Гомельский государственный университет

Им. Франциска Скорины

**Лабораторная работа №7**

Тема: «Функции. Строки.»

Вариант 6

Выполнил студент группы

МС-12

Молокович С.В.

Гомель 2018

**Задание 1.**

Проверить, есть ли в матрице хотя бы одна строка, содержащая отрицательный элемент, и найти ее номер.

**Текст программы:**

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

int arr[50][50];

void vvod (int arr[50][50], int n1, int m1);

int vivod (int arr[50][50], int n1, int m1);

int otr (int arr[50][50], int n1, int m1);

int main(int argc, char \*argv[]) {

setlocale (LC\_ALL, "Russian");

int \*n,\*m,i;

puts ("Введите n:");

scanf ("%d", &n);

puts ("Введите m:");

scanf ("%d", &m);

vvod (arr, n, m);

vivod (arr, n ,m);

otr (arr,n,m);

}

void vvod (int arr[50][50], int n1, int m1)

{

int i,j;

for (i=0;i<n1;i++)

for (j=0;j<m1;j++)

{

printf ("Введите элемент массива arr[%d,%d]: ", i+1,j+1);

scanf ("%d", &arr[i][j]);

}

}

int otr (int arr[50][50], int n1, int m1)

{

puts ("Номера строк с отрицательными элементами: ");

int i,j,k;

for (i=0;i<n1;i++)

for(j=0;j<m1;j++)

{

if (arr[i][j]<0)

{

printf ("%d ", i+1);

}

}

}

int vivod (int arr[50][50], int n1, int m1)

{

int i,j;

printf ("\n");

for (i=0;i<n1;i++)

{

for (j=0;j<m1;j++)

printf ("%d ", arr[i][j]);

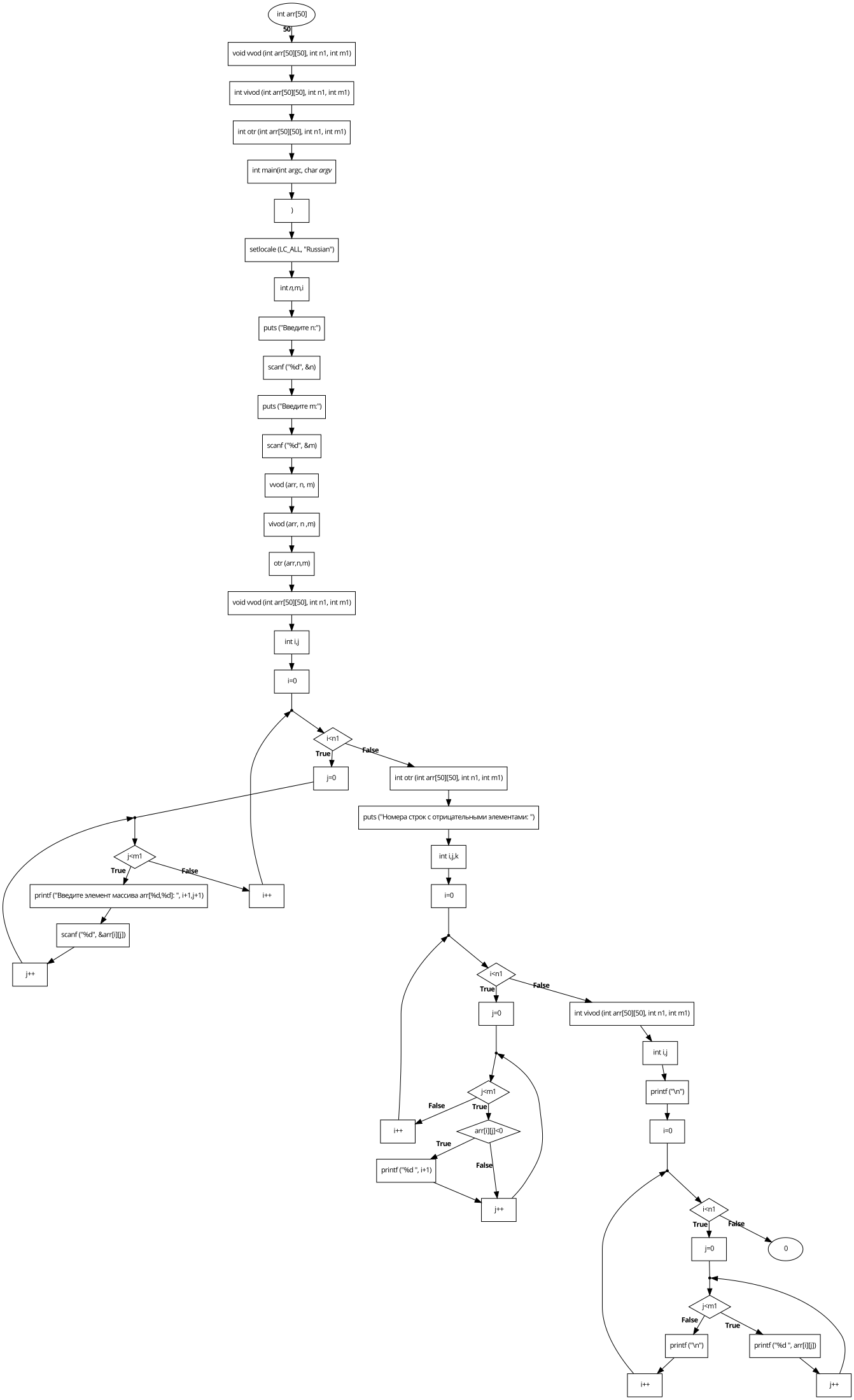
printf ("\n");

