

Se citește de la tastatură un vector cu n elemente numere naturale ($n \leq 100$).

Să se înlocuiască fiecare element al vectorului cu produsul cifrelor impare sau cu -1 în cazul în care elementul vectorului conține doar cifre pare.. Indexarea elementelor începe de la 1.

Să se afișeze vectorul după actualizarea lui.

Se vor scrie și folosi funcții recursive pentru:

- citirea vectorului
- afișarea vectorului
- calculul produsului cifrelor impare
- înlocuirea cerută

Recursivitate

Exemplu:

Intrare

7 223 435 6667 24 55 662 122

Ieșire

3 15 7 -1 25 -1 1

Rezolvare test

Exemplu:

Intrare

7 223 435 6667 24 55 662 122

Iesire

3 157 -1 25 -1 1

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

void citire(int A[], int n)
{
    if(n>0)
    {
        citire(A,n-1);
        cin>>A[n];
    }
}
```

```
void afisare(int A[], int n)
{
    if(n>0)
    {
        afisare(A,n-1);
        cout<<A[n]<<" ";
    }
}

int produs(int n, int &p)
{
    if(n==0) return 1;
    else
        if(n%2==1)
        {
            p=1;
            return produs(n/10,p) * (n%10);
        }
    else
        return produs(n/10,p);
}
```

```
void inlocuire(int A[], int n)
{
    if(n>0)
    {
        inlocuire(A,n-1);
        int ok=-1;
        int x=produs(A[n],ok);
        if(ok!=-1)
            A[n]=x;
        else
            A[n]=ok;
    }
}
```

```
int main()
{
    int A[101],n;
    cin>>n;
    citire(A,n);
    inlocuire(A,n);
    afisare(A,n);
    return 0;
}
```

Se citește de la tastatură un vector cu n elemente numere naturale ($n \leq 100$).

Să se înlocuiască fiecare element al vectorului cu suma cifrelor cu aceeași paritate ca și indicele elementului. Indexarea elementelor începe de la 1. Să se afișeze vectorul după actualizarea lui.

Se vor scrie și folosi funcții recursive pentru:

- citirea vectorului
- afișarea vectorului
- calculul sumei cifrelor de o anumită paritate
- înlocuirea cerută

Rezolvare test

Recursivitate

Exemplu:

Pentru datele de mai jos

7

223 435 6667 24 55 662 122

Șirul rezultat este

3 4 7 6 10 14 1

Exemplu:

Pentru datele de mai jos

7

223 435 6667 24 55 662 122

Șirul rezultat este

3 4 7 6 10 14 1

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

void citire(int A[], int n)
{
    if(n>0)
    {
        citire(A,n-1);
        cin>>A[n];
    }
}
```

```
void afisare(int A[], int n)
{
    if(n>0)
    {
        afisare(A,n-1);
        cout<<A[n]<<" ";
    }
}
```

```
int sumcifp(int n, int p)
{
    if(n==0) return 0;
    else if(n%2==p)
        return sumcifp(n/10,p)+n%10;
    else
        return sumcifp(n/10,p);
}
```

```
void inlocuire(int A[], int n)
{
    if(n>0)
    {
        inlocuire(A,n-1);
        A[n]=sumcifp(A[n],n%2);
    }
}
```

```
int main()
{
    int A[101],n;
    cin>>n;
    citire(A,n);
    inlocuire(A,n);
    afisare(A,n);
    return 0;
}
```

Se citeste un numar natural n ($n \leq 20$).

Afisati un desen format din caracterul * ca in exemplul de mai jos.

Se vor folosi exclusiv subprograme recursive.

Exemplu:

Pentru $n=3$ se afiseaza

```
*  
**  
***  
**  
*
```

```
#include <iostream>  
using namespace std;  
void stelute(int n)  
{  
    if(n>0)  
    {  
        cout<<"*";  
        stelute(n-1);  
    }  
}
```

```
void sus(int n)  
{  
    if(n>0)  
    {  
        sus(n-1);  
        stelute(n);  
        cout<<endl;  
    }  
}  
void jos(int n)  
{  
    if(n>0)  
    {  
        stelute(n);  
        cout<<endl;  
        jos(n-1);  
    }  
}  
int main()  
{  
    int n;  
    cin>>n;  
    sus(n);  
    jos(n-1);  
    return 0;  
}
```

Se citeste un numar natural n ($n \leq 20$).

Afisati un desen format din caracterul * ca in exemplul de mai jos.

Se vor folosi exclusiv subprograme recursive.

Exemplu:

Pentru $n=3$ se afiseaza

```
*
***
*****
***
*
```

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

void character(int n, char c)
{
    if(n>0)
    {
        character(n-1,c);
        cout<<c;
    }
}
```

```
void sus(int n, int r)
{
    if(r<=n)
    {
        character(n-r, ' ');
        character (2*r-1, '*');
        cout<<endl;
        sus(n,r+1);
    }
}

void jos(int n, int r)
{
    if(r<=n)
    {
        jos(n,r+1);
        character(n-r+1, ' ');
        character(2*r-1, '*');
        cout<<endl;
    }
}

int main()
{
    int n;
    cin>>n;
    sus(n,1);
    jos(n-1,1);
    return 0;
}
```