

- Enunțați problema pe care o rezolvă subalgoritmul dat dacă este apelat cu $a = 1$ și $b = 0$.
- Ce valoare returnează apelul $f(1, 0, 10)$?
- Scrieți o variantă *recursivă* a subalgoritmului dat, respectând antetul subalgoritmului din varianta iterativă (ne-recursivă).

Subiectul III (40 puncte)

Prefix

Cifra de control a unui număr natural se determină calculând suma cifrelor numărului, apoi suma cifrelor sumei și așa mai departe până când suma obținută reprezintă un număr cu o singură cifră. De exemplu, *cifra de control* a numărului 182 este 2 ($1 + 8 + 2 = 11$, $1 + 1 = 2$).

Un număr p format din exact k cifre este *prefix* al unui număr q cu cel puțin k cifre dacă numărul format din primele k cifre ale numărului q (parcurs de la stânga la dreapta) este egal cu p . De exemplu, 17 este prefix al lui 174, iar 1713 este prefix al lui 1713242.

Se consideră un număr nr natural ($0 < nr \leq 30\,000$) și o matrice (un tablou bidimensional) A cu m linii și n coloane ($0 < m \leq 100$, $0 < n \leq 100$), având ca elemente numere naturale mai mici decât 30 000. Scrieți un program care determină și afișează *cel mai lung prefix* al numărului nr care se poate construi folosind cifrele de control corespunzătoare elementelor din matricea dată. O astfel de cifră de control poate fi folosită de oricâte ori în construirea prefixului. Dacă nu se poate construi un prefix, programul va afișa mesajul „*nu există prefix*”.

Exemplu: dacă avem $nr = 12319$, $m = 3$ și $n = 4$ și matricea $A = \begin{pmatrix} 182 & 12 & 274 & 22 \\ 22 & 1 & 98 & 56 \\ 5 & 301 & 51 & 94 \end{pmatrix}$,

cel mai lung prefix este 1231, cifrele de control fiind:

Element din matrice	182	12	274	22	1	98	56	5	301	51	94
Cifră control	2	3	4	4	1	8	2	5	4	6	4

În rezolvare folosiți subprograme pentru:

- citirea datelor de intrare de la tastatură;
- determinarea cifrei de control asociată unui număr;
- determinarea celui mai lung prefix;
- afișarea pe ecran a celui mai lung prefix sau a mesajului corespunzător dacă acesta nu a putut fi găsit.

Notă:

- Toate subiectele sunt obligatorii.
- Rezolvările trebuie scrise detaliat pe foile de examen (ciornele nu se iau în considerare).
- Se acordă 10 puncte din oficiu.
- Timpul efectiv de lucru este de 3 ore.

BAREM – VARIANTA I

OFICIU..... 10 puncte

SUBIECTUL I 35 puncte

1. Rază 20 puncte

Varianta 1: determinarea corectă a valorii *nrSchimb* bazată pe utilizarea *cmmdc(a, b)* 20 puncte

- *cmmdc(a, b)* (sau *cmmmc(a,b)*)..... 5 puncte
- calculul valorii *nrSchimb* 15 puncte

Varianta 2: determinarea corectă a valorii *nrSchimb* cu alt algoritm corect (simulare) 15 puncte

2. Numere cu „forță” 15 puncte

- stabilirea forței unui număr..... 5 puncte
- determinarea numărului grupurilor de elemente cu aceeași forță (*nrGr*) și a componenței lor (*grupuri*) 10 puncte

SUBIECTUL II..... 15 puncte

Numărul numerelor *Fibonacci* mai mici decât numărul *nr* dat

Cerința a.

- enunț problemă 5 puncte

Cerința b.

- rezultat calculat corect (7)..... 3 puncte

Cerința c.

- algoritm 3 puncte
- autoapel corect 2 puncte
- condiție de oprire din recursivitate 2 puncte

SUBIECTUL III 40 puncte

Prefix

Subprograme:

- citirea datelor de intrare 3 puncte
- determinarea cifrei de control asociată unui număr..... 5 puncte
- determinarea celui mai lung prefix 15 puncte
- afișarea celui mai lung prefix/mesaj 3 puncte

Program principal: 3 puncte

- comunicare prin parametri:
(signatura subalgorimilor și apelul corect)..... 5 puncte
- lizibilitate:
 - comentarii 2 puncte
 - indentare 2 puncte
 - denumiri sugestive 2 puncte

Concurs Mate-Info - 1 aprilie 2017

Proba scrisă la Informatică

VARIANTA I

// SUBIECTUL I.1

//determina cmmdc a 2 numere a si b

```
int cmmdc(int a, int b){
    if ((a == b) && (a != 0)){
        return a;
    }
    if (a * b == 0){
        return a + b;
    }
    while (a != b){
        if (a > b)
            a -= b;
        else
            b -= a;
    }
    return a;
}
```

