Министерство образования и науки Российской Федерации

Федеральное агентство по образованию

КАЗАНСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. А.Н.Туполева - КАИ

Кафедра АСОИУ

Лабораторная работа № 6

по дисциплине

«Основы программирования»

……………………………………………………………………………………

ОБРАБОТКА МАССИВОВ

Обучающийся 4109 Сагиров А.И.

Принял старший препод. каф. АСОИУ Бикмурзина А. Р.

Казань 2020

**Выполнение работы**

**Задание 1**

1. Дана строка символов. Признак конца строки – символ ‘\n’ (перевод строки). Строка состоит из слов, разделенных пробелами. Вывести:

Д) самое короткое слово и его длину

Код программы:

#include <stdio.h>

#include <Windows.h>

#define L 51

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251); // для корректного ввода и вывода кириллицы

int i, j, n, k, lmin = 100, nmin, kmin;

int s; // число одинаковых наименее коротких слов

char str[L];

printf("Введите строку символов (не более %d):\n", L-1);

gets\_s(str);

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

{

if (str[i] != ' ')

{

n = i;

while (str[i] != ' ' && str[i] != '\0')

++i;

k = i - 1;

if (k - n + 1 == lmin) ++s;

if (k - n + 1 < lmin)

{

nmin = n; kmin = k;

lmin = kmin - nmin + 1;

s = 1;

}

}

}

if (s != 1) printf("%d одинаково наименее коротких слов, т.е нет самого короткого\n", s);

else

{

puts("Самое короткое слово: ");

for (j = nmin; j <= kmin; ++j)

putchar(str[j]);

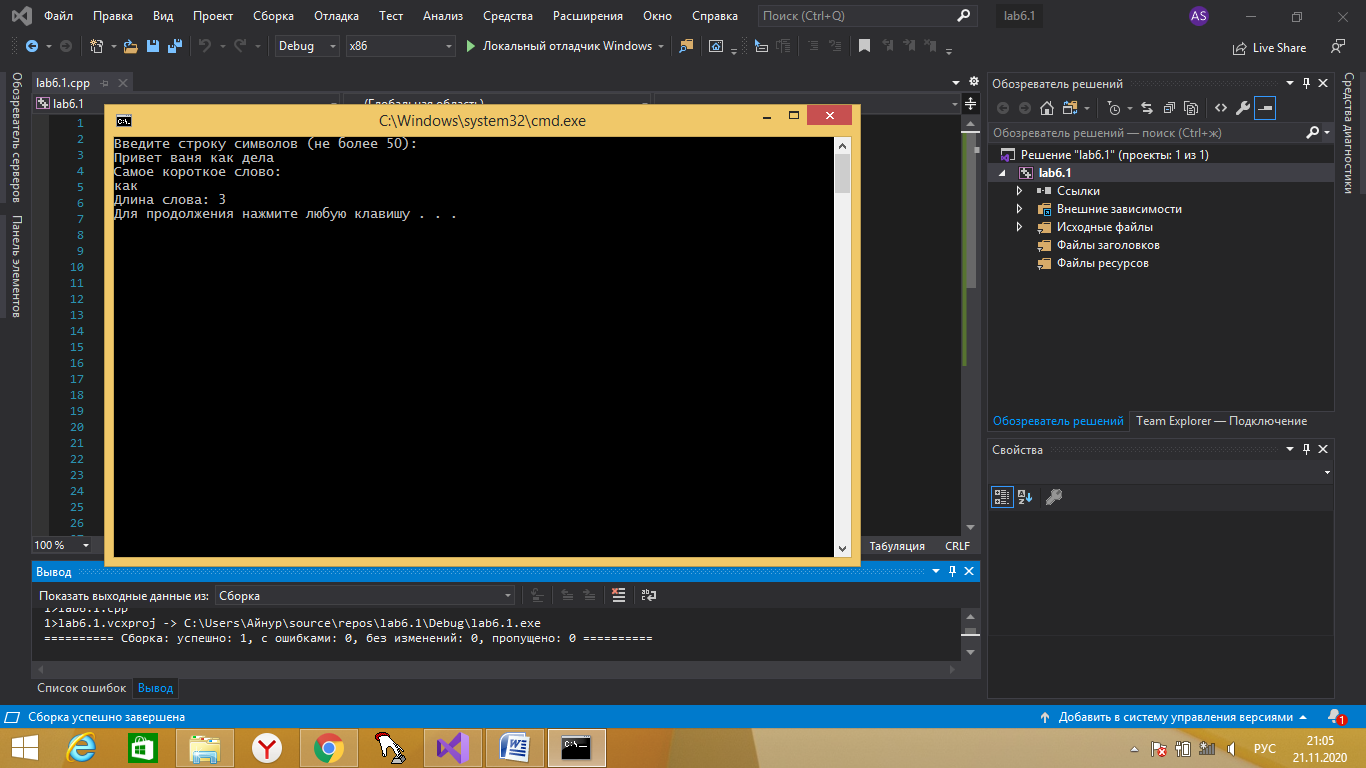
printf("\nДлина слова: %d\n", lmin);

