Lucrarea 7 . HEAP

#include <iostream>

#include <fstream>

using namespace std;

ifstream fin("vector.in");

int max(int a[], int n){

int max = a[0];

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

if (a[i] > max)

max = a[i];

return max;

}

int min(int a[], int n){

int min = a[0];

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

if (a[i] < min)

min = a[i];

return min;

}

void creare\_heap(int a[],int n){

int i, j;

int aux;

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

for (j = 0;j < n/2; j++){

if (a[j] < a[2\*j+1]){

aux = a[j];

a[j] = a[2\*j+1];

a[2\*j+1] = aux;

}

if (a[j] < a[2\*j+2]){

aux = a[j];

a[j] = a[2\*j+2];

a[2\*j+2] = aux;

}

}

}

void sortare\_desc(int a[], int heap[],int &n, int &k){

int i;

int f = 0;

cout << "Max:" << a[0] << endl;

while (n){

int val = max(a,n);

int aux;

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

if (a[i] == val && i != n-1){

aux = a[0];

a[0] = a[i];

a[i] = aux;

}

aux = a[0];

a[0] = a[n-1];

a[n-1] = aux;

if (a[n-1] == val){

heap[k] = a[n-1];

n--; k++;

}

if (n == 0)

cout << "Min:" << a[n] << endl;

}

for (i = 0;i < k; i++)

cout << heap[i] << ' ';

}

void sortare\_cresc(int a[], int heap[],int &n, int &k){

int i;

int f = 0;

cout << "Max:" << a[0] << endl;

while (n){

int val = min(a,n);

int aux;

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

if (a[i] == val && i != n-1){

aux = a[0];

a[0] = a[i];

a[i] = aux;

}

aux = a[0];

a[0] = a[n-1];

a[n-1] = aux;

if (f == 0 && a[n-1] == val){

cout << "Min:" << a[n-1] << endl;

}

if (a[n-1] == val){

heap[k] = a[n-1];

n--; k++;

f++;

}

}

for (i = 0;i < k; i++)

cout << heap[i] << ' ';

cout << endl;

}

void problema2(int arr[], int m){

int ind;

cout << "Introduceti pozitia nodului carui vreti sa aflati informatie :";

cin >> ind;

if (ind >= 1){

cout << "Parintele nodului " << arr[ind] << " este nodul: " << arr[ind/2] << endl;

}

else

cout << "Nodul dat este radacina arborelui.\n";

if (ind\*2+1 < m){

cout << "Descendentul stang " << arr[ind];

cout << " este nodul: " << arr[ind\*2+1] << endl;

}

else

cout << "Nu are descendent stang\n";

if (ind\*2+2 < m){

cout << "Descendentul drept " << arr[ind];

cout << " este nodul: " << arr[ind\*2+2] << endl;

}

else

cout << "Nu are descendent drept.\n";

cout << "\n";

}

int main(){

int a[100];

int heap[100];

int k = 0;

int i, n;

int optiuni;

fin >> n;

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

fin >> a[i];

cout << "Vector citit:\n";

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

cout << a[i] << ' ';

cout << endl;

cout << "Problema 1" << endl;

cout << "Vector heap:\n";

creare\_heap(a,n);

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

cout << a[i] << ' ';

cout << endl;

cout << "Problema 2" << endl;

problema2(a,n);

cout << "Problema 3" << endl;

cout << "Alegeti cifra care va trebuie\n";

cout << "1.Sortare descrescatoare\n";

cout << "2.Sortare crescatore\n";

cin >> optiuni;

switch(optiuni){

case 1 : {

sortare\_desc(a,heap,n,k);

}; break;

case 2 : {

sortare\_cresc(a,heap,n,k);

}; break;

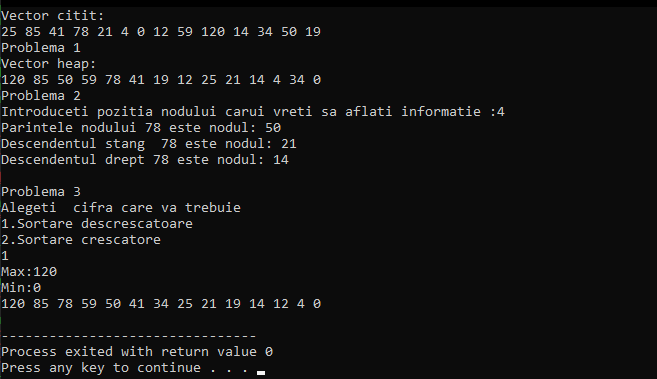
default :

cout << "A-ti ales cifra gresita";

}

cout << endl;

return 0; }



vector.in contine :