// calcularea numărului de schimbări de direcție a razei

```
int raza(int a, int b){
    int nrSchimb;
    int d = cmmdc(a, b);
    nrSchimb = b / d + a / d - 2;
    return nrSchimb;
}
```

// SUBIECTUL I.2

const int MAXSIZE = 100;

const int MAXCIF = 32;

// calcularea fortei unui numar

```
int calculForta(int nr){
    int forta = 0;
    do{
        nr &= nr - 1; // „și” logic între nr și nr-1
        forta++;
    } while (nr);
    return forta;
}
```

//gruparea numerelor pe clase de forta

//grupurile se vor memora într-un tablou bidimensional

//fiecare linie i corespunde grupului de forta i

//primul element de pe linia i reprezintă dimensiunea grupului de forta i

//urmatoarele elemente sunt numerele din sirul x cu forta i

```
void forte(int n, int x[], int &nrGr, int grupuri[][MAXSIZE]){
    for (int i = 1; i < MAXCIF; i++) // inițializarea tabloului grupuri
        grupuri[i][0] = 0;
    nrGr = 0;
    for (int i = 0; i < n; i++){
        int forta = calculForta(x[i]); // determinarea forței elementului x[i]
        if (grupuri[forta][0] == 0)
            nrGr++;
        int pos = grupuri[forta][0] + 1;
        grupuri[forta][pos] = x[i];
        grupuri[forta][0]++;
    }
}
```

```

// SUBIECTUL II
int fRec(int a, int b, int nr){
    if (a == 0)
        return 0;
    if (b < nr)
        return fRec(b, a + b, nr) + 1; // (*)
    else
        return 0;
}

// SUBIECTUL III
const int MAX = 100;
const int NRMAXCIFRE = 10;

//citirea unui numar
void citireNumar(int &nr){
    cin >> nr;
}

//citirea elementelor unei matrici
void citireMatrice(int &m, int &n, int A[][MAX]){
    cin >> m >> n;
    for (int i = 0; i < m; i++){
        for (int j = 0; j < n; j++)
            cin >> A[i][j];
    }
}

//citirea numarului si a elementelor din matrice
void citireDate(int &nr, int &m, int &n, int A[][MAX]){
    citireNumar(nr);
    citireMatrice(m, n, A);
}

//calcularea cifrei de control a unui numar
int cifraControl(int x){
    while (x > 9){
        int y = x;
        int s = 0;
        while (y > 0){
            s += y % 10;
            y /= 10;
        }
        x = s;
    }
    return x;
}

```

```

//determinarea prefixului maxim
//matricea se parcurgere o singură dată și se construiește un vector de apariții
//pentru cifrele de control corespunzătoare elementelor din matrice.
//Se reține într - un vector cifrele numărului dat(nr).
//Se parcurge vectorul acestor cifre(începând cu cifra cea mai semnificativă) și
//se verifică apariția cifrelor în vectorul de apariții construit anterior
int prefixMaxCifre(int nr, int m, int n, int A[][MAX], int cifre[], int &nrCifre){
    bool aparitii[NRMAXCIFRE];
    int i = 0;
    for (i = 0; i < 10; i++) // initializarea vectorului de frecvente
        aparitii[i] = 0;
    for (i = 0; i < m; i++){
        for (int j = 0; j < n; j++){
            aparitii[cifraControl(A[i][j])] = 1; // aparitia cifrei de control
            // corespunzatoare elementului din matricea A
        }
    }
    nrCifre = 0; // numarul cifrelor numarului nr dat
    while (nr > 0){
        cifre[nrCifre++] = nr % 10; // elementele sirului cifrelor numarului nr dat
        nr /= 10;
    }

    i = nrCifre - 1; // parcurgem sirul cifrelor
    while ((i >= 0) && (aparitii[cifre[i]])) // cifra de control = cu cifra curenta din nr
        i--;
    return nrCifre - i - 1;
}

//se afiseaza prefixul maxim de lungime lung cu cifre din vectorul de cifre
void afisarePrefix(int lung, int cifre[], int nrCifre){
    if (lung == 0)
        cout << "nu exista prefix";
    for (int i = 0; (i < lung && i < nrCifre); i++)
        cout << cifre[nrCifre - i - 1];
    cout << endl;
}

int main(){

    int nr = -1;
    int m = -1;
    int n = -1;
    int A[MAX][MAX];
    citireDate(nr, m, n, A);

    int cifre[NRMAXCIFRE];
    int nrCifre = 0;
    int lung = prefixMaxCifre(nr, m, n, A, cifre, nrCifre);
    afisarePrefix(lung, cifre, nrCifre);

    return 0;
}

```