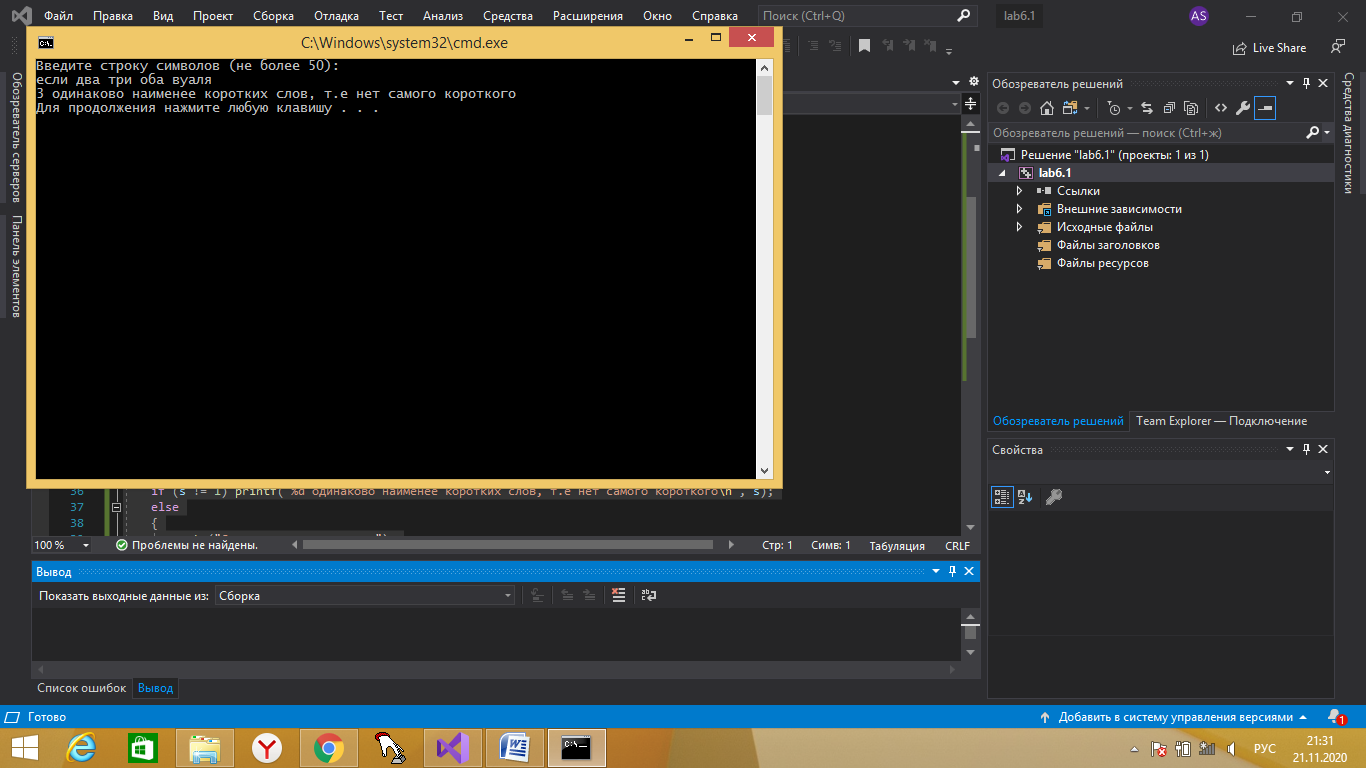
}

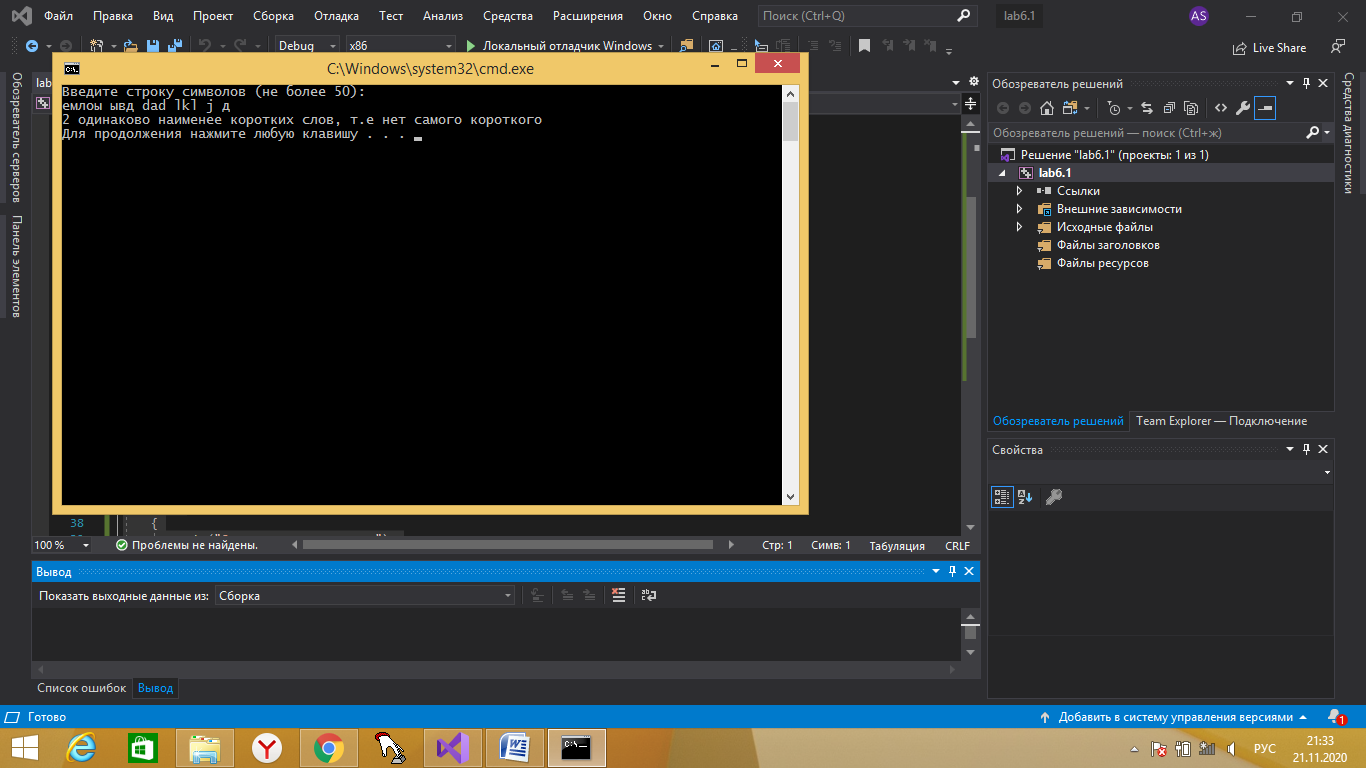
return 0;

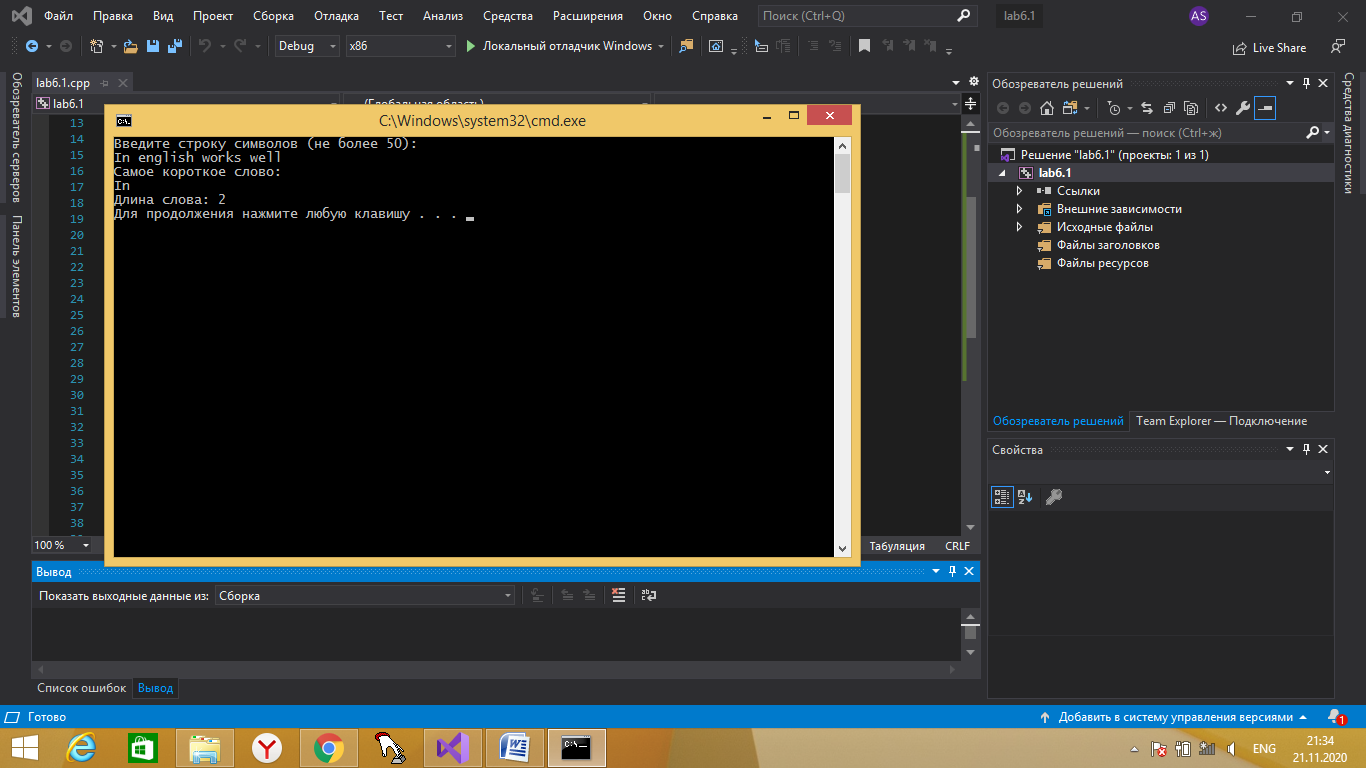
}

Результаты программы:









