) (0,5) (5,0), nu neapărat în această ordine.

- (b) Dacă secvența s citită la intrare este formată din numere în ordine crescătoare, să se scrie un program cât mai eficient care afișează numărul de perechi (x, y) cu proprietatea de la punctul (a). Să se calculeze complexitatea timp a soluției prezentate.

Exemplu: Dacă programul citește la intrare 1 1 25 5 0 3 4 5 18, atunci afișează 4.

Notă: Programele vor fi scrise într-unul dintre limbajele de programare studiate în liceu (Pascal,C,C++). Pentru fiecare soluție se vor descrie informal detaliile algoritmului folosit și ale implementării sub formă de program: semnificația variabilelor, a structurilor de date, a structurilor repetitive, a instrucțiunilor condiționale.

Barem subiect INFO 2015

IV. Informatică. Oficiu	1 p
(a) Afişarea a cel puțin unei perechi (dacă există) cu proprietatea din enunț	1 p
Afișarea doar a unor perechi cu proprietatea din enunț	1 p
Afișarea tuturor perechilor cu proprietatea din enunț	1 p
(b) Afişarea corectă a numărului de soluții, indiferent de complexitate	1 p
Afișarea corectă a numărului de soluții în timp cel mult $O(n \log n)$	1 p
Afișarea corectă a numărului de soluții în timp cel mult $O(n)$	1 p
Calculul corect al complexității timp	1 p
Programele nu au greșeli de limbaj	1 p
Claritatea rezolvărilor	1 p

Rezolvare subiect INFO 2015 – punctul a

```
info2015.c X
1 #include <stdio.h>
2 #include<stdlib.h>
3
4 int verificaConditie(int a, int b, int c,int x, int y)
5 {
6     if (a*x*x + b*y*y == c)
7         return 1;
8     return 0;
9 }
10
11 int main()
12 {
13     int a,b,c,n;
14     scanf("%d",&a);scanf("%d",&b);scanf("%d",&c);scanf("%d",&n);
15     //citesc secventa s
16     int s[n],i;
17     for(i=0;i<n;i++)
18         scanf("%d",&s[i]);
19     //punctul a
20     int j;
21     for(i=0;i<n;i++)
22         for(j=i+1;j<n;j++)
23     {
24         if(verificaConditie(a,b,c,s[i],s[j]))
25             printf("(%d,%d)",s[i],s[j]);
26         if(verificaConditie(a,b,c,s[j],s[i]))
27             printf("(%d,%d)",s[j],s[i]);
28     }
29     printf("\n");
30     return 0;
31 }
32
33 }
```

1
1
25
5
3
18
5
0
4
 $(3, 4)(4, 3)(5, 0)(0, 5)$

$\mathcal{O}(n^2)$

Rezolvare subiect INFO 2015 – punctul b

info2015_b.c

```
1 #include <stdio.h>
2 #include<stdlib.h>
3
4 int gasesteSolutieCautareBinara(int poz, int stanga, int dreapta, int s[], int a, int b, int c)
5 {
6     int mijloc;
7     while(stanga <=dreapta){
8         mijloc = (stanga + dreapta)/2;
9         int valoareExpresie = a*s[poz]*s[poz] + b*s[mijloc]*s[mijloc];
10        if (valoareExpresie == c) //avem solutie?
11        {
12            if (poz == mijloc) //solutie este perechea (i,i) care nu e buna, nr tb sa fie diferite
13                return 0;
14            printf("(%d,%d) ",s[poz],s[mijloc]);
15            return 1;
16        }
17        if(valoareExpresie < c)
18            stanga = mijloc + 1;
19        else
20            dreapta = mijloc - 1;
21    }
22    return 0;
23 }
24
25 int main()
26 {
27     int a,b,c,n;
28     scanf("%d",&a);scanf("%d",&b);scanf("%d",&c);scanf("%d",&n);
29     //citesc secventa s
30     int s[n],i;
31     for(i=0;i<n;i++)
32         scanf("%d",&s[i]);
33     //punctul b
34     int j;
35     int nrPerechiSolutie = 0;
36     for(i=0;i<n;i++)
37         nrPerechiSolutie = nrPerechiSolutie + gasesteSolutieCautareBinara(i,0,n-1,s,a,b,c);
38     printf("NrPerechiSolutie = %d\n",nrPerechiSolutie);
39     return 0;
40 }
```

$$\mathcal{O}(n \cdot \log_2(n))$$

Rezolvare subiect INFO 2015 – punctul b

info2015_b_liniar.c 

```
1 #include <stdio.h>
2 #include<stdlib.h>
3
4 int gasesteSolutieLiniar(int s[], int n, int a, int b, int c)
5 {
6     int nrPerechiSolutie = 0;
7     int stanga = 0, dreapta = n-1;
8     while(dreapta > stanga)
9     {
10         int valoareExpresie = a*s[stanga]*s[stanga] + b*s[dreapta]*s[dreapta];
11         if (valoareExpresie == c)
12         {
13             nrPerechiSolutie++;
14             printf("(%d,%d) ",s[stanga],s[dreapta]);
15             dreapta--;stanga++;
16         }
17         else
18         {
19             if (valoareExpresie > c)
20                 dreapta--;
21             else
22                 stanga++;
23         }
24     }
25     return nrPerechiSolutie;
26 }
27 }
```

