КПІ ім. Ігоря Сікорського

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт до комп’ютерного практикуму з курсу “Основи програмування”

Прийняв
асистент кафедри ІПІ
Пархоменко А. В.
07.12.2022 р.

Виконала
студентка групи ІП-22
Андреєва У.А.

Київ 2022

Комп’ютерний практикум №5

Тема: *Покажчики і масиви. Робота з рядками.*

Завдання: *Написати програму для впорядкування масиву рядків, використовуючи покажчики.*

Текст програми

```
#include <stdio.h>
#include <time.h> //we need it to call randomizer in C
#include <stdlib.h>
#include <string.h>

/*listString - matrix (arrays of arrays)
listString[i] - massive
listString[i][j] - element of matrix*/

int random_range(int range_min, int range_max); //we bring our function to the top
void output(int quantityLine, int quantitySymbols, char matrix[quantityLine][quantitySymbols]);
void copy_array(const char *from, char *to, int symbolsAmount);

int main()
{
   srand(time(NULL)); //randomizer in C, which attaches to time in your device

   char ch,error; //set the variables
   int quantityLine;
   int quantitySymbols;

   do
   {
      error=0;
      printf("Enter lines amount(columns): "); //checking symbols
      scanf("%d%c",&quantityLine,&ch);
```

```
        if ((ch!='\n')||(quantityLine<=0))//checking buffering zone inputting (stdin) + substracrion check
        {
            error = 1;
            fflush(stdin);
            printf("Error inputting\n");
        }
        ch = 0;

    }
    while(error);

    do
    {
        error=0;
        printf("Enter amount of symbols in one line (rows) : ");  //checking symbols
        scanf("%d%c",&quantitySymbols,&ch);
            if ((ch!='\n')||(quantitySymbols<=0)) //checking buffering zone inputting (stdin) + substracrion check
        {
            error = 1;
             fflush(stdin);
              printf("Error inputting\n");
        }
        ch = 0;
    }
    while(error);

     char listString [quantityLine][quantitySymbols]; //creating two massives in one matrix(listString)

     for (int i = 0 ; i < quantityLine; i++) { //creating two loops for lines(i) and for crossing row and column(j)
        for (int j = 0; j < quantitySymbols; j++) {
            char ch;
            do{
                ch = (char)random_range(65,122);
                    /* creating the random range relying on the the table ASCII,we have appropriate diapazone
```

```
            from 65 to 125 */
          } while((int)ch <= 96 && (int)ch >= 91);/*but here we have checking only for letters, cause in that diapazone, we have also symbols*/
          listString[i][j] = ch; //char assignment to elements of massive
      }
   }
   output(quantityLine,quantitySymbols, listString);
      for(int i=0;i<quantityLine-1;i++){ //here we operate sorting operations in increasing manner with lines
      for(int j=i+1;j<quantityLine;j++){
         if(strcmp(listString[i],listString[j]) > 0){ /*comparing two lines,their ASCII codes, that continues to the end of process of finding out all different symbols or to the end of the line */
            char tmp[quantitySymbols];
            copy_array(listString[i], tmp, quantitySymbols);
            copy_array(listString[j], listString[i], quantitySymbols);
            copy_array(tmp, listString[j], quantitySymbols);
         }
      }
   }
   output(quantityLine, quantitySymbols, listString);
}
int random_range(int range_min, int range_max ){ //returning random int number from range_min to range_max diapazone
    return (range_min + rand() % (range_max - range_min +1 ));//getting random value
}
void output(int quantityLine, int quantitySymbols, char matrix[quantityLine][quantitySymbols]){
   printf("\n");
   for (int i = 0 ; i < quantityLine; i++) {
      for (int j = 0; j < quantitySymbols; j++) {
         printf("%c ", matrix[i][j]);
      }
      printf("\n");
   }
}
```

```
void copy_array(const char *from, char *to, int symbolsAmount){ /*we need to overturn our massive one by one,so we set massive from,massive to and exactly size of massive*/
    for (int i = 0; i < symbolsAmount; ++i) { /* with the help of that loop, we overturn symbols from massive (from) to massive (to)*/
      to[i] = from[i];
   }
}
```

Введенні та одержані результати

```
Enter lines amount(columns): 10
Enter amount of symbols in one line (rows) : 5

w u Q r Q
H B U P K
k d V P w
b O y d y
d G Y U U
M t V H A
Q R a w z
s G G w Y
H E T X L
O U P D g

H B U P K
H E T X L
M t V H A
O U P D g
Q R a w z
b O y d y
d G Y U U
k d V P w
s G G w Y
w u Q r Q
Program ended with exit code: 0
```

All Output

```
Enter lines amount(columns): 4
Enter amount of symbols in one line (rows) : 6

u x J l Z U
z z V h u c
a Z g G q v
I n M W G c

I n M W G c
a Z g G q v
u x J l Z U
z z V h u c
Program ended with exit code: 0
```

All Output

Теоретичні розрахунки

```
Enter lines amount(columns): 3
Enter amount of symbols in one line (rows) : 3

S j p
R D D
r J H

R D D
S j p
r J H
Program ended with exit code: 0
```

Здійснимо порівняння символів через таблицю ASCII :

- S = 83, j = 106, p = 112 => 1 String.
- R = 82, D = D = 68 => 2 String.
- r = 114, J = 74, H = 72 => 3 String.

S < R < r -> RDD < Sjp < rJH- відсортовано

Висновок

Отже, перевіривши всі значення, програма працює коректно та правильно сортує масив.