**Задание 2**

1. Дан массив, состоящий из n целых чисел (n<=10). Необходимо:

В) упорядочить массив по убыванию элементов методом последовательного нахождения минимума

Код программы:

#include <stdio.h>

#include <locale.h>

#define NMAX 10

int main()

{

setlocale(LC\_ALL, "Rus");

int x[NMAX];

int i, n, r, imin, k;

puts("Введите количество чисел (не более 10):");

scanf\_s("%d", &n);

puts("Введите целые числа:");

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

scanf\_s("%d", &x[i]);

for (k = n-1; k > 0; --k)

{

imin = k;

for (i = k; i >= 0; --i)

{

if (x[i] < x[imin]) imin = i;

}

r = x[imin];

x[imin] = x[k];

x[k] = r;

}

puts("Упорядоченные по убыванию числа:");

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

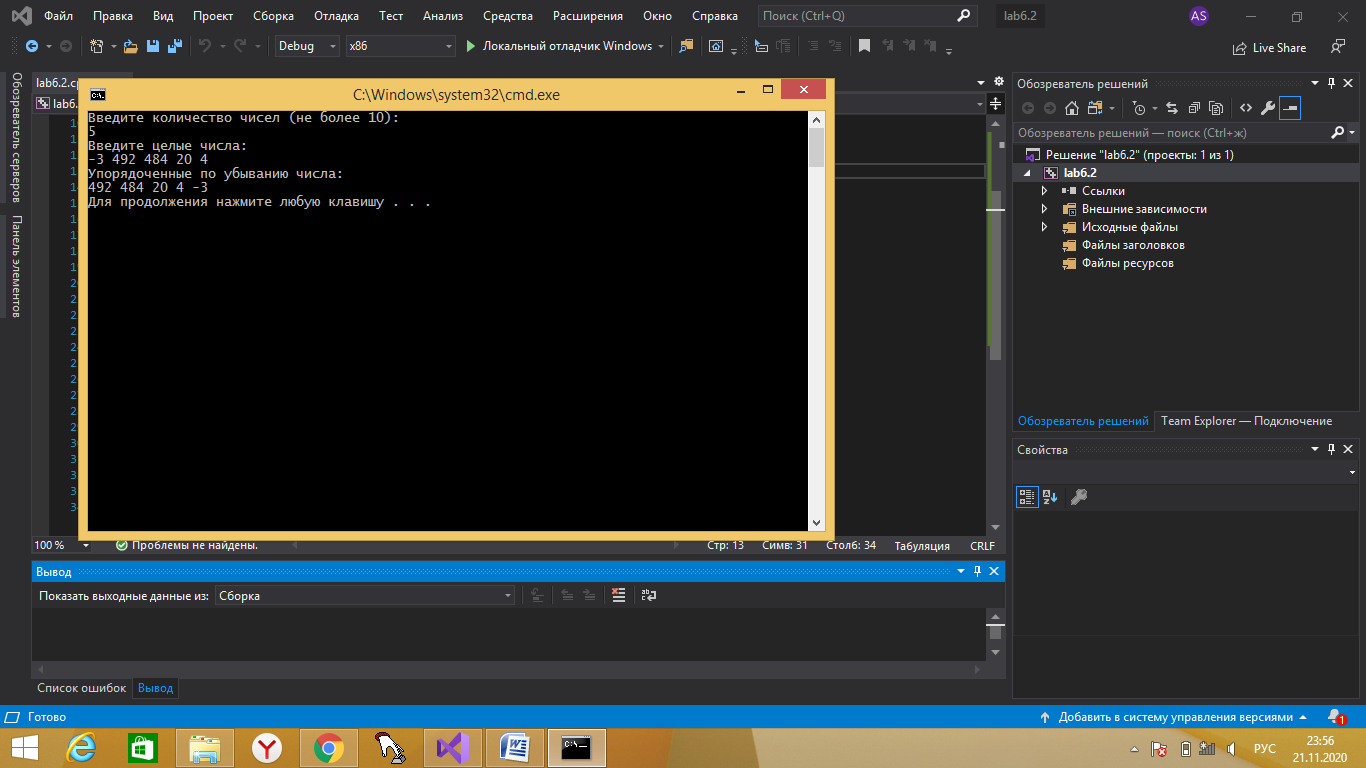
printf("%d ", x[i]);

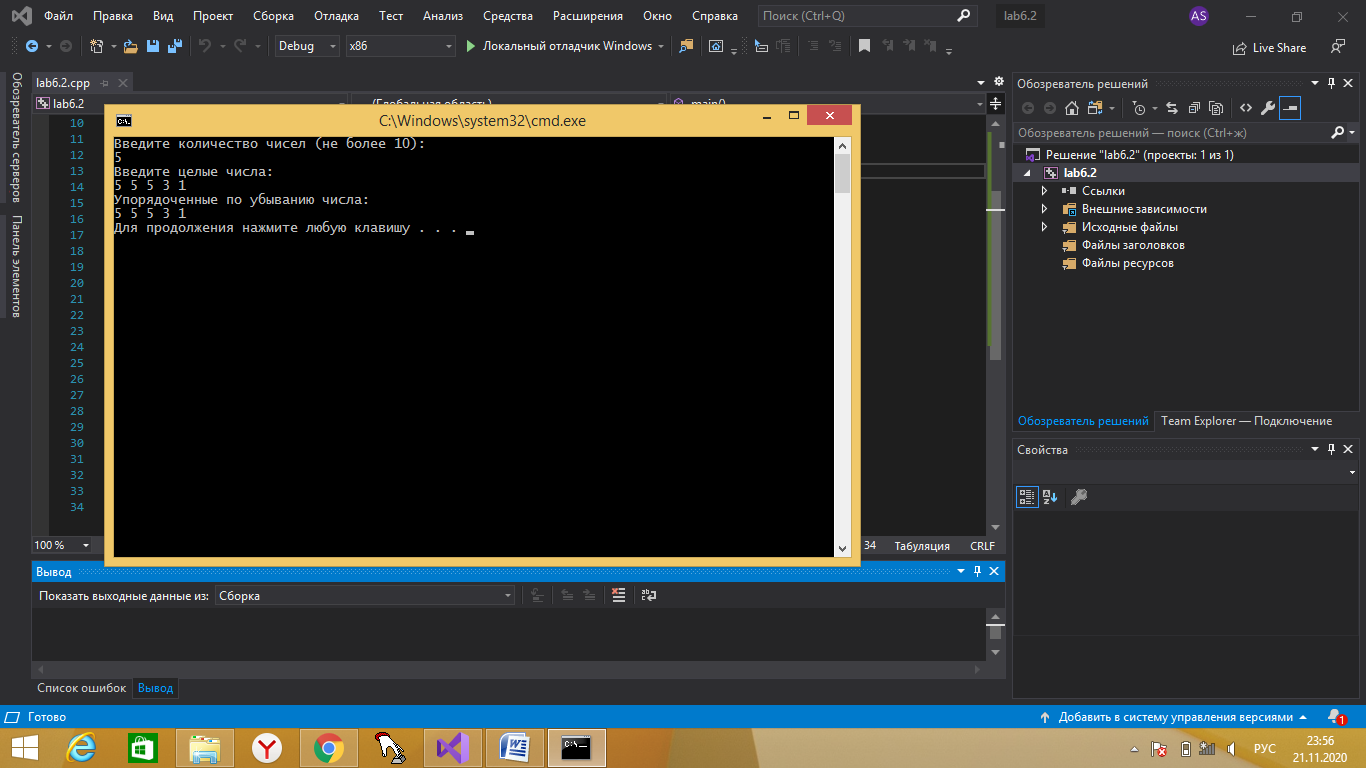
printf("\n");