Rezolvare subiect INFO 2015 – punctul b

```
28 int main()
29 {
30     int a,b,c,n;
31     scanf("%d",&a);scanf("%d",&b);scanf("%d",&c);scanf("%d",&n);
32     //citesc secenta s
33     int s[n],i;
34     for(i=0;i<n;i++)
35         scanf("%d",&s[i]);
36     //punctul b - colutie liniara
37     int j;
38     int nrPerechiSolutie = 0;
39     //verific solutiile de tipul a*s[stanga]*s[stanga] + b*s[dreapta]*s[dreapta] = c
40     nrPerechiSolutie = gasesteSolutieLiniar(s, n, a, b, c);
41     if (a != b)
42         //verific solutiile de tipul b*s[stanga]*s[stanga] + a*s[dreapta]*s[dreapta] = c
43         nrPerechiSolutie += gasesteSolutieLiniar(s, n, b, a, c);
44     printf("NrPerechiSolutie = %d\n",nrPerechiSolutie);
45     return 0;
46 }
```

--

```
1
2
102
5
1
2
7
9
10
(2,7) (1,10) NrPerechiSolutie = 2
```

$$\mathcal{O}(n)$$

Subject MATE 2015

Enunț subiect MATE 2015

IV. Informatică.

- (a) Să se arate că orice număr natural nenul se poate scrie în mod unic ca o sumă de puteri ale lui 2 care nu se repetă (exemplu: $77 = 2^0 + 2^2 + 2^3 + 2^6$).
- (b) Numerele naturale de la 1 la 255 se codifică astfel:
- puterile lui 2 se reprezintă prin literele: $a = 1, b = 2, c = 4, d = 8, e = 16, f = 32, g = 64, h = 128$;
 - orice alt număr din intervalul menționat va fi reprezentat ca o combinație de aceste litere, aranjate în ordine alfabetică, în care orice literă apare cel mult o singură dată, astfel încât suma valorilor acestor litere să fie egală cu valoarea numărului (exemplu: $acdgh = 77$).

Să se scrie un program care, citind două siruri de caractere ce reprezintă numere în convenția de mai sus, să scrie, la ieșire, sirul ce reprezintă suma numerelor astfel reprezentate (exemplu: dacă la intrare programul primește sirurile $acdgh$ și ac atunci, la ieșire, va scrie beg). Sirurile de intrare sunt alese astfel încât suma numerelor pe care le reprezintă să fie mai mică sau egală cu 255.

Este posibil ca programul să calculeze sirul de ieșire fără a transforma sirurile în numere? Dacă da, dați o astfel de soluție.

Notă: Programele vor fi scrise într-unul dintre limbajele de programare studiate în liceu (Pascal, C, C++). Pentru fiecare soluție se vor descrie informal detaliile algoritmului folosit și ale implementării sub formă de program: semnificația variabilelor, a structurilor de date, a structurilor repetitive, a instrucțiunilor condiționale.

Barem subiect MATE 2015

IV. Informatică.	Oficiu	1 p
(a)	Numărul poate fi scris ca sumă de puteri ale lui 2	1 p
	Există cel puțin o scriere a numărului în care puterile lui 2 nu se repetă	1 p
	Această scriere este unică	1 p
(b)	Programul citește corect șirurile de caractere	1 p
	Programul “adună” șirurile corect	1 p
	Programul afișează corect șirul rezultat	1 p
	Programul nu transformă șirurile în numere ci le “adună” direct	1 p
	Programele nu au greșeli de limbaj	1 p
	Claritatea rezolvărilor	1 p

