Ministerul Educaţiei, Culturii și Cercetării

al Republicii Moldova   
  
Universitatea Tehnică a Moldovei   
  
  
  
  
  
  
  
RAPORT

Lucrarea de laborator nr. 1   
la Structuri de date si algoritmi

A efectuat: st. gr. TI-211 Popa Cătălin   
  
A verificat: Stratulat Ștefan

UTM, Chișinău 2021

**Tema:**

Implementarea tipului de date abstract “Tablou de structuri” în limbajul C.

**Scopul lucrării**

Scopul lucrării este de a familiariza studentul cu mecanismul de creare a tablourilor statice și dinamice de structuri și operații elementare asupra acestuia, utilizînd pentru aceasta limbajul C.

**Problema**

1. **Să se elaboreze un program ce va aloca static un tablou unidimensional de structuri și va realiza următoarele funcții:**

* citirea de la tastatură a elementelor tabloului;
* afișarea la consolă a elementelor tabloului;
* căutarea unui element după o valoare a unui cîmp din structură și returnarea poziției unde se află primul element găsit.

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <string.h>

typedef struct Telefoane

{

char denumire[15];

char pret[10];

} Telefoane;

void citire\_array(Telefoane \* x, int num)

{

for(int i = 0; i < num; i++)

{

printf("Introduceti datele telefonului -[%d]-:\n", i);

printf("Denumirea: ");

scanf("%s",&x[i].denumire);

printf("Pretul: ");

scanf("%s",&x[i].pret);

}

}

void print\_array(Telefoane \* x, int num)

{

for(int i = 0; i < num; i++)

{

printf("\nTelefonul: %d\n",i);

printf("Denumirea: %s\n", x[i].denumire);

printf("Pretul: %s\n", x[i].pret);

}

}

void search\_by\_key(Telefoane \* x, int n)

{

int key;

char cauta[10];

printf("\nCauta dupa: \n1)Denumire \n 2)Pret\n");

scanf("%d",&key);

if(key == 1)

{

printf("Dati denumirea:");

scanf("%s",&cauta);

for(int j=0; j<n; j++)

{

if(strcmp(x[j].denumire,cauta) == 0)

{

printf("Elementul cautat se afla pe pozitia: %d\n",j);

break;

}

}

}

else if(key == 2)

{

printf("Dati pretul:");

scanf("%s",&cauta);

for(int j=0; j<n; j++)

{

if(strcmp(x[j].pret,cauta) == 0)

{

printf("Elementul cautat se afla pe pozitia: %d\n",j);

break;

}

}

}

}

int main()

{

int num;

printf("Dati numarul de telefoane: \n");

scanf("%d",&num);

Telefoane tel[num];

citire\_array(tel,num);

printf("-------------------------------------------------");

print\_array(tel,num);

printf("-------------------------------------------------");

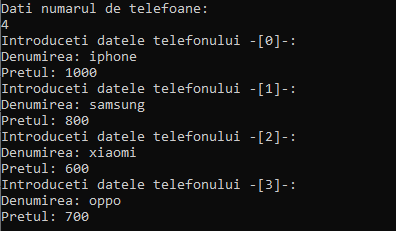
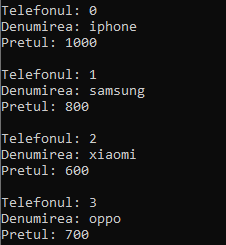
search\_by\_key(tel,num);

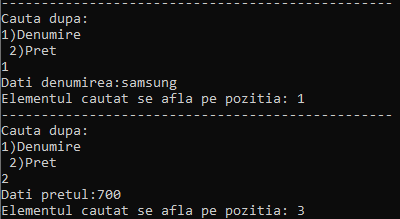
printf("-------------------------------------------------");

search\_by\_key(tel,num);

return 0;

}



1. **Să se elaboreze un program ce va aloca dinamic un tablou unidimensional de structuri și va realiza următoarele funcții, unde funcțiile date reprezintă opțiuni organizate într-un meniu în cadrul programului:**

* citirea de la tastatură a elementelor tabloului;
* afișarea la consolă a elementelor tabloului;
* căutarea unui element după o valoarea a unui cîmp din structură și returnarea poziției unde se află primul element găsit;
* eliberarea memoriei tabloului;
* sortarea elementelor după un cîmp a structurii;
* inserarea unui element nou la capătul tabloului;
* inserarea unui element nou la începutul tabloului;
* insearea unui element la o poziție X (introdusă de la tastatură) în cadrul tabloului;
* ștergerea unui element de pe o poziție X ( introdusă de la tastatură) din cadrul tabloului.

