

PROBLEME REZOLVATE

1.

Scrieți definiția completă a subprogramului `numar`, cu trei parametri, care primește prin intermediul parametrului `n` un număr natural format din cel mult 9 cifre, iar prin intermediul parametrilor `c1` și `c2` câte o cifră nenulă. Subprogramul caută prima apariție (de la stânga spre dreapta) a cifrei `c1` în `n`, și dacă aceasta apare, o înlocuiește cu `c2`, iar următoarele cifre, dacă există, sunt înlocuite cu câte o cifră 0. Subprogramul furnizează tot prin `n` numărul astfel obținut. Dacă cifra `c1` nu apare în `n`, atunci valoarea lui `n` rămâne nemodificată.

Exemplu: pentru `n=162448`, `c1=4` și `c2=7` valoarea furnizată prin `n` va fi `162700`.

```
void numar ( int &n, int c1, int c2 ) {
    int p=1, q, a=n;
    while (n>0) {
        if (n%10==c1) q=p;
        p=p*10;
        n=n/10;
    }
    n=(a/q-c1+c2)*q;
}
```

2.

Subprogramul `dublu` primește prin intermediul parametrului `k` un număr natural cu maxim 9 cifre și prin intermediul parametrului `x` o cifră zecimală. Subprogramul furnizează prin intermediul parametrului `a` numărul natural obținut prin înlocuirea fiecărei cifre din `k` ce este egală cu `x` prin anterioara cifră.

Exemplu : dacă `k=123242` și `x=2` atunci `a` va fi `113344`

Scrieți definiția completă a subprogramului `dublu`.

```
void dublu ( int k, int x, int &a ) {
    int p=1, u;
    a=0;
    while (k>9) {
        u=k%10;
        if (u==x) u=k/10%10;
        a=a+u*p;
        p=p*10;
        k=k/10;
    }
    a=a+k*p;
}
```

3.

Funcția `verif` primește prin intermediul parametrului `n` un număr natural format din cel mult 9 cifre, și prin intermediul parametrului `a`, un număr natural nenul ($2 \leq a \leq 9$). Funcția returnează valoarea 1 dacă `n` este un număr format din cifre aparținând intervalului închis $[0, a]$ și valoarea 0 în caz contrar.

a) Scrieți definiția completă a funcției `verif`.

b) Spunem că `n` poate fi o reprezentare în baza `b` ($1 < b \leq 10$), dacă toate cifrele lui `n` sunt strict mai mici decât `b`. Scrieți un program care citește de la tastatură o valoare naturală `n` cu cel mult 9 cifre și, utilizând apeluri ale funcției `verif`, afișează pe ecran, în ordine crescătoare, cu spații între ele, toate valorile lui `b` pentru care valoarea citită **nu** poate fi o reprezentare în baza `b`.

Exemplu: Pentru `n=4101`, se afișează 2 3 4.

```
#include <iostream>
using namespace std;

int verif ( int n, int a ) {
    while (n>0) {
        if (n%10>a) return 0;
        n=n/10;
    }
    return 1;
}

int main()
{
    int n,b;
    cin>>n;
    b=2;
    while (b<=9)
        if (verif(n,b-1)==0) break;
        else b++;
    while (b<=9) {
        cout<<b<<' ';
        b++;
    }
    return 0;
}
```

3.

Subprogramul `trei` primește prin intermediul parametrilor `a` și `b` două numere naturale nenule formate doar din cifre nenule. Subprogramul înlocuiește în `a` toate cifrele egale cu cifra maximă din `b` prin cifra minimă din `b` iar toate cifrele din `b` egale cu cifra minimă din `a` le înlocuiește cu cifra maximă din `a`. Subprogramul furnizează cele două valori tot prin parametrii `a` și `b`.

