MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE NATIONAL TECHNICAL UNIVERSITY

“KHARKIV POLYTECHNIC INSTITUTE”

DEPARTMENT OF SOFTWARE ENGINEERING AND MANAGEMENT INFORMATION TECHNOLOGIES

### PROGRAMING BASICS

### Laboratory Training 5

# Use of Arrays and Pointers

Executed by: SALEM CHAOUFI of group КН-221 iв.e

KHARKIV 2021

## 1 Training Tasks

### 1.1 Checking Palindromes

Read a sentence (array of characters) from the keyboard using the getline() function, check whether it is a palindrome and display the corresponding message. Recommendation: do not use capital letters when entering.

A palindrome is a sentence that reads the same in both directions (from left to right and from right to left), for example, "never odd or even" Spaces and punctuation are not taken into account. In the software implementation, it is advisable to remove them and check the resulting string.

#include <iostream>

#include <cstring>

#include <string>

using namespace std;

int main()

{

string word;

cout << "input sentence (do not use capital letter) : ";

std::getline (std::cin,word);

int n = word.length();

string new\_word = "";

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

new\_word += word.at(i);

}

if(word == new\_word){

cout << "char is palindrome" << endl ;

}

else{

cout << "char is not palindrome" << endl ;

}

}



### 1.2 3D-Points

Create a structure to describe a 3D-point. Write a program that calculates distance between two 3D-points. Read the coordinates of points using the scanf() function. Use printf() function for output.

#include <iostream>

#include <stdio.h>

#include <math.h>

#include <cmath>

int main () {

float x1 , x2 , y1 , y2 , z1 , z2 ;

printf("input x1: ") ;

scanf("%f",&x1) ;

printf("input x2: ") ;

scanf("%f",&x2) ;

printf("input y1: ") ;

scanf("%f",&y1) ;

printf("input y2: ") ;

scanf("%f",&y2) ;

printf("input z1: ") ;

scanf("%f",&z1) ;

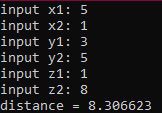
printf("input z2: ") ;

scanf("%f",&z2) ;

float d = sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2) + pow(z2 - z1, 2));

printf("distance = %f", d);

}



### 

### 1.3 Individual Assignment

You should write a program that provides file input and output and implements an assignment of previous laboratory training. You should implement following steps:

* definition of a constant (n) which determines column count of two-dimensional array
* opening file for reading (file should be prepared using some text editor)
* reading integer values until the end of file and storing them in the linked list
* creation of two-dimensional array in free store; row count should be calculated based on amount of integer values read from file and count of columns
* filling of two-dimensional array row by row; missing elements of the last row should be set to zeroes;
* removing elements of linked list from free store
* implementation of [previous assignment](http://iwanoff.inf.ua/programming_1_en/LabTraining04.htm)
* storing results in a new file
* removing both arrays using **delete** operators.

#include <iostream>

#include <stdio.h>

#include <math.h>

#include <cmath>

#include <fstream>

#include <string>

using namespace std;

struct Node

{

int data;

Node \*next;

};

struct LinkedList

{

Node \*head = NULL, \*tail = NULL;

void add\_node(int n)

{

Node \*tmp = new Node;

tmp->data = n;

tmp->next = NULL;

if (head == NULL)

{

head = tmp;

tail = tmp;

}

else

{

tail->next = tmp;

tail = tail->next;

}

}

};

int \*convertStrtoArr(string str)

{

int str\_length = str.length();

int \*arr = new int[str\_length];

for (int i = 0; str[i] != '\0'; i++)

{

arr[i] = 0;

}

int j = 0, i, sum = 0;

for (i = 0; str[i] != '\0'; i++)

{

if (str[i] == ',')

continue;

if (str[i] == ' ')

{

j++;

}

else

{

arr[j] = arr[j] \* 10 + (str[i] - 48);

}

}

return arr;

}

int main()

{

const int n = 3;

fstream myFile;

/\*

myFile.open("file.txt", ios::out ); //to write

if (myFile.is\_open()) {

myFile << "hello :/n";

myFile.close();

}

myFile.open("file.txt", ios::app); //to append

if (myFile.is\_open()) {

myFile << "hello2";

myFile.close();

}

\*/

//string myFile;

string line;

LinkedList list;

int count = 0;

int \*\*array;

myFile.open("file.txt", ios::in); //to read

if (myFile.is\_open())

{

while (getline(myFile, line))

{

//cout << line << endl ;

int \*arr = convertStrtoArr(line);

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

{

list.add\_node(arr[i]);

count++;

}

}

myFile.close();

}

array = new int \*[count];

Node \*head = list.head;

for (int k = 0; k < count; k++)

{

array[k] = new int[n];

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

{

if (head != NULL)

{

array[k][l] = head->data;

head = head->next;

}

else

{

array[k][l] = 0;

}

}

if (k > 3 && k < count - 1)

{

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

{

cout << "input element : ";

cin >> array[k][l];

}

}

}

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

{

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

{

cout << array[i][j] << ", ";

}

cout << endl;

}

delete head;

//----------------------------------------------------------------------------

int arra\_rows = 4;

int arra\_column = 3;

int i = arra\_rows;

int j = arra\_column;

int \*\*arra;

double \*arr;

arra = new int \*[4];

arr = new double[4];

for (int i = 0; i < 4; i++)

{

arra[i] = new int[i];

cout << " Input row " << i + 1 << " : " << endl;

for (int j = 0; j < 3; j++)

