

Tutoriat Programarea Calculatoarelor

Nicoi Alexandru - Duță Andrei

27 Noiembrie 2020

Utilizare qsort

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int cmp(const void * a, const void * b)
{
    if ( *(int*)a < *(int*)b )
        return -1;
    if ( *(int*)a == *(int*)b )
        return 0;
    if ( *(int*)a > *(int*)b )
        return 1;
}
/// comparator standard ce suporta
/// multiple tipuri de date pentru numere

int cmpchar(char a[256], char b[256])
{
    if(strcmp(a,b) == 0)
        return 0;
    else if(strcmp(a,b) < 0)
        return -1;
    else return 1;
}

/// comparator standard ce suporta
/// siruri de caractere pt comparare

/// rezultatul comparatorului
/// se interpreteaza asa:

/// -1 => a va sta inaintea lui b
/// 0 => pozitie neschimbata
/// 1 => a va sta dupa b

/// pentru a sorta descrescator
/// interschimbati returnurile 1 si -1

int main()
{
    int n;
```

```

int v[101];
printf("-----CITESTE_N_NUMERE-----\n");
scanf("%d", &n);
for(int i = 0 ; i < n ; i++)
    scanf("%d", &v[i]);
qsort(v,n,sizeof(int),cmp);
/// s - vectorul de nr
/// n - numarul de elemente
/// sizeof(int) - marimea primului element din
///          vector, care e int
/// cmp - comparatorul de numere
for(int i = 0 ; i < n ; i++)
    printf("%d_", v[i]);
printf("\n\n-----CITESTE_N_CUVINTE-----\n");
scanf("%d", &n);
char s[101][256]; /// vector de siruri de caractere
/// in esenta e o matrice
/// in sensul ca
/// fiecare linie reprezinta cate un cuvant
/// iar coloanele, literele cuvantului
/// g e n c a s a
/// a l e x
/// m e r e
/// mai multe detalii cand vom face pointeri
for(int i = 0 ; i < n ; i++)
    scanf("%s", s[i]);
printf("\n"); /// endl din C++ e \n aici
qsort(s,n,sizeof(s[0]),cmpchar);
/// s - vectorul de siruri
/// n - numarul de elemente
/// sizeof(s[0]) - marimea primului element din
///          vector
/// cmpchar - comparatorul de siruri
for(int i = 0 ; i < n ; i++)
    printf("%s\n", s[i]);
}

```