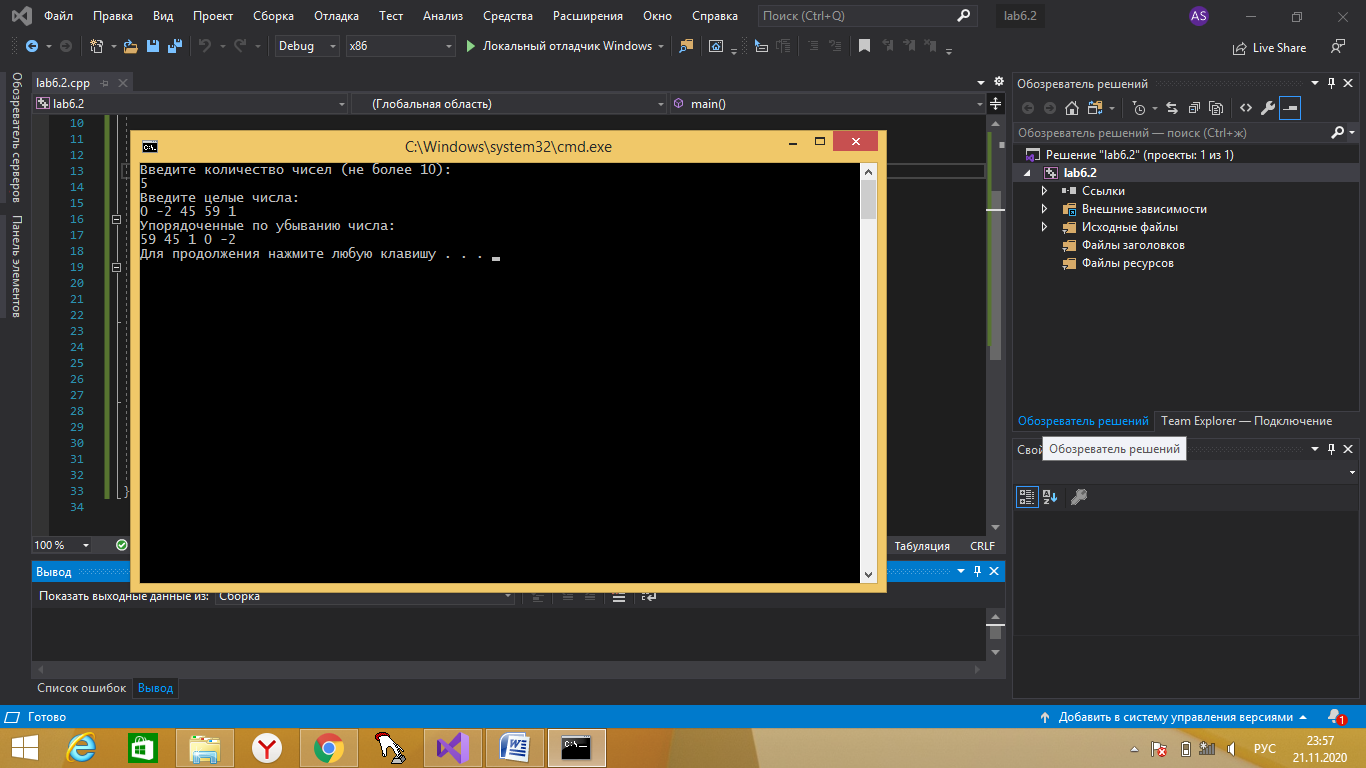
return 0;

}

Результаты программы:







**Задание 3**

1. Дана строка длиной не более 80 символов. Подчеркнуть все гласные буквы в строке.

Код программы:

#include <stdio.h>

#include <Windows.h>

#define N 81

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char s1[N], s2[12] = { 'e', 'E', 'y', 'Y', 'i', 'I', 'o', 'O', 'a','A', 'u', 'U'};

int i, j;

printf("Введите строку (не более 80 символов): \n");

gets\_s(s1);

for (i = 0; s1[i] != '\0'; ++i)

{

for (j = 0; j < 12; ++j)

{

if (s1[i] == s2[j])

{

putchar('-');

break;