Rezolvare subiect MATE 2015 – punctul a

1. Existenta: $\exists k_1 > k_2 > \dots > k_n \geq 0$

$$m = 2^{k_1} + 2^{k_2} + \dots + 2^{k_n}$$

62

$$62 = 2^5 + 2^4 + 2^3 + 2^2 + 2^1$$

2. Unicitate: scrierea e unica

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     int m;
7     scanf("%d", &m);
8     int k, primulTermenScris = 0;
9     while(m)
10    {
11        k=0;
12        while (pow(2,k)<=m)
13            k++;
14        k--;
15        if(primulTermenScris==0)
16        {
17            printf("%d=2^%d", m, k);
18            primulTermenScris = 1;
19        }
20        else
21            printf(" + 2^%d", k);
22        m = m - pow(2, k);
23    }
24    printf("\n");
25
26 }
```

Rezolvare subiect MATE 2015 – punctul b

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     char sir1[9],sir2[9];
7     scanf("%s",sir1);
8     scanf("%s",sir2);
9     int i,k,nr1=0,nr2=0,suma;
10
11    for(i=0;i<strlen(sir1);i++)
12        nr1 = nr1 + (int) pow(2,sir1[i]-'a');
13    printf("nr1 = %d\n",nr1);
14
15    for(i=0;i<strlen(sir2);i++)
16        nr2 = nr2 + (int) pow(2,sir2[i]-'a');
17    printf("nr2 = %d\n",nr2);
18 }
```

```
acdg
ac
nr1 = 77
nr2 = 5
01010010
beg
```

```
Process returned 0 (0x0)  execution time : 5.453 s
Press ENTER to continue.
```

```
19 suma = nr1+nr2;
20 int reprezentareBinara[8]={0,0,0,0,0,0,0,0};
21 while(suma)
22 {
23     k=0;
24     while (pow(2,k)<=suma)
25         k++;
26     k--;
27     reprezentareBinara[k]=1;
28     suma = suma - pow(2,k);
29 }
30
31 for(i=0;i<8;i++)
32     printf("%d",reprezentareBinara[7-i]);
33 printf("\n");
34
35 for(i=0;i<8;i++)
36     if(reprezentareBinara[i])
37         printf("%c",'a'+i);
38     printf("\n");
39
40 }
41 }
```

Rezolvare subiect MATE 2015 – punctul b

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     char sir1[9],sir2[9];
7     scanf("%s",sir1);
8     scanf("%s",sir2);
9     int i;
10    int v1[8]={0,0,0,0,0,0,0,0},v2[8]={0,0,0,0,0,0,0,0},v3[8]={0,0,0,0,0,0,0,0};
11    for(i=0;i<strlen(sir1);i++)
12        v1[sir1[i]-'a']=1;
13    for(i=0;i<strlen(sir2);i++)
14        v2[sir2[i]-'a']=1;
15    int rest = 0;
16    for(i = 0;i<8;i++)
17    {
18        v3[i] = (v1[i] + v2[i] + rest)%2;
19        rest = (v1[i] + v2[i] + rest)/2;
20        if(v3[i])
21            printf("%c",'a' + i);
22    }
23    return 0;
24 }
```

abc
ade
f
-

Subject INFO 2014

Enunț subiect INFO 2014

IV. Informatică. Se dă operația $\bar{-} : \{1, 2\} \rightarrow \{1, 2\}$ astfel încât $\bar{1} = 2$ și $\bar{2} = 1$. Operația se extinde asupra oricărei secvențe formate cu cifre de 1 și 2, de exemplu $\overline{1211212121} = 2122121212$. Se consideră sirul infinit s format cu cifre de 1 și 2, generat incremental prin extindere după următoarea regulă de concatenare: $s_1 = 1221$, $s_2 = 1221211221121221$, ..., $s_{k+1} = s_k \overline{s_k s_k} s_k$, ..., pentru orice număr natural nenul k .

Fie n un număr natural nenul, $n < 1000000$.

- (a) Să se scrie un program care citește n și afișează primele n cifre ale sirului s .

Exemplu: Pentru $n = 18$ programul va afișa 122121122112122121.

- (b) Să se scrie un program care citește n și afișează a n -a cifră a sirului s , astfel încât numărul de pași ai programului să fie proporțional cu $\log_2 n$ (complexitate timp logaritmică în funcție de n).

Exemplu: Pentru $n = 11$ programul va afișa 1, iar pentru $n = 20$ programul va afișa 2.

Notă: Programele vor fi scrise într-unul dintre limbajele de programare studiate în liceu (Pascal,C,C++). Pentru fiecare soluție se vor descrie informal detaliile algoritmului folosit și ale implementării sub formă de program: semnificația variabilelor, a structurilor de date, a structurilor repetitive, a instrucțiunilor condiționale.

Timp de lucru 3 ore.

Barem subiect INFO 2014

- IV. Informatică. Oficiu 1 punct
- (a) • Sirul s este generat corect 3 puncte
 • Programul afișează exact n cifre 1 punct
 • Cifrele afișate sunt corecte în raport cu sirul s 1 punct
- (b) • Programul afișează corect a n -a cifră, într-o complexitate mai mică decât $O(n)$ 1 punct
 • Programul afișează corect a n -a cifră, într-o complexitate egală cu $O(\log n)$ 1 punct
– Programele nu au greșeli de limbaj 1 punct
– Claritatea rezolvărilor 1 punct

Rezolvare subiect INFO 2014 – punctul a

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     int n, l, i, k;
7     scanf("%d", &n);
8
9     //s_k are 4^k cifre
10    k = 1;
11    while(pow(4, k) < n)
12        k++;
13    printf("s_k are %d cifre\n", (int) pow(4, k));
14    char x[(int) pow(4, k)];
15
16    x[0] = 1; x[1] = 2; x[2] = 2; x[3] = 1;
17
18    l = 4;
19    while (l < n) {
20        for (i = 0; i < l; i++) x[l + i] = 3 - x[i];
21        for (i = 0; i < l; i++) x[2 * l + i] = 3 - x[i];
22        for (i = 0; i < l; i++) x[3 * l + i] = x[i];
23        l *= 4;
24    }
25    for (i = 0; i < n; i++) printf("%d", x[i]);
26    printf("\n");
27    return 0;
28 }
```