}

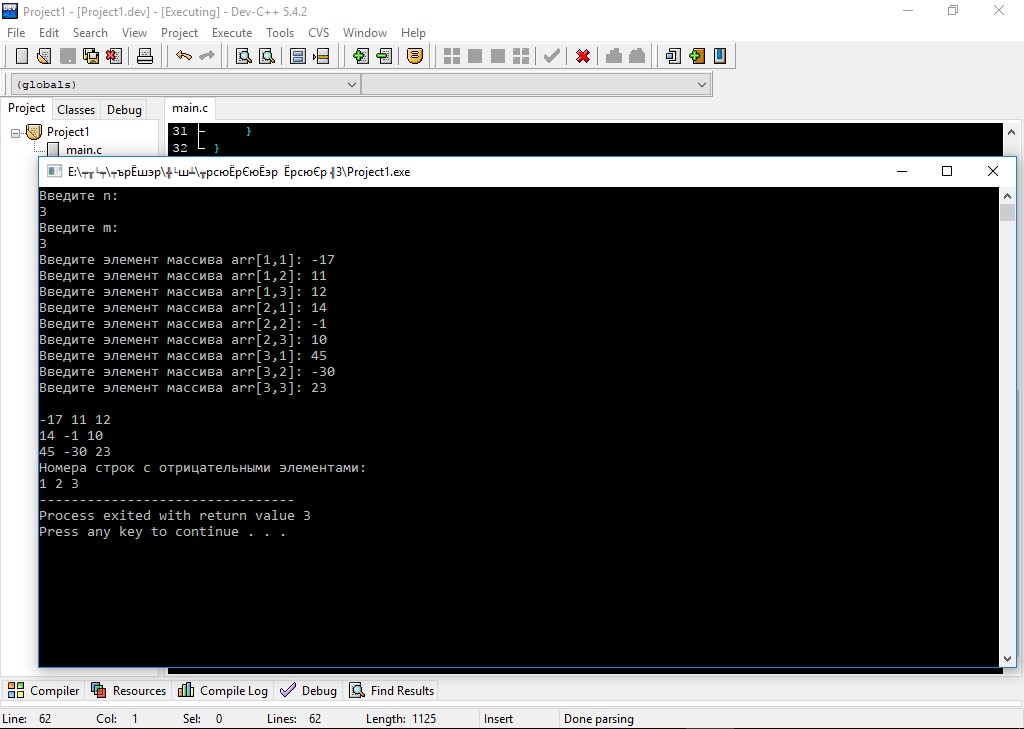
return 0;

}

**Блок-схема алгоритма:**

****

**Скриншот работы программы:**

****

**Задание 2.**

Ввести строку. Найти в функции в ней слово минимальной длины.

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

#include <math.h>

#include <string.h>

void dln (char \*s);

char s1[100];

int main()

{

setlocale (LC\_ALL, "Russian");

printf ("Введите строку:");

gets (s1);

dln (s1);

}

void dln(char \*s)

{

int i,k,min;

min=100;

i=0;

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

{

if (s[i]!=' ')

{

k++;

}

else

{

if (k<min)

{

min=k;

}

k=0;