{

cout << "Input column " << j + 1 << " : ";

cin >> arra[i][j];

}

}

cout << " All items with odd values should be doubled : " << endl;

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

{

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

{

if (arra[s][v] % 2 != 0)

{

int d = arra[s][v] \* 2;

cout << "item : " << d << endl;

}

}

}

cout << " Square roots of minimal positive items of rows : " << endl;

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

{

int min;

int max;

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

{

if (arra[s][v] > 0)

{

if (v == 0)

{

max = arra[s][v];

min = arra[s][v];

}

if (max < arra[s][v])

{

max = arra[s][v];

}

if (min > arra[s][v])

{

min = arra[s][v];

}

}

}

cout << "row : " << s + 1 << endl;

cout << " Min: " << min << ", square root: " << sqrt(min) << endl;

arr[s] = sqrt(min);

}

//----------------------------------------------------------------------------

fstream newFile;

newFile.open("newfile.txt", ios::out); //to write

if (newFile.is\_open())

{

newFile << "result : " << endl;

newFile << " All items with odd values should be doubled : " << endl;

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

{

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

{

if (arra[s][v] % 2 != 0)

{

int d = arra[s][v] \* 2;

newFile << "item : " << d << endl;

}

}

}

newFile << " Square roots of minimal positive items of rows : " << endl;

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

{

int min;

int max;

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

{

if (arra[s][v] > 0)

{

if (v == 0)

{

max = arra[s][v];

min = arra[s][v];

}

if (max < arra[s][v])

{

max = arra[s][v];

}

if (min > arra[s][v])

{

min = array[s][v];

}

}

}

newFile << "row : " << s + 1 << endl;

newFile << " Min: " << min << ", square root: " << sqrt(min) << endl;

arr[s] = sqrt(min);

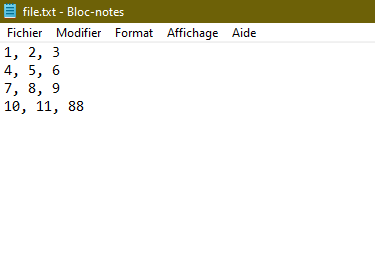
}

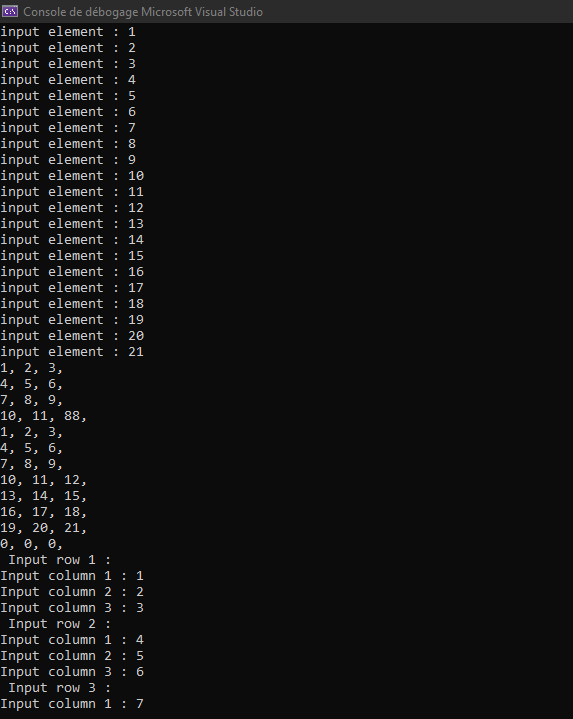
newFile.close();

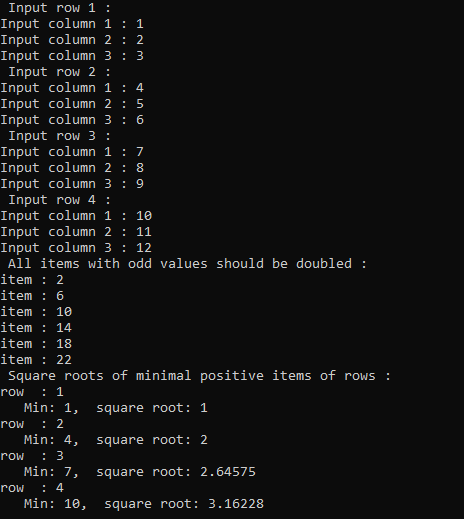
}

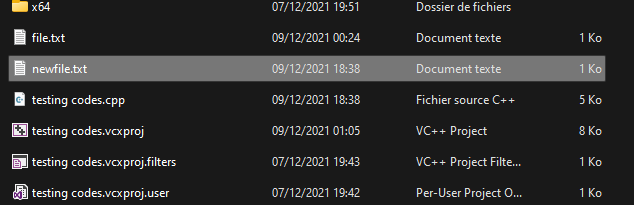
delete array, arra, arr;

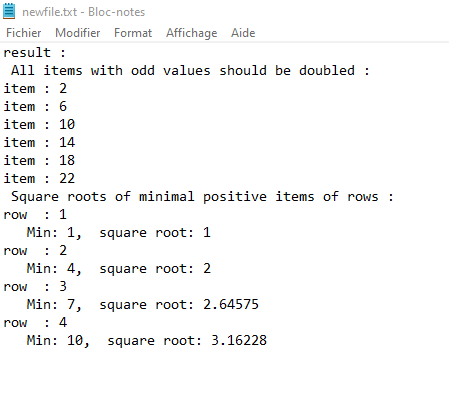
}











CONCLUSION :

Because of this lab, now I know how to deal with files of all kinds in c++.