18
s_k are 64 cifre
122121122112122121

Rezolvare subiect INFO 2014 – punctul a

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 int main()
6 {
7     int n, l, i;
8     scanf("%d", &n);
9     char x[10000];
10    x[0] = '1'; x[1] = '2'; x[2] = '2'; x[3] = '1'; x[4] = '\0';
11
12    char *tmp, *tmpb, *rez;
13    rez = (char*) malloc(n + 1);
14    l = strlen(x);
15    while (l < n) {
16        tmp = (char*) malloc(1 + l); strcpy(tmp, x);
17        tmpb = (char*) malloc(1 + l); strcpy(tmpb, x);
18        for (i = 0; i < l; i++) {
19            if (tmpb[i] == '1') tmpb[i] = '2';
20            else tmpb[i] = '1';
21        }
22        strcat(x, tmpb); strcat(x, tmpb); strcat(x, tmp); l = strlen(x);
23    }
24    strncpy(rez, x, n);
25    rez[n] = '\0';
26    printf("%s\n", rez);
27    return 0;
28 }
```

Rezolvare subiect INFO 2014 – punctul b

```
1 #include <stdio.h>
2
3 int val_elem(int n) {
4     int p, k, tmp, inv;
5     if (n < 4) {
6         if ((n == 0) || (n == 3)) return 1;
7         if ((n == 1) || (n == 2)) return 2;
8     }
9     else {
10        k = 1;p = 4;
11        while (p <= n) { p *= 4; k++;}
12        k--;p = p / 4;
13        inv = n / p;
14        tmp = val_elem(n % p);
15        if ((inv == 1) || (inv == 2)) return 3 - tmp;
16        else return tmp;
17    }
18}
19
20 int main()
21 {
22     int n;
23     scanf("%d", &n);
24     printf("%d\n", val_elem(n-1));
25     return 0;
26 }
```

Subject MATE 2014

Enunț subiect MATE 2014

IV. Informatică.

Se consideră ecuația de gradul al 2-lea cu coeficienți reali $ax^2+bx+c = 0$ cu $a \neq 0$ și expresia: $S_n = x_1^n + x_2^n$, unde x_1 și x_2 sunt rădăcinile ecuației. Să se scrie un program care primind coeficienții a, b, c ai ecuației și un număr natural n calculează și afișază valoarea expresiei S_n , știind că S_n este un număr real indiferent dacă rădăcinile ecuației sunt reale sau nu. De exemplu, dacă programul va primi la intrare numerele: 1 1 1 6 (ceea ce înseamnă că ecuația este $x^2 + x + 1 = 0$ și se cere S_6) va afișa 2.

Notă: Programele vor fi scrise într-unul dintre limbajele de programare studiate în liceu (Pascal,C,C++). Pentru fiecare soluție se vor descrie informal detaliile algoritmului folosit și ale implementării sub formă de program: semnificația variabilelor, a structurilor de date, a structurilor repetitive, a instrucțiunilor condiționale.

Barem subiect MATE 2014

IV. Informatică.	Oficiu	1 punct
•	Calculul lui S_1 și S_2	1 punct
•	Găsirea relației de recurență pentru S_n	3 puncte
•	Implementarea corectă a relației de recurență	3 puncte
•	Programele nu au greșeli de limbaj	1 punct
•	Claritatea rezolvărilor	1 punct

Rezolvare subiect MATE 2014

mate2014.c

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int a,b,c,n,i;
6     scanf("%d",&a);
7     scanf("%d",&b);
8     scanf("%d",&c);
9     scanf("%d",&n);
10
11     double S[n+1];
12     S[0] = 2;
13     S[1] = -b/a;
14     for(i=2;i<=n;i++)
15         S[i] = (-b/a) * S[i-1] -(c/a)*S[i-2];
16     printf("S[%d] = %f \n",n,S[n]);
17     return 0;
18 }
```

```
. . . . .
1 1 1 6
S[6] = 2.000000
Process returned 0 (0x0)
Press ENTER to continue.
```