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <string.h>

typedef struct Telefoane

{

char denumire[15];

char pret[10];

} Telefoane;

Telefoane \* create\_array(int n)

{

Telefoane \* x = calloc(sizeof(Telefoane),n);

return x;

}

void citire\_array(Telefoane \* x, int num)

{

for(int i = 0; i < num; i++)

{

printf("Introduceti datele telefonului -[%d]-:\n", i);

printf("Denumirea: ");

scanf("%s",&x[i].denumire);

printf("Pretul: ");

scanf("%s",&x[i].pret);

}

}

void delete\_array(Telefoane \* x)

{

free(x);

printf("\nMemoria a fost eliberata cu succes!\n");

}

void print\_array(Telefoane \* x, int num)

{

for(int i = 0; i < num; i++)

{

printf("Telefonul: %d\n",i);

printf("Denumirea: %s\n", x[i].denumire);

printf("Pretul: %s\n", x[i].pret);

}

}

void search\_by\_key(Telefoane \* x, int n)

{

int key;

char cauta[10];

printf("Cauta dupa: 1)Denumire \n 2)Pret\n");

scanf("%d",&key);

if(key == 1)

{

printf("Dati denumirea:");

scanf("%s",&cauta);

for(int j=0; j<n; j++)

{

if(strcmp(x[j].denumire,cauta) == 0)

{

printf("Elementul cautat se afla pe pozitia: %d\n",j);

break;

}

}

}

else if(key == 2)

{

printf("Dati pretul:");

scanf("%s",&cauta);

for(int j=0; j<n; j++)

{

if(strcmp(x[j].pret,cauta) == 0)

{

printf("Elementul cautat se afla pe pozitia: %d\n",j);

break;

}

}

}

}

//sortare dupa marimea denumirii

void sortare(Telefoane \*x, int n)

{

Telefoane au;

for(int i = 0; i < n; i++)

{

for(int j = 0; j < (n - 1 - i); j++)

{

if(stricmp(x[j].denumire,x[j+1].denumire))

{

au = x[j];

x[j] = x[j +1];

x[j + 1] = au;

}

}

}

}

void add\_element(Telefoane \*x, int n)

{

for(int i=0; i<n; i++)

{

citire\_array(x,i);

}

}

void add\_start(Telefoane \*x,int n)

{

x = realloc(x,n\*sizeof(Telefoane));

for(int i=n -1;i>=0; i--)

{

x[i]=x[i - 1];

citire\_array(x,0);

}

}

void add\_end(Telefoane \*x, int n)

{

x = realloc(x,(n)\*sizeof(Telefoane));

citire\_array(x,n=1);

}

//void add\_element\_pos(Telefoane \*x, int n)

//{

// printf("\nDati pozitia: \n");

// int a;

// scanf("%d",&a);

// for(int i=n-1; i>=a; i--)

// {

// x[i] = x[i-1];

// }

// x = realloc(x, n\* sizeof(Telefoane));

// citire\_array(x,a);

//}

void add\_element\_pos(Telefoane \*x, int n)

{

printf("\nDati pozitia: \n");

int pos;

scanf("%d",&pos);

n++;

for(int i=n-1; i=pos; i--)

{

x[i] = x[i-1];

citire\_array(x,pos);

break;

}

// x[pos - 1] = n;

x = realloc(x, n\* sizeof(Telefoane));

}

void delete\_element(Telefoane \*x, int n)

{

int pozitie;

printf("Dati pozitia: \n");

scanf("%d",&pozitie);

for(int i = pozitie - 1; i < n - 1; i++)

{

x[i] = x[i + 1];

}

n--;

x = realloc(x, n\*sizeof(Telefoane));

}

int menu()

{

int error\_input\_found = 0;

menu\_start:

system("cls");

printf("---------Menu---------\n");

printf("1) Creaza tablou\n");

printf("2) Initializeaza tablou\n");

printf("3) Afiseaza tablou\n");

printf("4) Elibereaza tablou\n");

printf("5) Cauta in tablou un element\n");

printf("6) Sortarea elementelor\n");

printf("7) Adaugare element la sfarsit\n");

printf("8) Adaugare element la inceput\n");

printf("9) Adaugare element pe o poztitie dorita\n");

printf("10) Stergerea elementului de pe o pozitie\n");

printf("0) Opriti programul");

int command = 100;

printf("\n\nSelectati commanda: ");

scanf("%d",&command);

if((command>10)||(command<0))

{

error\_input\_found = 1;

goto menu\_start;

}

return command;

}

int main()

{

int command=100;

Telefoane \* x = NULL;

int n;

while(command)

{

command = menu();

switch(command)

{

case 1:

{

printf("Dati n=");

scanf("%d",&n);

x = create\_array(n);

break;

}

case 2 ... 10:

{

if(x == NULL)

{

printf("\nTabloul este NULL!\n");

break;

}

switch(command)

{

case 2:

{

citire\_array(x,n);

break;

}

case 3:

{

print\_array(x,n);

break;

}

case 4:

{

delete\_array(x);

x = NULL;

n = 0;

break;

}

case 5:

{

search\_by\_key(x,n);

break;

}

case 6:

{

sortare(x,n);

break;

}

case 7:

{

n++;

add\_end(x,n);

break;

}

case 8:

{

n++;

add\_start(x,n);

break;

}

case 9:

{

n++;

add\_element\_pos(x,n);

break;

}

case 10:

{

delete\_element(x,n);

n--;

break;

}

}

printf("\n\nApasati orice buton pentru continuare!\n");

\_getch();

}

}

}

return 0;

}

