**МІНІСТЕРСТВО ОСВІТИ І НАУКИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАІНИ**

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»**

**ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ**

**КАФЕДРА СИСТЕМНОГО ПРОГРАМУВАННЯ І СПЕЦІАЛІЗОВАНИХ КОМП`ЮТЕНИХ СИСТЕМ**

**ЛАБОРАТОРНАЯ РОБОТА №2**

з дисципліни «**Паралельні та розподілені обчислення**»

Тема: «**Робота з компіляторами мови C та Java в режимі командного рядка**»

|  |  |
| --- | --- |
|  | Виконав  студент 2 курсу  гр. КВ-43  Вітовщик Михайло Васильович  Перевірив(ла):  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Київ – 2016

***Постановка задачі***

**Постановка завдання для програми мовою C**

1. Написати програму розв’язання задачі пошуку (за варіантом) у двовимірному масиві (матриці) одним з алгоритмів методу лінійного пошуку.

2. Розміри матриці m та n взяти самостійно у межах від 7 до 10.

3. Програма обов'язково повинна бути написана і структурована наступним чином:

a) оголошення структур даних (typedef) повинно бути зроблено у окремому заголовочному файлі;

b) повинно бути щонайменше три файли із вихідним кодом (не враховуючи необхідні заголовочні файли), що міститимуть реалізації функцій введення (випадкові значення, наперед сортовані значення, з клавіатури), обробки, та виведення на друк (pretty\_print) елементів матриці;

c) для виконання завдання обробки елементів матриці повинно бути

написано дві різні функції:

1) з додатковими операторами виведення налагоджувальної інформації на друк (debug-версія);

2) з виконанням заданих дій без додаткового виведення налагоджувальної інформації (release-версія).

4. Для компіляції написаної багатофайлової програми написати окремий make-файл, причому:

a) при зміні одного із вихідних файлів повинен перекомпільовуватися лише цей файл (а також відбуватися дії, необхідні для генерації бінарного файлу);

b) при видаленні бінарного файлу та незмінних вихідних файлах повинна відбуватися лише лінковка;

c) забезпечити окрему ціль для очистки згенерованих файлів;

5. Вміти компілювати написану багатофайлову програму двома способами:

a) за допомогою однієї команди gcc;

b) за допомогою make-файлу.

6. Виконати тестування та налагодження програми на комп’ютері. При тестуванні програми необхідно підбирати такі вхідні набори початкових значень матриці, щоб можна було легко відстежити коректність виконання пошуку і ця коректність була б протестована для всіх можливих випадків. З метою тестування дозволяється використовувати матриці меншого розміру.

**Постановка завдання для програми мовою Java**

1. Написати консольну програму розв’язання задачі пошуку (за варіантом) у двовимірному масиві (матриці) одним з алгоритмів методу лінійного пошуку.

2. Розміри матриці m та n взяти самостійно у межах від 7 до 10.

3. При написанні програми повинно бути щонайменше три класи, один із яких буде відповідати за пошук елементу в матриці, другий відповідати за ввід-вивід матриці, а третій — головний клас, що міститиме метод main.

4. Для компіляції та запуску написаної програми написати окремий make-файл, причому забезпечити окремі цілі для очистки згенерованих файлів, а також генерації JAR-архіву.

5. Вміти компілювати написану програму двома способами:

a) за допомогою однієї команди javac;

b) за допомогою make-файлу.

6. Виконати тестування та налагодження програми на комп’ютері. При тестуванні програми необхідно підбирати такі вхідні набори початкових значень матриці, щоб можна було легко відстежити коректність виконання пошуку і ця коректність була б протестована для всіх можливих випадків. З метою тестування дозволяється використовувати матриці меншого розміру.

***3 варіант***

Задано матрицю дійсних чисел A[m,n]. При обході матриці по рядках знайти в ній перший додатний елемент і його місцезнаходження (координати).

***Текст програми мовою С***

***struct.h***

#pragma once

#define M 7

#define N 7

typedef struct{

double mas[M][N];

double first\_positive\_element;

int coord\_i;

int coord\_j;

} matrix;

**algo.h**

#pragma once

#include "struct.h"

// prototypes

void search\_first\_positive\_element(matrix \*);

**algo.c**

#include "algo.h"

void search\_first\_positive\_element(matrix \*matr){

int i, j;

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

for(j = 0; j < N; j++)

if(matr->mas[i][j] > 0){

matr->first\_positive\_element = matr->mas[i][j];

matr->coord\_i = i;

matr->coord\_j = j;

return;

}

matr->first\_positive\_element = -1;

}

**inout.h**

#pragma once

#include <stdio.h>

#include <stdlib.h>

#include "algo.h"

//prototypes

void creat\_sorted\_array(matrix \*);

void creat\_inversely\_sorted\_array(matrix \*);

void creat\_randomly\_sorted\_arrat(matrix \*);

void creat\_keyboard\_input(matrix \*);

void print\_matrix(matrix \*);

void print\_first\_positive\_element(matrix \*);