}

}

if (j == 12) putchar(' '); // если j=12 => не было гласного

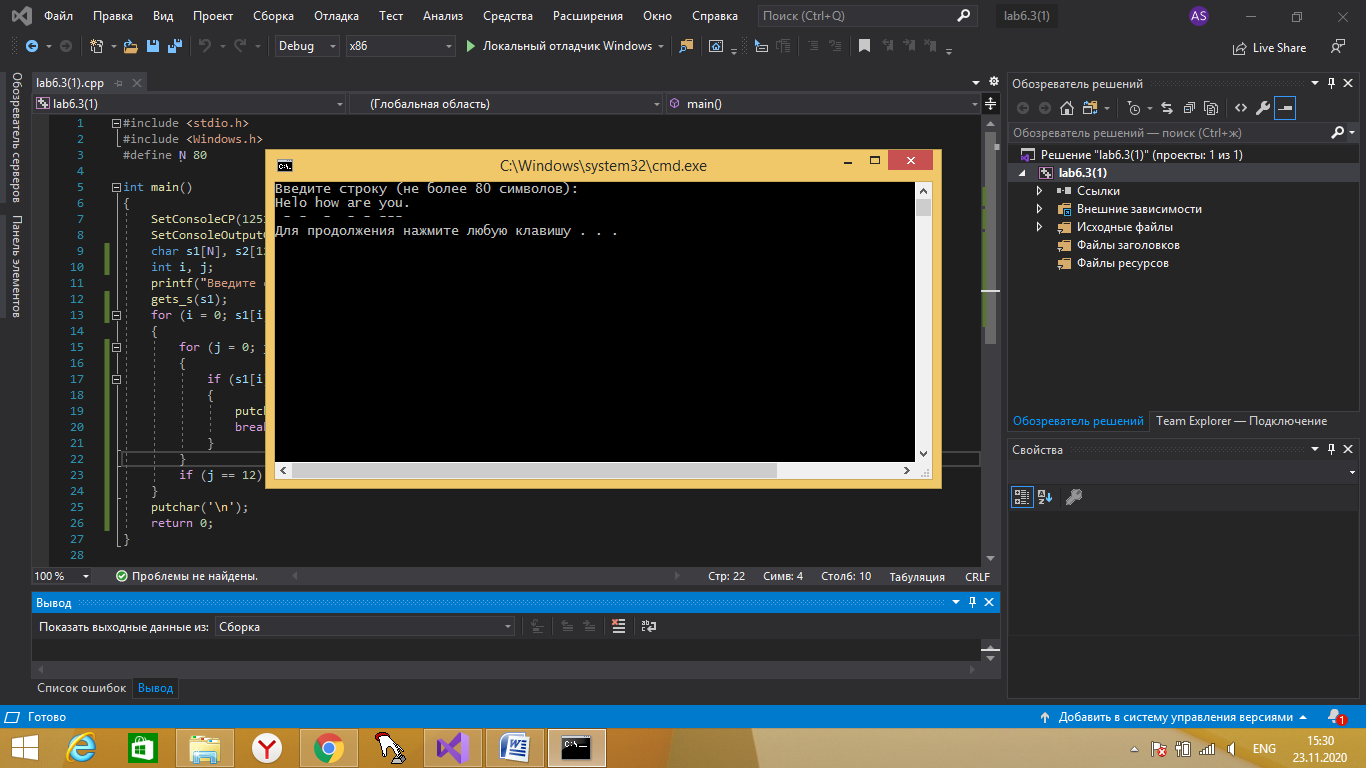
}

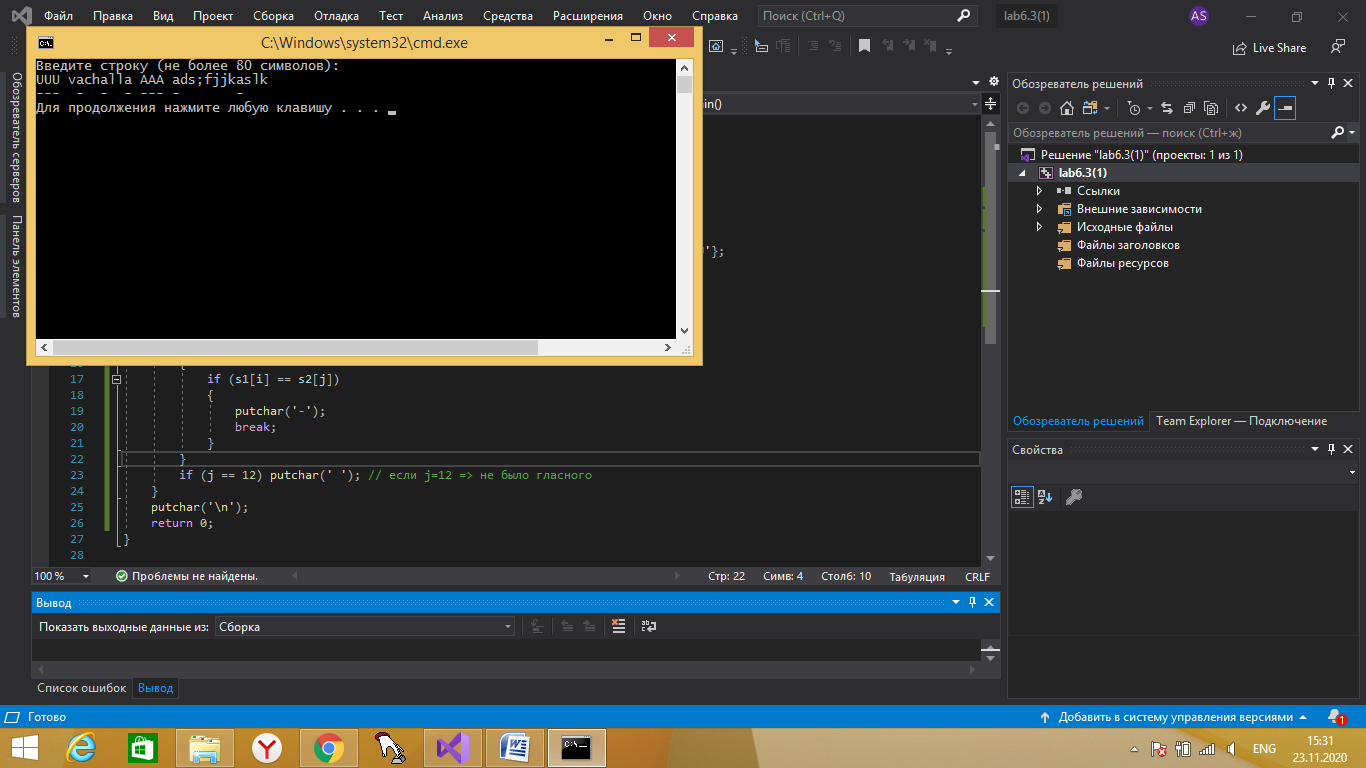
putchar('\n');

return 0;

}

Результаты программы:





**Задание 4**

1. Дана строка не более 80 символов (без кириллицы), оканчивающаяся точкой. Определить:

Д) символ, встречающийся в тексте с максимальной частотой.

Код программы:

#include <stdio.h>

#include <Windows.h>

#define N 81

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char s[N];

int i, j, o = 1, jmax = 0, m[127]; //числа массива соответсвуют номеру символа ASCII

puts("Введите строку (не более 80 символов):");

gets\_s(s);

while (s[0] == '\0') //'\n' - не считается за символ

{

puts("Введите строку");

gets\_s(s);

}

for (i = 0; s[i] != '\0'; ++i) //высчитывается количество каждого символа

{

j = (s[i]);

++m[j];

}

for (j = 0; j < 127; ++j)

{

if (m[j] == m[jmax]) ++o;

if (m[j] > m[jmax])

{

jmax = j;

o = 1;

}

}

if (o != 1) printf("%d - одинаково максимально встречающихся символа, т.е нет самого частого\n", o);

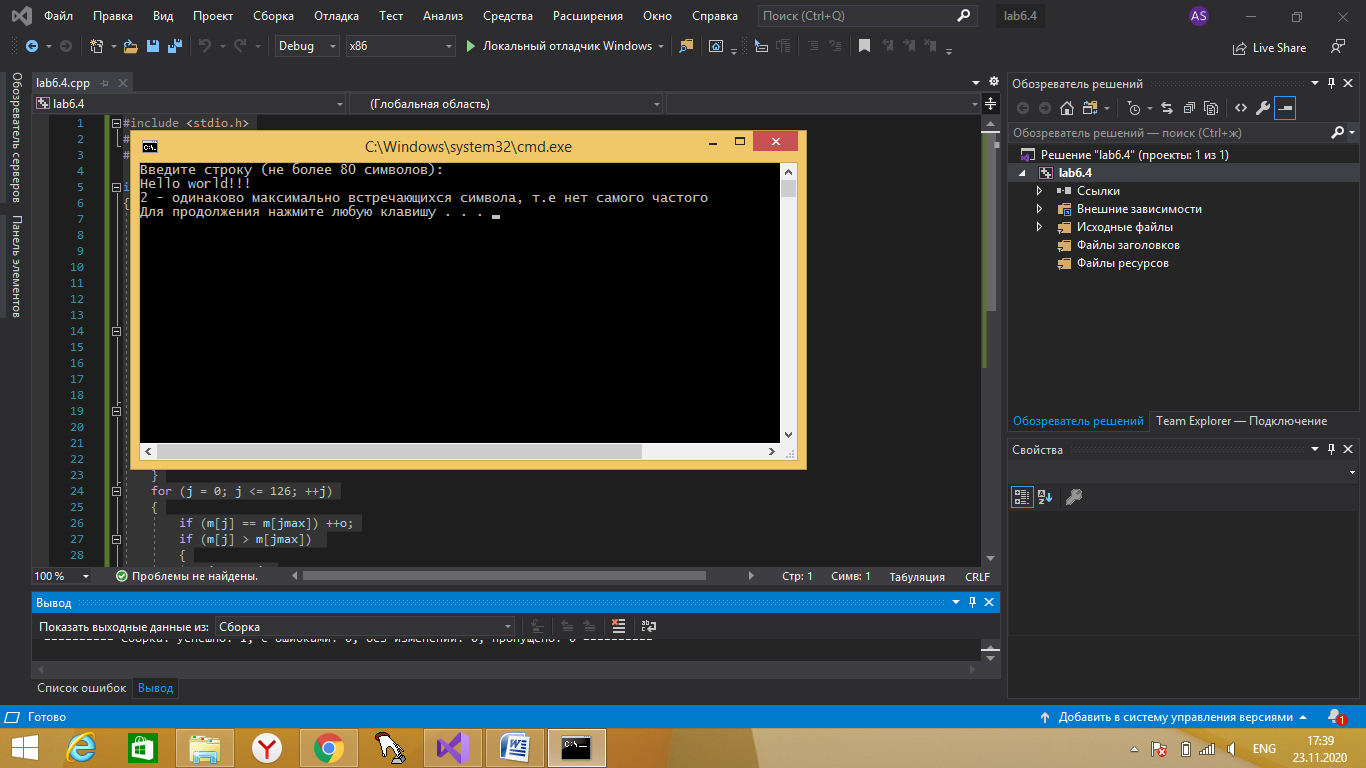
else printf("Символ, встречающийся с максимальной частотой: %c\n", jmax);