Subject INFO 2013

Enunț subiect INFO 2013

IV. Informatică. Fie n un număr natural nenul și $m = 2^n$. Se dă vectorul $0, 1, 2, 3, \dots, m, m + 1$ și p , cu $1 \leq p \leq m$. În acest vector, marcăm numerele $0, p$ și $m + 1$ ca fiind șterse. *Exemplu:* Pentru $n = 3$ și $p = 5$, avem vectorul $X, 1, 2, 3, 4, X, 6, 7, 8, X$ unde elementele $0, 5$ și 9 sunt marcate cu X ca fiind șterse.

- Scriți un program care să șteargă toate elementele vectorului, în n pași, în aşa fel încât la pasul k să se șteargă 2^{k-1} elemente, dintre cele neșterse până la pasul respectiv. Programul va afișa $m - 1$ perechi de forma (k, q) unde q este unul dintre elementele vectorului, diferit de p , iar k este pasul la care a fost șters q . Programul scris trebuie să aibă complexitatea timp liniară în funcție de m , adică numărul de instrucțiuni ale programului să fie aproximativ egal cu dimensiunea vectorului.
- Scriți un program similar cu cel de la punctul (a), dar cu următoarea condiție suplimentară: după pasul k , între oricare două elemente deja șterse consecutive să nu fie o distanță mai mare de 2^{n-k} , unde prin distanță dintre i și j se înțelege $|j - i|$. Calculați complexitatea timp în funcție de n a programului pe care l-ați scris. *Exemplu:* Considerăm vectorul $X, 1, 2, 3, 4, X, 6, 7, 8, X$. Printr-o posibilă strategie de ștergere, conținutul vectorului după fiecare pas k este: $X, 1, X, 3, 4, X, 6, 7, 8, X$ (după pasul 1), $X, 1, X, 3, X, X, 6, X, 8, X$ (după pasul 2), respectiv $X, X, X, X, X, X, X, X, X$ (după pasul 3). Rezultatul afișat de program în acest caz este secvența $(1,2),(2,4),(2,7),(3,1),(3,3),(3,6),(3,8)$.

Notă: Programele vor fi scrise într-unul dintre limbajele de programare studiate în liceu (Pascal,C,C++). Pentru fiecare soluție se vor descrie informal detaliile algoritmului folosit și ale implementării sub formă de program: semnificația variabilelor, a structurilor de date, a structurilor repetitive, a instrucțiunilor condiționale.

Barem subiect INFO 2013

IV. Informatică.	Oficiu	1 punct.
(a)	• Programul nu șterge un element deja șters	2 puncte.
	• Programul șterge toate elementele din vector	2 puncte.
	• Programul este liniar în m	1 punct.
(b)	• Programul respectă cerința suplimentară	1 punct.
	• Calculul complexității programului	0,5 puncte.
-	Programele nu au greșeli de limbaj	1,5 puncte.
-	Claritatea rezolvărilor	1 punct.

Rezolvare subiect INFO 2013 – punctul a

Șterge elementele consecutive din sir
pornind de la prima componentă.

Verifica la fiecare pas dacă nu e deja șters.

Rezolvare subiect INFO 2013 – punctul a

```
1 #include <stdio.h>
2 #include <math.h>
3 int main()
4 {
5     int n,m,p,i,k,poz;
6     scanf("%d", &n);
7     scanf("%d", &p);
8     m = (int) pow(2,n);
9     int v[m+2];
10    for(i=0;i<m+2;i++)
11        v[i] = i;
12    v[0] = -1;v[m+1]=-1;v[p]=-1;
13    poz = 1;
14    for(k=1;k<=n;k++)
15    {
16        int elementeDeSters = (int) pow(2,k-1);
17        for(i=1;i<=elementeDeSters;i++)
18        {
19            if(v[poz] != -1)
20            { v[poz] = -1; printf("(%d,%d)",k,poz+1); poz++; }
21            else
22            { poz++;v[poz] = -1; printf("(%d,%d)",k,poz+1); poz++; }
23        }
24        //afiseaza v
25        printf("\n");
26        for(i=0;i<m+2;i++)
27            printf("%d ",v[i]);
28        printf("\n");
29    }
30    return 0;
31 }
```

```
3
5
(1,2)
-1 -1 2 3 4 -1 6 7 8 -1
(2,3)(2,4)
-1 -1 -1 -1 4 -1 6 7 8 -1
(3,5)(3,7)(3,8)(3,9)
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
```

```
Process returned 0 (0x0)   execution time : 3.216 s
Press ENTER to continue.
```

Rezolvare subiect INFO 2013 – punctul b

Șterge elementele din sir care sunt mijloacele intervalelor (deplasează +1,-1 în funcție de poziția lui p) la fiecare pas, astfel încât să fie îndeplinită condiția.

