

Министерство образования Республики Беларусь

Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных средств

**ОТЧЁТ**  
по лабораторной работе №7  
«Функции. Строки»

Выполнил:  
Студент  
Студент гр. 150702-1п  
Ефимчик А.И.

Проверил:  
Ст. преп. каф. ЭВС  
Демидович Г. Н.

Минск 2021

## 1. ЦЕЛЬ РАБОТЫ

Изучить механизм создания пользовательских функций, разобраться с механизмом обмена информации между функциями.

## 2. ЗАДАНИЕ

Создать программу, которая выполняет действия из задания 1 и задания 2.

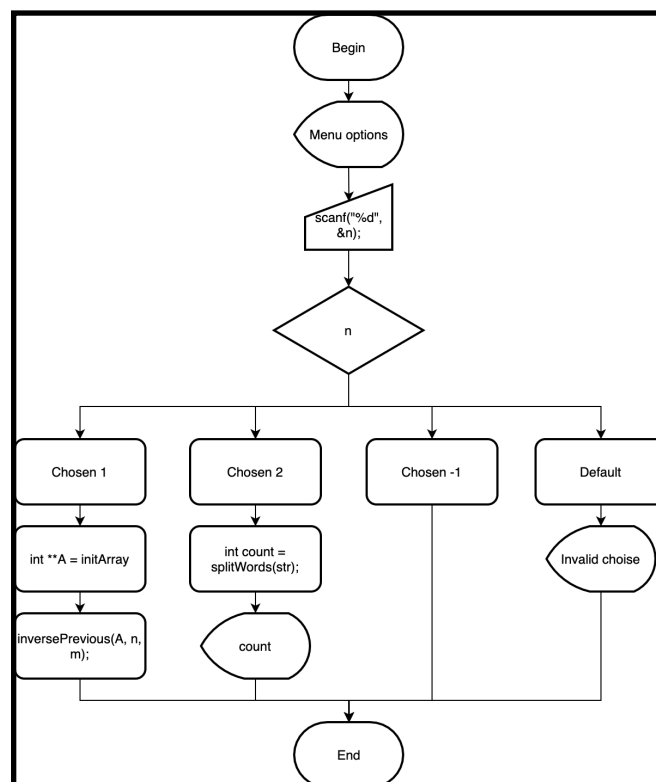
1. Проверить, есть ли в матрице хотя бы одна строка, содержащая положительный элемент, и найти ее номер. Знаки элементов предыдущей строки изменить на противоположные.

2. Дана строка. Необходимо определить количество слов, которые слева и справа читаются одинаково (палиндромы). Слова разделены пробелами.

В программе должно быть реализовано меню, каждое действие должно быть реализовано в отдельной функции.

## 3. ХОД РАБОТЫ

Блок-схема для выполнения задания №1 представлена ниже:



Исходный код программы для выполнения задания №1 представлен

ниже:

```
#include <stdio.h>
#include <stdlib.h>
#include <strings.h>

int **initArray(int width, int height);
void inversePrevious(int **arr, int width, int height);

int checkPalindrom (char* s, int start, int end);
int splitWords(char* s);

int main()
{
    printf("LAB 7 by Efimchik Alexandr from GROUP 150702 \n");
    printf("Select menu option: \n");
    printf(" 1) Inverse previous row signs\n");
    printf(" 2) Count palindroms\n");
    int choise_index;
    scanf("%d", &choise_index);
    switch (choise_index)
    {
        case 1:
            printf("Enter array sizes: ");
            int m = 0; // rows
            int n = 0; // column
            scanf("%d %d", &n, &m);
            int **A = initArray(n, m);
            inversePrevious(A, n, m);
            break;
        case 2:
            printf("Cheking palindromes in string\n");
            char str[13] = {'e','b','b','e', ' ', 's', 'a', 'g', ' ', 'a', 'b', 'b',
'a'};
            int count = splitWords(str);
            printf("There are %d palindrome words\n", count);
            break;
        case -1:
    }
```

```

        return 0;

    default:
        printf("Invalid choise");
        break;
    }
}

int **initArray(int width, int height)
{
    int **ptr = (int **)calloc((height), sizeof(int *));
    for (int i = 0; i < height; i++)
    {
        *(ptr + i) = (int *)calloc(width, sizeof(int));
        for (int j = 0; j < width; j++)
        {
            scanf("%d", &ptr[i][j]);
        }
    }
    return ptr;
}

void inversePrevious(int **arr, int width, int height)
{
    int num = -1;
    int flag = 0;
    for (int i = 0; i < height; i++)
    {
        for (int j = 0; j < width; j++)
        {
            if (arr[i][j] > 0)
            {
                flag = 1;
                num = i;
                break;
            }
        }
        if (flag)
            break;
    }
}

```

```

    }
    printf("\n%d row consists above zero value\n\n", num);
    if (num == 0)
    {
        printf("First row consists above zero vanlue, passing ...\n");
    }
    else
    {
        for (int j = 0; j < width; j++)
        {
            int temp = arr[num - 1][j];
            arr[num - 1][j] = temp * -1;
        }
    }
    for (int i = 0; i < height; i++)
    {
        for (int j = 0; j < width; j++)
        {
            (i == num - 1) ? (printf("\033[0;34m%d\033[0m ", arr[i][j])) : (printf("%d", arr[i][j]));
        }
        printf("\n");
    }
}

int splitWords(char* s) {
    int start = 0;
    int end = 0;
    int result = 0;
    int counter = 0;
    for (int j = 0; j < 12; j++)
    {
        if (s[j] == ' ')
        {
            end = j-1;
            result = checkPalindrom(s, start, end);
            counter += result;
            start = j + 1;
        }
    }
}

```

```

    }
    result = checkPalindrom(s, start, 12);
    counter += result;
    return counter;
}

int checkPalindrom (char* s, int start, int end) {
    for (int i = start; i < end; i++)
    {
        if (s[i] != s[end-i+start])
        {
            return 0;
        }
    }
    return 1;
}

```

Результат выполнения программы:

```

LAB 7 by Efimchik Alexandr from GROUP 150702
Select menu option:
  1) Inverse previous row signs
  2) Count palindroms
1
Enter array sizes: 3
3
-1
-2
-3
1
2
3
0
0
0
1 row consists above zero value
1 2 3
1 2 3
0 0 0

```

```

LAB 7 by Efimchik Alexandr from GROUP 150702
Select menu option:
  1) Inverse previous row signs
  2) Count palindroms
2
Cheking palindromes in string
There are 2 palindrome words

```

#### 4. ВЫВОД

В ходе выполнения данной лабораторной работы были выполнены ключевые пункты. Программа отлажена и работает корректно.