Exemplu: dacă `a=51335143` și `b=1030331` subprogramul furnizează valorile de mai jos `a=51005140` `b=5030335`

Scrieți definiția completă a subprogramului `trei`.

```

void limite(int n, int &Max, int &Min ) {
    int u;
    Max=n%10; Min=n%10;
    while (n>0) {
        u=n%10;
        if (u>Max) Max=u;
        if (u<Min) Min=u;
        n=n/10;
    }
}

void inlocuire( int &n, int c1, int c2 ) {
    int u, x=0, p=1;
    while (n>0) {
        u=n%10;
        if (u==c1) u=c2;
        x=x+u*p;
        p=p*10;
        n=n/10;
    }
    n=x;
}

void trei ( int &a, int &b ) {
    int Ma, ma, Mb, mb;
    limite(a, Ma, ma);
    limite(b, Mb, mb);
    inlocuire(a, Mb, mb);
    inlocuire(b, ma, Ma);
}

```

4.

Subprogramul `bac` primește prin parametrul `x` un număr natural și returnează valoarea 1 dacă are cifrele ordonate strict crescător sau strict descrescător, în toate celelalte cazuri returnând valoarea 0.

Scrieți definiția completă a subprogramului.

```

int bac (int x ) {
    int a=x;
    while (x>9 && x%10>x/10%10)
        x=x/10;
    while (a>9 && a%10<a/10%10)
        a=a/10;
    if (x<=10 || a<=10) return 1;
    else return 0;
}

```

5.

Subprogramul **apropiat** primește prin parametrul **x** un număr natural și returnează numărul palindrom cel mai apropiat de **a**. În cazul în care sunt mai multe soluții, se va considera cea mai mică.

Scrieți definiția completă a subprogramului.

```
int oglindit ( int x ) {
    int og=0;
    while (x>0) {
        og=og*10+x%10;
        x=x/10;
    }
    return og;
}

int apropiat( int x ) {
    int a=x, b=x+1;
    while ( oglindit(a)!=a ) a--;
    while ( oglindit(b)!=b ) b++;
    if (x-a<b-x) return a;
    else return b;
}
```

6.

Subprogramul **replace** primește prin parametrii **a,b** două numere naturale și înlocuiește în **a** toate cifrele aflate pe poziții impare cu prima cifră din **b** iar în **b**, toate cifrele aflate pe pozițiile pare cu prima cifră din **a**. Pozițiile cifrelor se numerotează de la 1, începând cu cifra unităților.

Exemplu: pentru $a=12347$ și $b=567894$ vor rezulta $a=52545$ $b=161814$

Scrieți definiția completă a subprogramului.

```
int prima ( int x ) {
    while (x>9)
        x=x/10;
    return x;
}

void inlocuire(int &x, int c, int r) {
    int u, z=0, p=1, k=0;
    do {
        k++;
        u=x%10;
        if (k%2==r) u=c;
        z=z+u*p; p=p*10;
        x=x/10;
    }while (x>0);
    x=z;
}

void Replace (int &a, int &b) {
    int c1=prima(a), c2=prima(b);
    inlocuire(a, c2, 1);
    inlocuire(b, c1, 0);
}
```

7.

Subprogramul `insert` primește prin parametrii `x,y` două numere naturale și inserează toate cifrele lui `y` în interiorul lui `x`, din două în două poziții, conform exemplului următor.

Exemplu: pentru `x=123456` și `y=897` va rezulta `x=123849576` iar pentru `x=123` și `y=89765` va rezulta `x=89716253`.

Scrieți definiția completă a subprogramului.

```
void Insert(int &x, int y) {
    int a=0, p=1;
    while (x>0 && y>0) {
        a=a+x%10*p; p=p*10;
        a=a+y%10*p; p=p*10;
        x=x/10; y=y/10;
    }
    while (x>0) {
        a=a+x%10*p; p=p*10;
        x=x/10;
    }
    while (y>0) {
        a=a+y%10*p; p=p*10;
        y=y/10;
    }
    x=a;
}
```

8.

Subprogramul `verif` primește prin singurul său parametru, `x`, un număr natural nenul cu cel mult 9 cifre și returnează valoarea 1 dacă numărul conține cel puțin o secvență de 3 cifre impare alăturate și 0 în caz contrar.

Exemplu: dacă `x=7325972` se va returna valoarea 1.

Scrieți definiția completă a subprogramului `verif`.

```
int verific ( int x ) {
    int k=0, u;
    while (x>0) {
        u=x%10;
        if (u%2==1) k++;
        else k=0;
        if (k==3) return 1;
        x=x/10;
    }
    return 0;
}
```