Găsește mijloacele intervalelor la fiecare pas folosind două cozi, ce rețin capetele intervalelor (stânga și dreapta);

Rezolvare subiect INFO 2013 – punctul b

```
16 //punctul b
17 int stanga[m], dreapta[m];
18 stanga[0] = 1;
19 dreapta[0] = m-1;
20 int startQ=0;
21 int finalQ=0;
22 for(k=1;k<=n;k++)
23 {
24     int elementeDeSters = (int) pow(2,k-1);
25     for(i=1;i<=elementeDeSters;i++)
26     {
27         //extrage din coada
28         int mijloc = (dreapta[startQ] + stanga[startQ])/2;
29         printf("(%d,%d)",k,mijloc+(mijloc>=p));
30         v[mijloc+(mijloc>=p)] = -1;
31         if (k<n)
32         {
33             //insereaza in coada
34             finalQ++;
35             stanga[finalQ] = stanga[startQ];
36             dreapta[finalQ] = mijloc-1;
37             finalQ++;
38             stanga[finalQ] = mijloc+1;
39             dreapta[finalQ] = dreapta[startQ];
40         }
41         startQ++;
42     }
43     printf("\n");
44     //afiseaza v
45     for(i=0;i<m+2;i++)
46         printf("%d ",v[i]);
47     printf("\n");
4
3
(1,9)
-1 1 2 -1 4 5 6 7 8 -1 10 11 12 13 14 15 16 -1
(2,5)(2,13)
-1 1 2 -1 4 -1 6 7 8 -1 10 11 12 -1 14 15 16 -1
(3,2)(3,7)(3,11)(3,15)
-1 1 -1 -1 4 -1 6 -1 8 -1 10 -1 12 -1 14 -1 16 -1
(4,1)(4,4)(4,6)(4,8)(4,10)(4,12)(4,14)(4,16)
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
```

Process returned 0 (0x0) execution time : 3.914 s
Press ENTER to continue.

Subject MATE 2013

Enunț subiect MATE 2013

IV. Informatică. Se consideră o secvență de numere naturale x_1, x_2, \dots, x_n . Din această secvență se pot obține alte secvențe folosind următoarea operație: se extrage elementul de pe poziția i ($i > 1$), se mută toate elementele situate la stânga poziției i cu o poziție la dreapta, iar elementul de pe poziția i se plasează pe prima poziție a secvenței.

- Să se realizeze un program care primind o secvență de numere naturale x_1, x_2, \dots, x_n afișează toate secvențele care se pot obține din aceasta folosind o singură dată operația definită mai sus. Ordinea în care sunt afișate secvențele rezultate nu contează. De exemplu din secvența 1,2,3 folosind o singură operație, mutând elementul de pe poziția 2 se pot obține secvența 2,1,3 și mutând de pe poziția 3 se obține secvența 3,1,2.
- Să se realizeze un program care primind două permutări x_1, x_2, \dots, x_n și y_1, y_2, \dots, y_n ale mulțimii $\{1, \dots, n\}$ afișează o secvență de operații de tipul de mai sus prin care permutarea x_1, x_2, \dots, x_n se poate transforma în permutarea y_1, y_2, \dots, y_n . O operație va fi afișată prin acel element x_i care se mută pe prima poziție. De exemplu dacă se primesc permutările: 4,5,6,7,8,9,3,1,2 și 4,9,6,5,7,8,3,1,2 o posibilă ieșire a programului este: 6,9,4 adică din prima permutare se extrage 6 și se pune în față, apoi se extrage 9 și se pune în față, iar apoi se extrage 4 și se pune în față.

Notă: Programele vor fi scrise într-unul dintre limbajele de programare studiate în liceu (Pascal,C,C++). Pentru fiecare soluție se vor descrie informal detaliile algoritmului folosit și ale implementării sub formă de program: semnificația variabilelor, a structurilor de date, a structurilor repetitive, a instrucțiunilor condiționale.

Barem subiect MATE 2013

- IV. Informatică. Oficiu 1 punct.
- (a) • Programul generează o secvență corectă 1 punct.
 • Programul generează numai secvențe corecte 2 puncte.
 • Programul generează toate secvențele corecte 2 puncte.
- (b) • Programul găsește o secvență corectă de operații 1,5 puncte.
– Programele nu au greșeli de limbaj 1,5 puncte.
– Claritatea rezolvărilor 1 punct.