**inout.c**

#include "inout.h"

//input martix

void creat\_sorted\_array(matrix \*matr){

int i, j;

double elem = -N\*M;

srand(time(NULL));

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

for(j = 0; j < N; j++){

elem += rand()%3;

matr->mas[i][j] = elem / 10;

}

}

void creat\_inversely\_sorted\_array(matrix \*matr){

int i, j;

double elem = N\*M;

srand(time(NULL));

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

for(j = 0; j < N; j++){

elem -= rand()%5;

matr->mas[i][j] = elem / 10;

}

}

void creat\_randomly\_sorted\_array(matrix \*matr){

int i, j;

srand(time(NULL));

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

for(j = 0; j < N; j++)

matr->mas[i][j] = (rand()%(2\*N\*M) - N\*M) / 10.0;

}

void creat\_keyboard\_input(matrix \*matr){

printf("Enter elements the matrix [%d][%d]:\n", M, N);

int i, j;

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

for(j = 0; j < N; j++)

scanf("%lf", &(matr->mas[i][j]));

}

//print matrixt

void print\_matrix(matrix \*matr){

int i, j;

printf("Print matrix [%d][%d]:\n", M, N);

for(i = 0; i < M; i++){

for(j = 0; j < N; j++)

printf("%5.1lf", matr->mas[i][j]);

printf("\n");

}

printf("\n");

}

void print\_first\_positive\_element(matrix \*matr){

search\_first\_positive\_element(matr);

if (matr->first\_positive\_element < 0)

printf("Not found positive element!\n");

else

printf("First positive element [%d][%d]: %.1lf\n\n",

matr->coord\_i, matr->coord\_j, matr->first\_positive\_element);

}

**main.cpp**

#include "inout.h"

int main(){

matrix \*matr = (matrix \*) malloc(sizeof(matrix));

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

creat\_sorted\_array(matr);

print\_matrix(matr);

print\_first\_positive\_element(matr);

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

creat\_inversely\_sorted\_array(matr);

print\_matrix(matr);

print\_first\_positive\_element(matr);

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

creat\_randomly\_sorted\_array(matr);

print\_matrix(matr);

print\_first\_positive\_element(matr);

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

//creat\_keyboard\_input(matr);

//print\_first\_positive\_element(matr);

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

return 0;

}

***makefile***

source: source.o algo.o inout.o

gcc source.o algo.o inout.o -o main

source.o: source.c

gcc -c source.c

algo.o: algo.c algo.h struct.h

gcc -c algo.c

inout.o: inout.c inout.h struct.h

gcc -c inout.c

clean:

rm -rf \*.o main

***Командні рядки для компіляція та тест***

$ gcc -o source source.c inout.c algo.c

$ ./main

------------------------------------------------

Print matrix [7][7]:

-4.8 -4.6 -4.5 -4.4 -4.3 -4.2 -4.0

-4.0 -3.9 -3.8 -3.7 -3.5 -3.4 -3.2

-3.1 -3.0 -2.8 -2.7 -2.5 -2.4 -2.2

-2.1 -2.0 -1.8 -1.8 -1.7 -1.5 -1.4

-1.3 -1.1 -0.9 -0.9 -0.7 -0.6 -0.5

-0.4 -0.4 -0.3 -0.2 -0.1 -0.1 -0.1

0.0 0.2 0.2 0.2 0.3 0.3 0.4

First positive element [6][1]: 0.2

------------------------------------------------

Print matrix [7][7]:

4.8 4.8 4.6 4.4 4.1 3.9 3.5

3.4 3.2 2.9 2.7 2.3 2.0 1.9

1.9 1.7 1.7 1.5 1.1 1.0 0.9

0.6 0.6 0.2 0.0 0.0 -0.2 -0.4

-0.5 -0.6 -0.6 -0.9 -1.2 -1.6 -1.6

-1.8 -2.0 -2.1 -2.4 -2.8 -2.9 -3.1

-3.5 -3.5 -3.5 -3.6 -4.0 -4.1 -4.4

First positive element [0][0]: 4.8

------------------------------------------------

Print matrix [7][7]:

-2.3 -2.0 -2.7 0.2 4.6 4.0 2.6

-3.2 -2.5 3.3 2.3 1.9 -0.5 3.5

4.0 3.7 -3.8 -4.7 3.7 4.6 2.6

-4.9 -0.8 -0.3 -3.2 4.5 1.2 -1.1

2.6 -3.5 0.9 -4.5 4.8 -1.3 0.6

4.6 -2.1 3.7 -3.5 0.3 -2.3 -0.7

-2.6 2.2 3.3 1.9 -3.4 4.5 2.1

First positive element [0][3]: 0.2

------------------------------------------------

***Текст програми мовою Java***

***\_struct/struct.java***

package \_struct;

public class struct{

public double [][]mas;

public double first\_positive\_element;

public int coord\_i;

public int coord\_j;

public int M, N;

public struct(int m, int n){

mas = new double[m][n];

first\_positive\_element = -1;