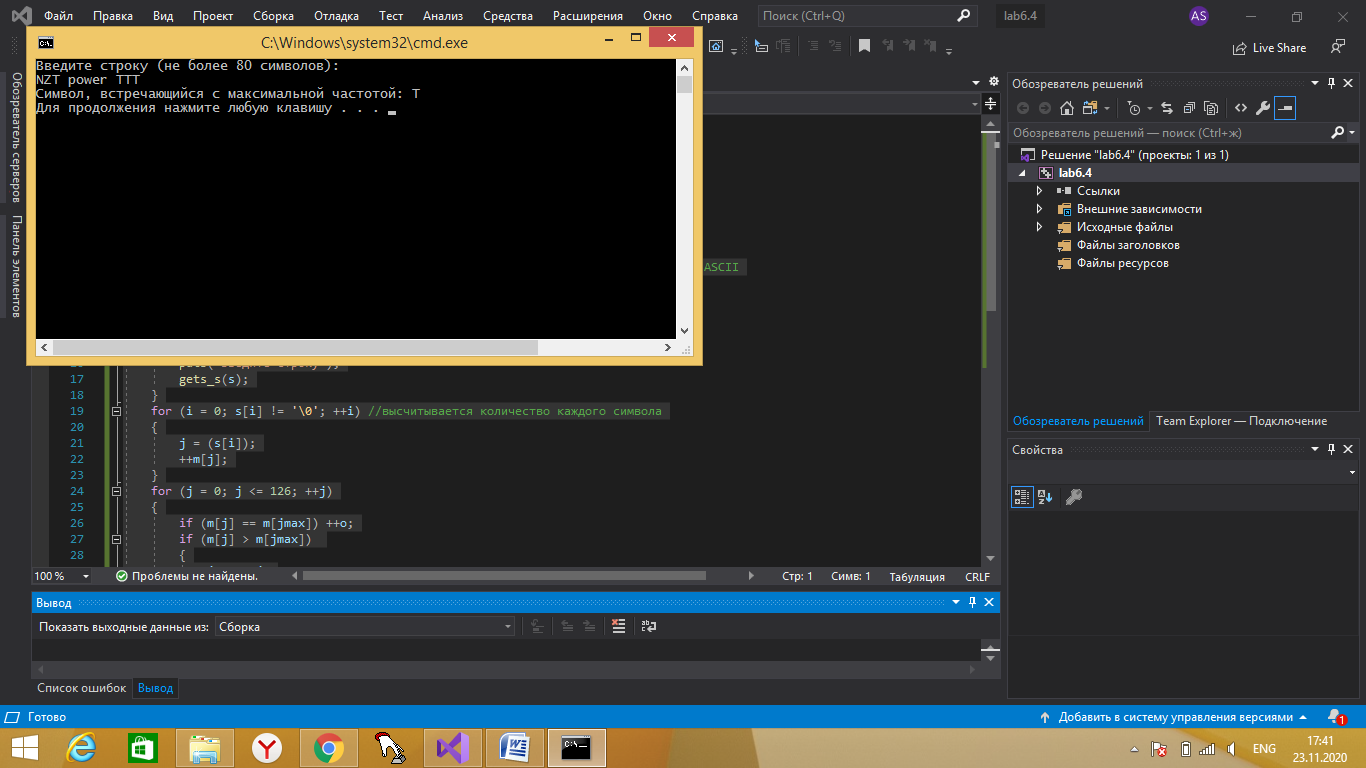
if (jmax == 32) puts("пробел");

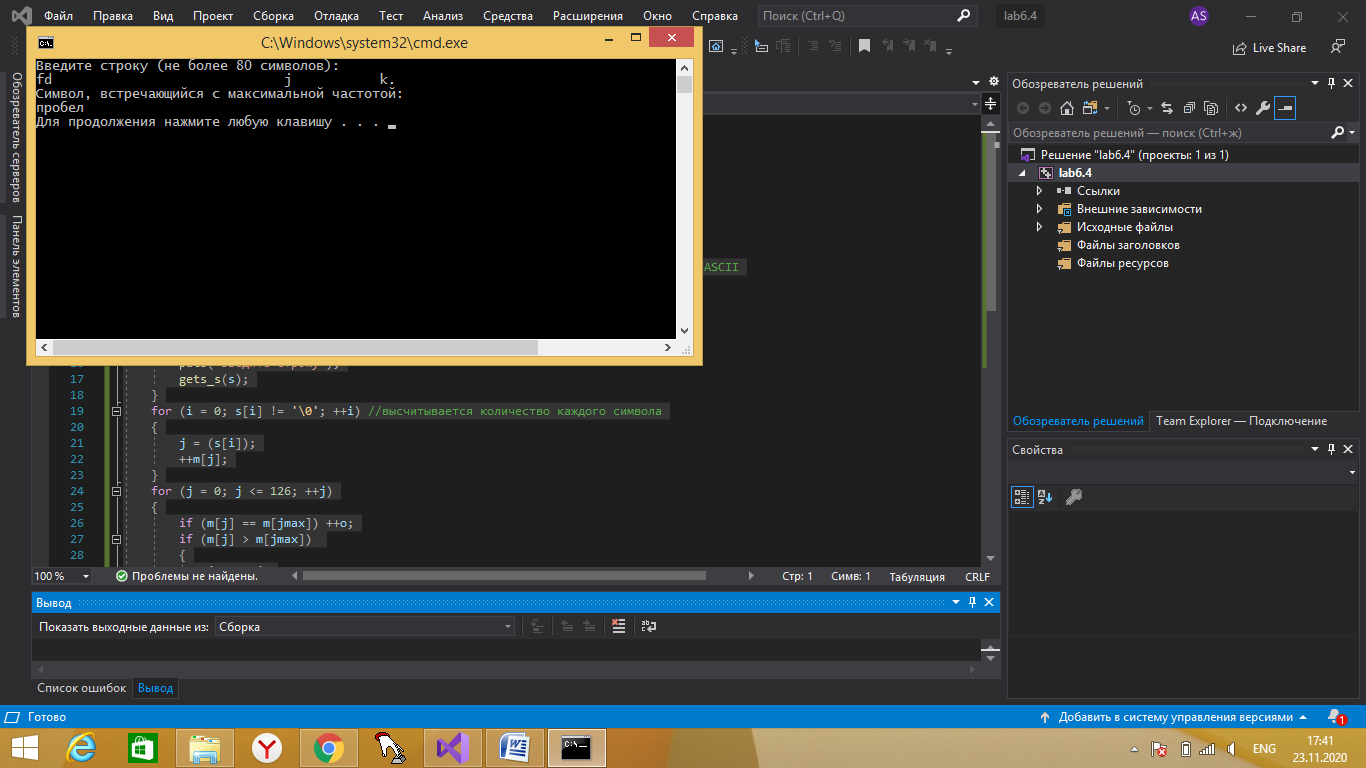
return 0;