Rezolvare subiect MATE 2013 – punctul a

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main()
4 {
5     int n,i;
6     printf("n=");scanf("%d",&n);
7     int x[n];
8     for(i=0;i<n;i++)
9     {
10         printf("x[%d]=",i);scanf("%d",&x[i]);
11     }
12 //punctul a
13 int aux,z[n],j,k;
14 //z e copia lui x
15 for(i=0;i<n;i++)
16     z[i] = x[i];
17 for(i=1;i<n;i++)
18 {
19     //pune elementul de pe pozitia i pe pozitia 0
20     aux = x[i];
21     for(j=i;j>0;j--)
22         x[j] = x[j-1];
23     x[0] = aux;
24     //printeaza permutarea
25     for(k=0;k<n;k++)
26         printf("%d ",x[k]);
27     printf("\n");
28     //revino la x initial
29     for(k=0;k<n;k++)
30         x[k]=z[k];
31 }
32 return 0;
```

```
n=9
x[0]=4
x[1]=5
x[2]=6
x[3]=7
x[4]=8
x[5]=9
x[6]=3
x[7]=1
x[8]=2
5 4 6 7 8 9 3 1 2
6 4 5 7 8 9 3 1 2
7 4 5 6 8 9 3 1 2
8 4 5 6 7 9 3 1 2
9 4 5 6 7 8 3 1 2
3 4 5 6 7 8 9 1 2
1 4 5 6 7 8 9 3 2
2 4 5 6 7 8 9 3 1
```

```
Process returned 0 (0x0)    execution time ...
Press ENTER to continue.
```

Rezolvare subiect MATE 2013 – punctul b

