

METODA QUICK SORT (SORTARE RAPIDĂ)

Se consideră un șir cu n elemente aleatoare. Să se ordoneze elementele șirului crescător.

Prin această metodă de sortare se execută următoarele operații:

- Primul element din șir (numit **pivot**) este mutat în cadrul vectorului pe poziția pe care trebuie să se găsească în final (acest element rămâne fix);
- Vectorul este apoi împărțit (cu ajutorul metodei **divide et impera**) în alți doi vectori (se vor obține 2 **pivoți**):
 - o primul având elementele din vectorul inițial începând de la **poziția 1** și până la poziția **pivot-1**)
 - o al doilea având elementele din vectorul inițial începând de la poziția **pivot+1** și până la **poziția n**)
- În mod analog se stabilește poziția celor doi pivoți în vectorul final
- Procesul continuă până vor fi așezate pe pozițiile finale toți pivoții (toate elementele din șir)

Algoritmul în C++ are următorul cod:

```
#include<iostream>
using namespace std;
int v[100],n,k;

void poz(int li,int ls,int &k,int v[100])
{
    int i=li,j=ls,c,i1=0,j1=-1;
    while(i<j)
    {
        if(v[i]>v[j])
        {
            c=v[j];
            v[j]=v[i];
            v[i]=c;
            i1=1; //i1 creste
            j1=0; //j1 ramane constant
        }
        i=i+1; //i va creste daca s-a facut interschimbarea pivotului
        j=j-1; //j va scadea daca nu s-a facut interschimbarea pivotului
    }
}
```

```

    k=i; //k va memora pozitia finala a pivotului
}

void quick(int li,int ls)
{
    if(li<ls)
    {
        poz(li,ls,k,v); /* va stabili pozitia finala a pivotului in sir
                           (pivotul este primul elemnt din fiecare sir)*/
        quick(li,k-1); // divide sirul de la inceput pana la pivot-1
        quick(k+1,ls); // divide sirul de la pivot+1 pana la sfarsit
    }
}

int main()
{
    int i;
    cout<<"n=";
    cin>>n;
    for(i=1;i<=n;i++)
    {
        cout<<"v["<<i<<"]=";
        cin>>v[i];
    }
    quick(1,n);
    for(i=1;i<=n;i++)
        cout<<v[i]<<" ";
    return 0;
}

```