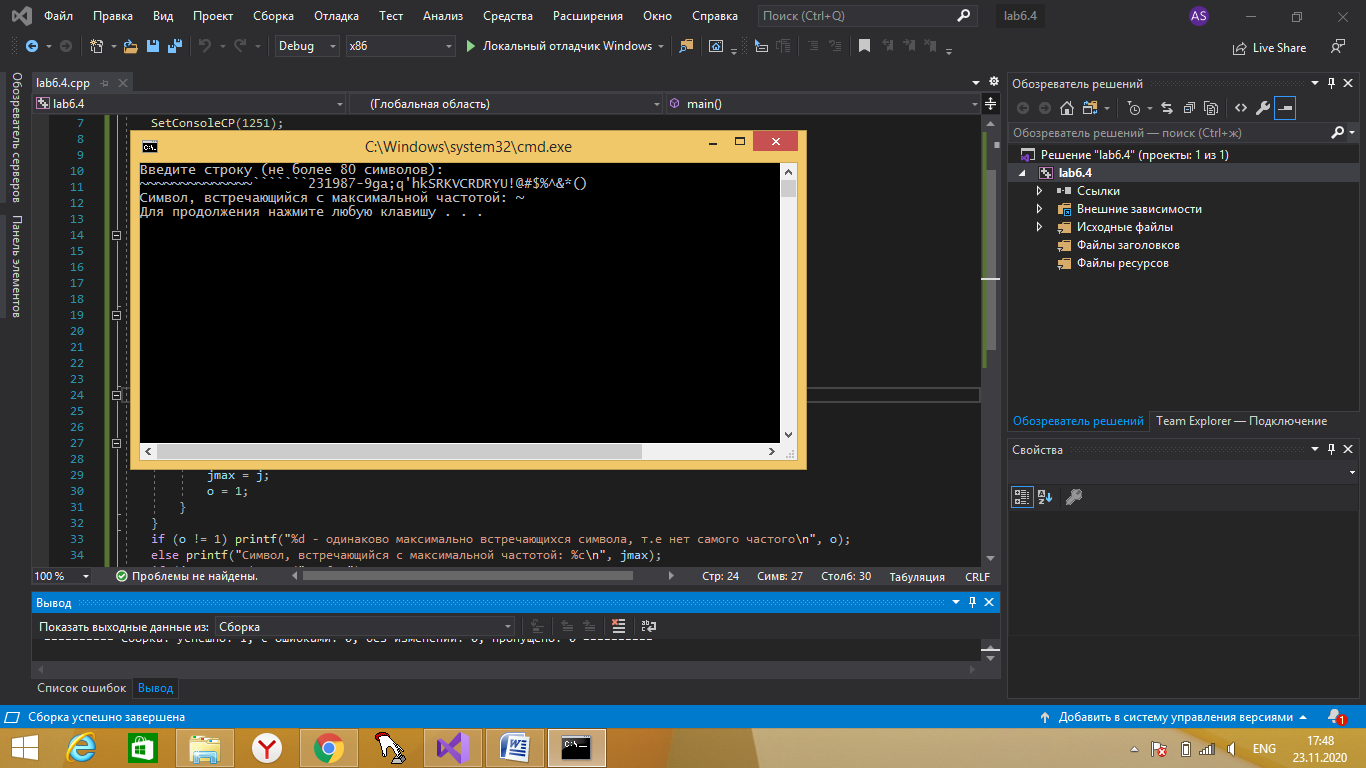
}

Результаты программы:









**Задание 5**

1. Дана целочисленная матрица из 4-х строк и 5-ти столбцов. Определить:

Г) количество отрицательных элементов в каждом столбце матрицы

Код программы:

#include <stdio.h>

#include <locale.h>

int main()

{

setlocale(LC\_ALL, "Rus");

int a[4][5], i, j, n = 0;

puts("Введите матрицу 4\*5:");

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

{

for (j = 0; j < 5; ++j)

{

scanf\_s("%d", &a[i][j]);

}

}

puts("Количество отрицательных чисел в соответствующих столбцах:");

for (j = 0; j < 5; ++j)

{

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

{

if (a[i][j] < 0) ++n;

}

printf(" %d ", n);

n = 0;