M = m;

N = n;

}

}

***\_algo/algo.java***

package \_algo;

import \_struct.\*;

public class algo{

public static void search\_first\_positive\_element(struct matr){

int i, j;

for(i = 0; i < matr.M; i++)

for (j = 0; j < matr.N; j++)

if(matr.mas[i][j] > 0){

matr.first\_positive\_element = matr.mas[i][j];

matr.coord\_i = i;

matr.coord\_j = j;

return;

}

}

}

***\_inout/inout.java***

package \_inout;

import java.util.Scanner;

import \_struct.\*;

import \_algo.\*;

public class inout{

//input matrix

public void creat\_sorted\_array(struct matr){

int i, j;

int elem = -matr.N\*matr.M;

for(i = 0; i < matr.M; i++)

for(j = 0; j < matr.N; j++){

elem += (i\*j+i+j+i)%3;

matr.mas[i][j] = elem / 10.0;

}

}

public void creat\_inversely\_sorted\_array(struct matr){

int i, j;

double elem = matr.N\*matr.M;

for(i = 0; i < matr.M; i++)

for(j = 0; j < matr.N; j++){

elem -= (i\*j+j+i)%5;

matr.mas[i][j] = elem / 10;

}

}

public void creat\_randomly\_sorted\_array(struct matr){

int i, j;

for(i = 0; i < matr.M; i++)

for(j = 0; j < matr.N; j++)

matr.mas[i][j] = ((2\*matr.N\*matr.M\*i)%15 - (matr.N\*matr.M)%(i+j+5)) / 10.0;

}

public void creat\_keyboard\_input(struct matr){

System.out.printf("Enter elements the matrix [%d][%d]:\n", matr.M, matr.N);

int i, j;

Scanner in = new Scanner(System.in);

for(i = 0; i < matr.M; i++)

for(j = 0; j < matr.N; j++)

matr.mas[i][j] = in.nextDouble();

}

// output matrix

public void print\_matrix(struct matr){

int i, j;

System.out.printf("Print matrix [%d][%d]:\n", matr.M, matr.N);

for(i = 0; i < matr.M; i++){

for(j = 0; j < matr.N; j++)

System.out.printf("%5.1f", matr.mas[i][j]);

System.out.printf("\n");

}

System.out.printf("\n");

}

public void print\_first\_positive\_element(struct matr){

algo al = new algo();

al.search\_first\_positive\_element(matr);

if (matr.first\_positive\_element < 0)

System.out.printf("Not found positive element!\n");

else

System.out.printf("First positive element [%d][%d]: %.1f\n\n",

matr.coord\_i, matr.coord\_j, matr.first\_positive\_element);

}

}

***source.java***

import \_struct.\*;

import \_inout.\*;

public class source{

public static void main(String[] args){

struct matr = new struct(3,2);

inout io = new inout();

System.out.printf("----------------------------------------\n");

io.creat\_sorted\_array(matr);

io.print\_matrix(matr);

io.print\_first\_positive\_element(matr);

System.out.printf("----------------------------------------\n");

io.creat\_inversely\_sorted\_array(matr);

io.print\_matrix(matr);

io.print\_first\_positive\_element(matr);

System.out.printf("----------------------------------------\n");

io.creat\_randomly\_sorted\_array(matr);

io.print\_matrix(matr);

io.print\_first\_positive\_element(matr);

System.out.printf("----------------------------------------\n");

io.creat\_keyboard\_input(matr);

io.print\_first\_positive\_element(matr);

System.out.printf("----------------------------------------\n");

}

}

***makefile***

main: source.java \_struct/struct.java \_algo/algo.java \_inout/inout.java

javac source.java \_struct/struct.java \_algo/algo.java \_inout/inout.java

jar: source.class

jar -cfm source.jar manifest source.class \_struct/struct.class \_algo/algo.class \_inout/inout.class

source.class: source.java \_struct/struct.java \_algo/algo.java \_inout/inout.java

javac source.java \_struct/struct.java \_algo/algo.java \_inout/inout.java

clean:

rm -rv \*.class \_struct/\*.class \_algo/\*.class \_inout/\*.class

***Командні рядки для компіляція та тест***

$ javac source.java \_struct/struct.java \_algo/algo.java \_inout/inout.java

$ java source

----------------------------------------

Print matrix [3][2]:

-0,6 -0,5

-0,3 -0,2

-0,1 0,0

Not found positive element!

----------------------------------------

Print matrix [3][2]:

0,6 0,5

0,4 0,1

-0,1 -0,1

First positive element [0][0]: 0,6

----------------------------------------

Print matrix [3][2]:

-0,1 0,0

1,2 0,6

0,3 0,3

First positive element [1][0]: 1,2

----------------------------------------

Enter elements the matrix [3][2]:

1 2

1 3

1 2

First positive element [0][0]: 1,0

----------------------------------------