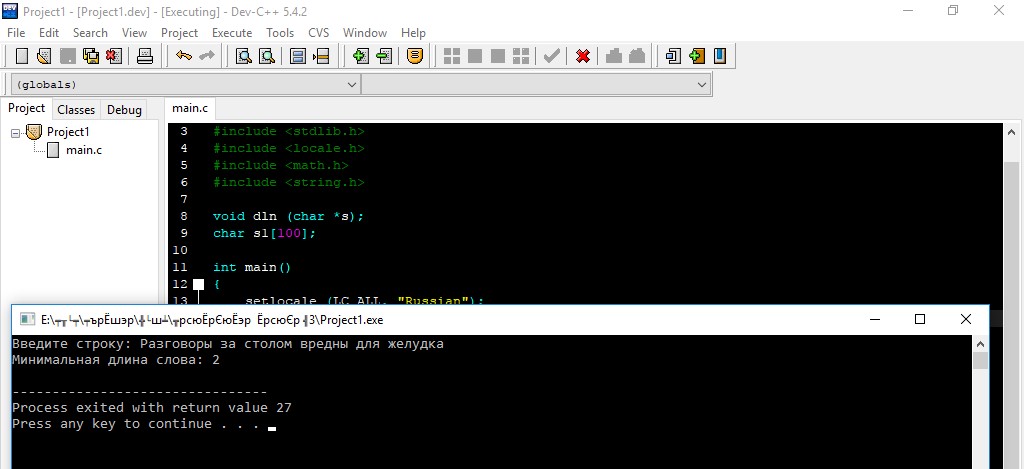
}

}

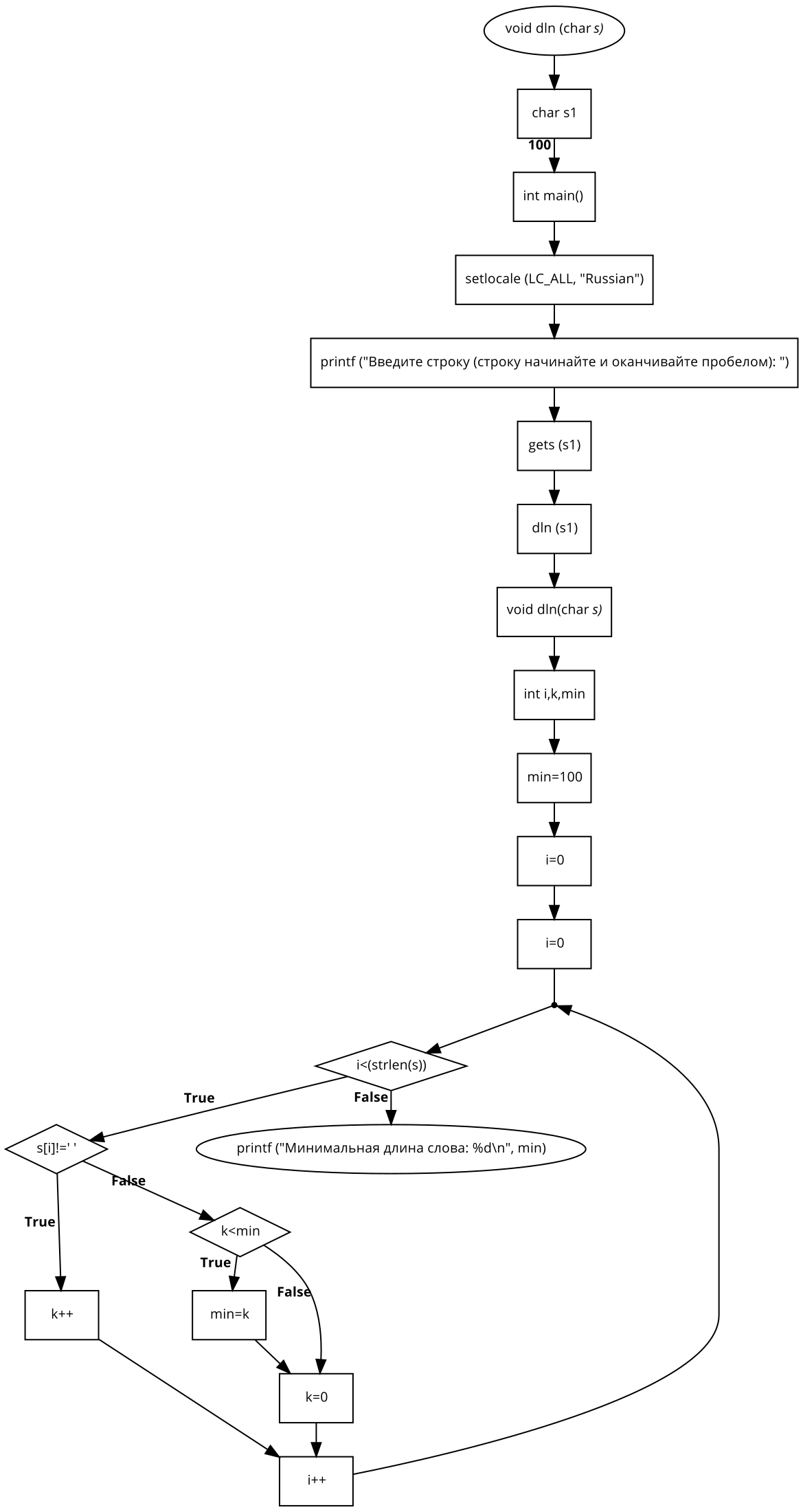
printf ("Минимальная длина слова: %d\n", min);

}

**Скриншот работы программы:**

****

**Блок-схема алгоритма:**

****