```
32 //punctul b
33 int y[n],lungimeSecventa,poz;
34 for(i=0;i<n;i++)
35 {printf("y[%d]=",i);scanf("%d",&y[i]);}
36 //incearca secventa de mutari y[0],y[1],y[2],...y[i]
37 for(lungimeSecventa=0;lungimeSecventa<n;lungimeSecventa++)
38 {
39     for(k=lungimeSecventa;k>=0;k--)
40     {
41         poz = 0;
42         while(x[poz]!=y[k])
43             poz++;
44         //adu-l pe x[poz] in fata
45         aux = x[poz];
46         for(j=poz;j>0;j--)
47             x[j] = x[j-1];
48         x[0] = aux;
49     }
50     int secventeEgale=1;
51     for(i=0;i<n;i++)
52     if(x[i]!=y[i])
53     {secventeEgale = 0;break;}
54     if(secventeEgale==0)
55     {//revino la x initial
56     for(k=0;k<n;k++)
57         x[k]=z[k];
58         continue;
59     }
60     printf("secventa minima de operatii este:\n");
61     for(k=0;k<=lungimeSecventa;k++)
62         printf("%d ",y[k]);
63         printf("\n");
```

Secvența optimă de mutări este dată de y (y[0], y[1], y[2], ...) : încercăm succesiv toate cele n posibilități pornind de la cea cu numărul minim de mutări (y[0]).

y[0]=4
y[1]=9
y[2]=6
y[3]=5
y[4]=7
y[5]=8
y[6]=3
y[7]=1
y[8]=2
secventa minima de operatii este:
4 9 6

Subject INFO 2012

Enunț subiect INFO 2012

IV. Informatică

Se consideră sirul de numere naturale $x = 1, 2, 2, 3, 3, 3, 4, 4, 4, 4, \dots$ (fiecare număr natural nenul apare, în ordine, de un număr de ori egal cu el însuși). a) Se dă un număr natural nenul n . Să se scrie un program care afișează primii n termeni al sirului x . b) Se dă un număr natural nenul n . Să se scrie un program care afișează în timp constant (care nu depinde de n) al n -lea termen al sirului x . c) Se dă un număr natural nenul n și n numere naturale nenule y_1, \dots, y_n . Să se scrie un program care verifică (afișând "DA", respectiv "NU") dacă există o permutare a termenilor y_1, \dots, y_n care să fie identică cu primii n termeni ai sirului x . d) Dați o soluție în timp liniar (în funcție de n) cerinței de la punctul c).

Notă: Programele vor fi scrise într-unul dintre limbajele de programare studiate în liceu (Pascal,C,C++). Pentru fiecare soluție se vor descrie informal detaliile algoritmului folosit și ale implementării sub formă de program: semnificația variabilelor, a structurilor de date, a structurilor repetitive, a instrucțiunilor condiționale.

Barem subiect INFO 2012

- **Informatică - IV**

Oficiu	1p
a)	4p
b)	1p
c)	3p
d)	1p

Nota: Se pot scadea maxim 2 puncte pentru greseli de limbaj sau de implementare sub formă de program.

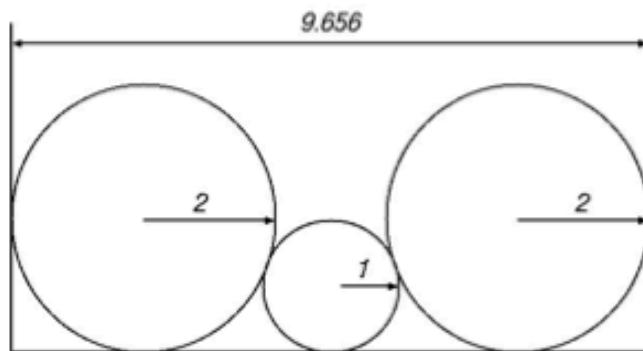
Subject MATE 2012

Enunț subiect MATE 2012

IV. Informatică

Se dau n cercuri de raze r_1, r_2, \dots, r_n . Aceste cercuri sunt “împachetate” într-un dreptunghi astfel: toate cercurile sunt tangente la baza dreptunghiului, cercurile sunt aranjate în ordinea inițială (cel mai din stânga fiind cercul de rază r_1 , cel mai din dreapta cercul de rază r_n), iar cercurile consecutive (de raze r_i și respectiv r_{i+1}) sunt tangente. Să se scrie un program care calculează lățimea minimă a dreptunghiului în care încap cercurile. Rezultatul se va afișa cu trei zecimale exacte.

Spre exemplu, dacă $n = 3$, $r_1 = 2$, $r_2 = 1$ și $r_3 = 2$, atunci rezultatul care trebuie afișat este 9.656.



Notă: Programul va fi scris într-unul dintre limbajele de programare studiate în liceu (Pascal,C,C++). Se vor descrie informal detaliile algoritmului folosit și ale implementării sub formă de program: semnificația variabilelor, a structurilor de date, a structurilor repetitive, a instrucțiunilor condiționale.

Barem subiect MATE 2012

• Informatică - IV

Oficiu	1p
Corectitudine algoritm	5p
Sintaxa limbajului de programare	2p
Detalii de algoritm și de implementare	2p

Nota: Se pot scadea maxim 2 puncte pentru greseli de limbaj sau de implementare sub formă de program.

Subject INFO 2011

Enunț subiect INFO 2011

IV. Informatică

Se dă un vector v de n elemente egale cu 1. Prin partiție a vectorului v înțelegem o împărțire a vectorului în subvectori, astfel încât fiecare element al vectorului v apare exact o dată într-unul dintre subvectori. Pentru fiecare partiție a vectorului v în k subvectori $v_{11}, \dots, v_{1n_1}, v_{21}, \dots, v_{2n_2}, \dots, v_{k1}, \dots, v_{kn_k}$, se calculează produsul sumelor elementelor din fiecare subvector al partiției, adică $\prod_{i=1}^k n_i$.

- Să se scrie un program care determină cel mai mare produs calculat în acest fel pentru toate partițiile posibile ale vectorului v .
- Există o soluție la punctul a) care să nu calculeze toate produsele posibile? Justificați.

Notă. Programele vor fi scrise într-unul dintre limbajele de programare studiate în liceu (Pascal,C,C++). Pentru fiecare soluție se vor descrie informal detaliile algoritmului folosit și ale implementării sub formă de program: semnificația variabilelor, a structurilor de date, a structurilor repetitive, a instrucțiunilor condiționale.

Barem subiect INFO 2011

IV. Informatică 1 p. din oficiu.

- a) Corectitudine algoritm (inclusiv descriere) 5 p.
- b) Soluție directă 1 p.
- c) Sintaxa limbajului de programare (inclusiv detalii de implementare) 3 p.

Subject MATE 2011

Enunț subiect MATE 2011

IV. Informatică Fie $n, m, p \leq 100$ numere naturale și fie \vec{x}_i , cu $i = 1, \dots, n$ și \vec{y}_j , cu $j = 1, \dots, m$ vectori p -dimensionali de numere reale.

a) Să se scrie un program care să calculeze elementele $d(i, j) = \|\vec{x}_i - \vec{y}_j\|^2$ (norma euclidiană la pătrat).

b) Dacă oricare 2 vectori $(\vec{x}_i, \vec{y}_j)_{i=1, \dots, n}^{j=1, \dots, m}$ sunt ortogonali, să se scrie un program care să calculeze elementele $d(i, j)$, cu efectuarea unui număr cât mai mic de operații.

Notă. Programele vor fi scrise într-unul dintre limbajele de programare studiate în liceu (Pascal, C, C++). Pentru fiecare soluție se vor descrie informal detaliile algoritmului folosit și ale implementării sub formă de program: semnificația variabilelor, a structurilor de date, a structurilor repetitive, a instrucțiunilor condiționale.

Barem subiect MATE 2011

IV. Informatică 1 p. din oficiu.

- a) Algoritm 4 p.
- b) Algoritm 1 p.
- Sintaxa limbajului de programare 2 p.
- Detalii de algoritm și de implementare 2 p.