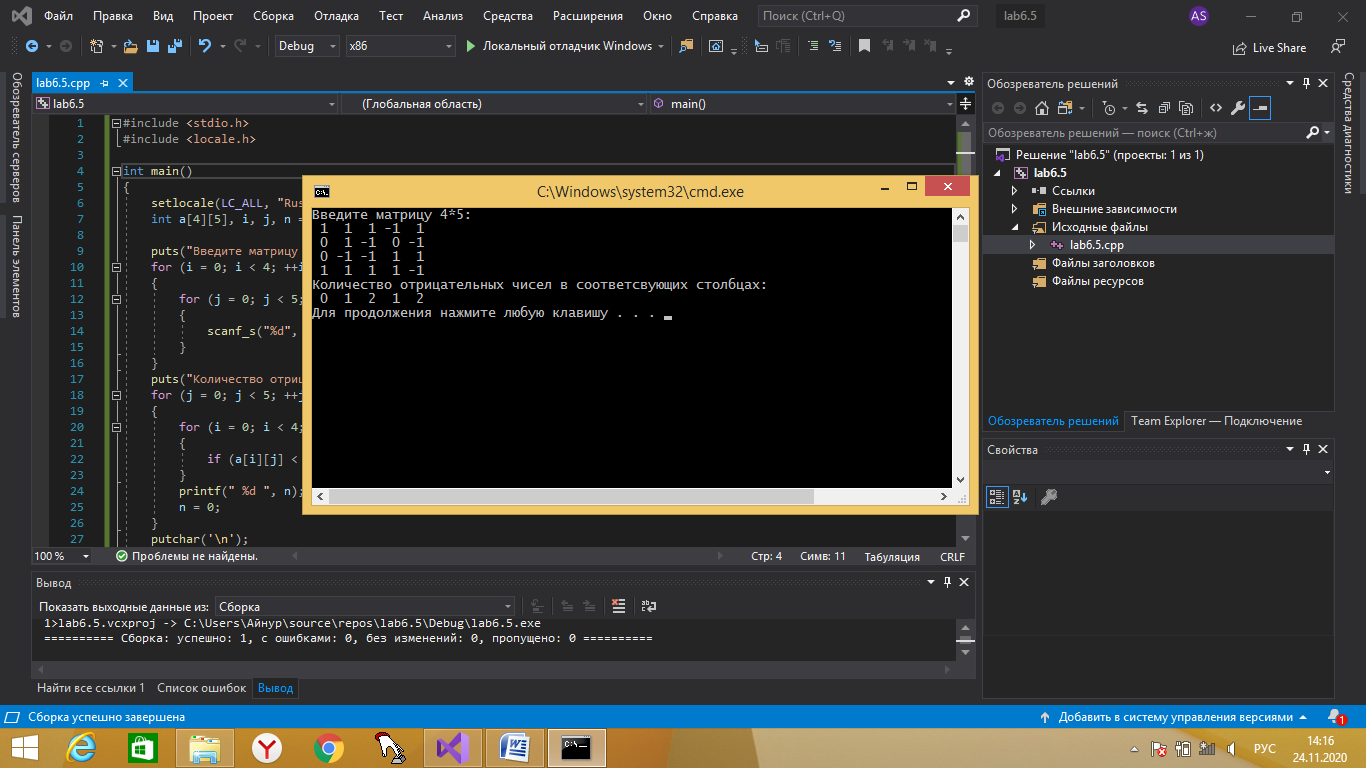
}

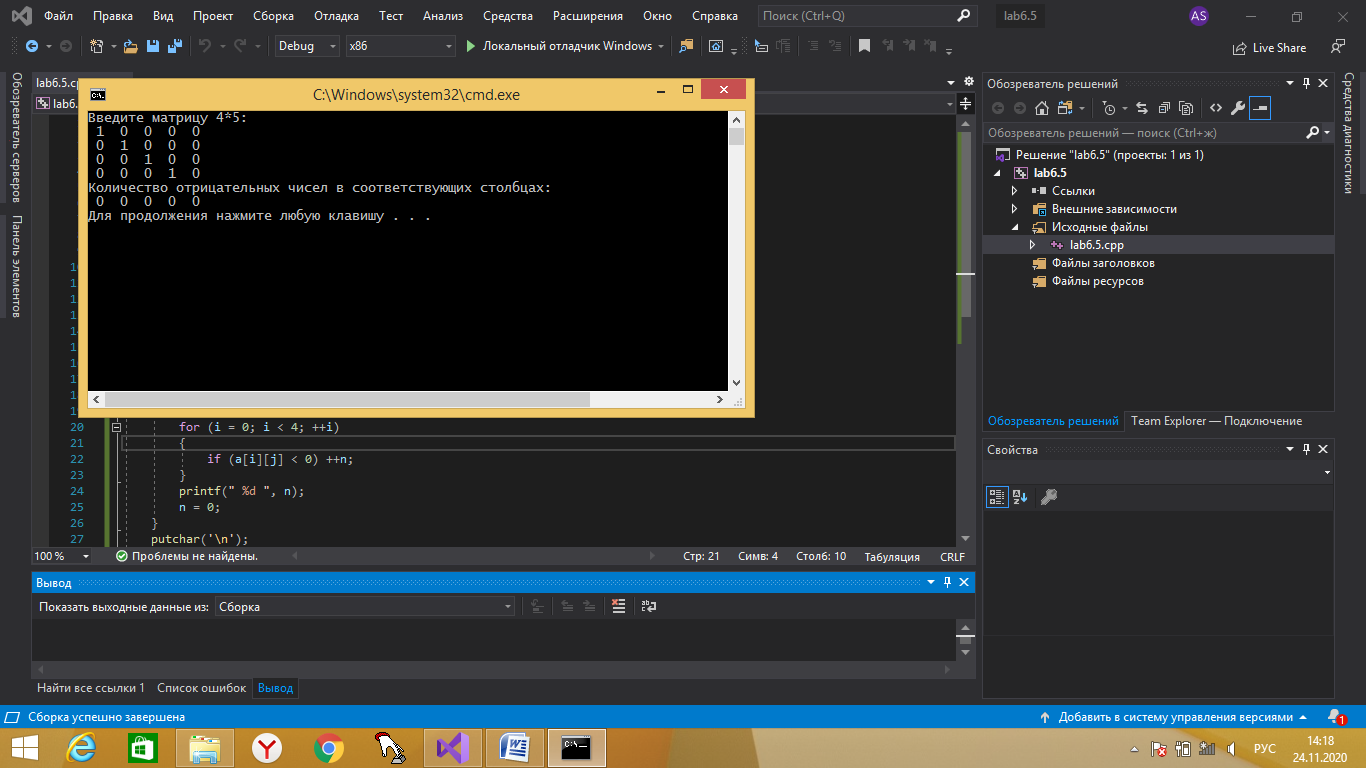
putchar('\n');

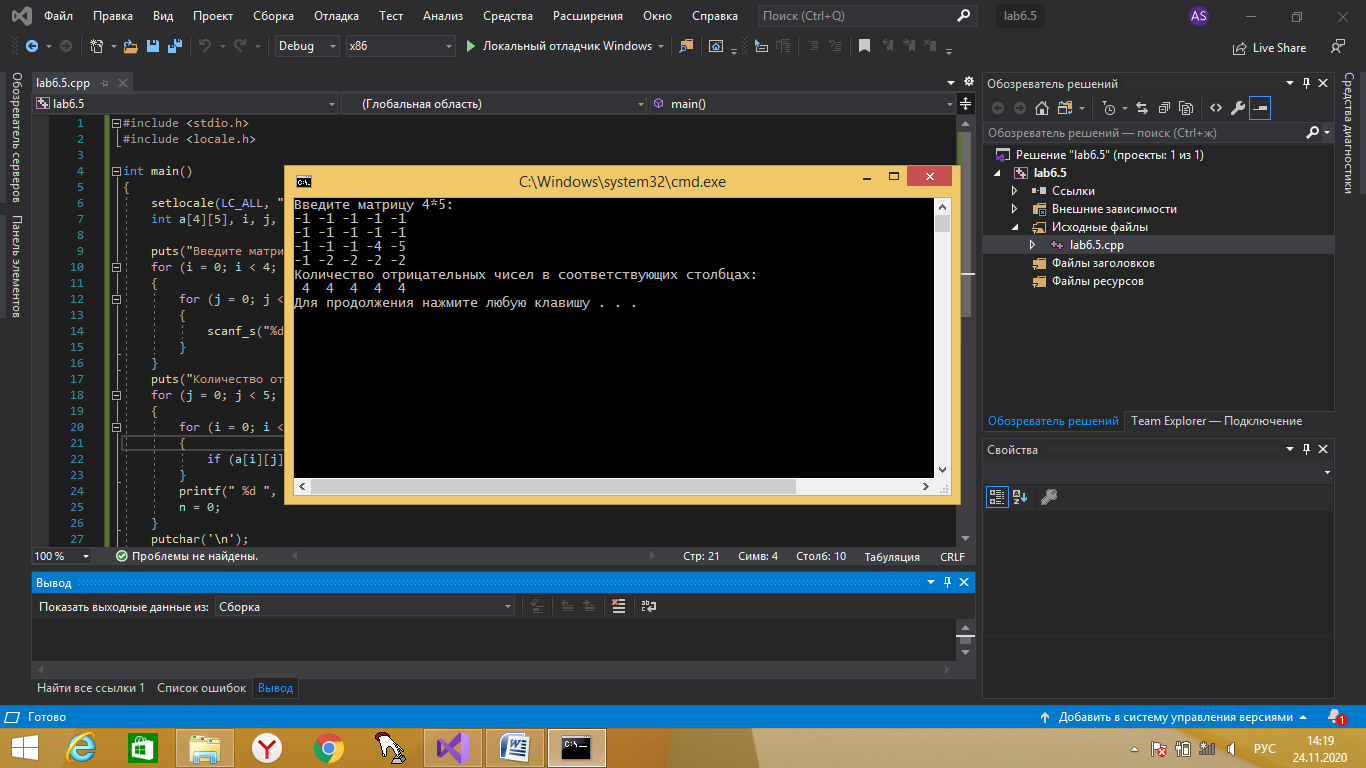
return 0;

}

Результаты программы:







**Задание 6**

1. Дана целочисленная квадратная матрица размером n\*n, где n – заданное число (n < 7). Определить:

Е) сумму и наименьший элемент побочной диагонали

Код программы:

#include <stdio.h>

#include <locale.h>

#define N 4

int main()

{

setlocale(LC\_ALL, "Rus");

int a[N][N], i, j, s = 0, min = 2147483647; //макс значение int

printf("Введите матрицу %d\*%d:\n", N,N);

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

{

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

{

scanf\_s("%d", &a[i][j]);

}

}

for (i = N - 1, j = 0; j != N; --i, ++j)

{

s += a[i][j];

if (a[i][j] < min) min = a[i][j];

}

printf("Сумма и наименьший элемент побочной диагонали соответственно - %d и %d\n", s, min);

return 0;

}

Результаты программы:

