**30333-2 Калевич Сергей Михайлович Лабораторная работа №6, вариант 4**

**ЗАДАНИЕ. (аргументы функций)**

Написать собственные отдельные функции

Организовать вызов каждой функции соответствующим образом из функции main. Проверить работу каждой из них.

*ОБЩИЕ ТРЕБОВАНИЯ К ФУНКЦИЯМ:*

1. Функция 1 без аргументов. Выполняет вычисления по индивидуальному заданию и выводит полученный результат на экран.
2. Функция 2 принимает аргументы (**по значению**),

вычисляет значение выражения и возвращает результат.

1. Функция 3 получает аргументы,

вычисляет значение выражения и значение первого аргумента заменяет на полученный результат (**по ссылке**).

1. Функция 4 получает в качестве аргументов – одномерный статический массив и его размер,

возвращает– **указатель** на искомый по заданию элемент (или NULL, если элемент не определен).

1. Функция 5 получает в качестве аргумента – указатель на квадратный двухмерный **динамический** массив,

возвращает – **указатель** на одномерный массив (тоже **динамический**), который построен по указанному в варианте правилу

|  |  |  |
| --- | --- | --- |
| **4** | 1 | Вычислить сумму всех нечетных чисел в диапазоне от 24 до 516 |
| 2 | Проверить, имеют ли два аргумента разные знаки |
| 3 | Найти максимум из 3-х чисел |
| 4 | Минимальный элемент массива |
| 5 | Одномерный массив — это минимумы строк двухмерного массива |

**Ход выполнения:**

// Lab #6

// variant #4

#include <iostream>

#include <Windows.h>

using namespace std;

void noArguments();

bool byValue(int, int);

void byReference(int&, int, int);

int\* byAddressStaticArray(int[], int);

int\* byAddressDynamicArray(int\*\*, int);

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

cout << "Функция 1 без аргументов" << endl;

noArguments();

cout << "\nФункция 2 принимает аргументы (по значению)" << endl;

cout << "Введите два числа" << endl;

int a{};

int b{};

int c{};

cin >> a >> b;

cout << "Проверить, имеют ли два аргумента разные знаки: "

<< (byValue(a, b) ? "Да" : "Нет") << endl;

cout << "\nФункция 3 получает аргументы, "

<< "вычисляет значение выражения "

<< "и значение первого аргумента заменяет на полученный результат(по ссылке)."

<< endl;

cout << "Введите три числа" << endl;

cin >> a >> b >> c;

cout << "Начальное значение первого аргумента: " << a << endl;

byReference(a, b, c);

cout << "Максимум из 3-х чисел: " << a << endl;

cout << "Конечное значение первого аргумента: " << a << endl;

cout << "\nФункция 4 получает в качестве аргументов - "

<< "одномерный статический массив и его размер, "

<< "возвращает – указатель на искомый по заданию элемент "

<< "(или NULL, если элемент не определен)."

<< endl;

int staticArray[] = {3, 5, 1, 0, 3, 13, -1, 17};

int staticArraySize = sizeof staticArray / sizeof staticArray[0];

for (int i = 0; i < staticArraySize; i++) cout << staticArray[i] << " | ";

cout << endl;

int\* pMin = byAddressStaticArray(staticArray, staticArraySize);

if (pMin) cout << "Минимальный элемент: " << \*pMin << " по адресу: " << pMin << endl;

else cout << "Минимальный элемент не найден." << endl;

cout << "\nФункция 5 получает в качестве аргумента – "

<< "указатель на квадратный двухмерный динамический массив, "

<< "возвращает – указатель на одномерный массив (тоже динамический), "

<< "который построен по указанному в варианте правилу."

<< endl;

const int dynamicArraySize{10};

srand(time(0));

int min = -10;

int max = 50;

int\*\* dynamicArray = new int\* [dynamicArraySize];

for (int i = 0; i < dynamicArraySize; i++)

{

dynamicArray[i] = new int[dynamicArraySize];

for (int j = 0; j < dynamicArraySize; j++)

{

dynamicArray[i][j] = min + rand() % (max - min + 1);

}

}

for (int i = 0; i < dynamicArraySize; i++)

{

for (int j = 0; j < dynamicArraySize; j++)

{

printf("%5d", dynamicArray[i][j]);

}

cout << endl;

}

cout << endl;

cout << "Одномерный массив — это минимумы строк двухмерного массива:" << endl;

int\* result = byAddressDynamicArray(dynamicArray, dynamicArraySize);

for (int i = 0; i < dynamicArraySize; i++)

{

cout << result[i] << ' ';

}

cout << endl;

delete[] dynamicArray;

delete[] result;

system("pause");

return 0;

}

void noArguments()

{

int sum{};

for (int i = 24; i < 516; i++)

{

if (i % 2)

{

sum += i;

}

}

cout << "Сумма всех нечетных чисел в диапазоне от 24 до 516: "

<< sum << endl;

}

bool byValue(int a, int b) {

/\*

\* Предложенный способ: return a \* b < 0; не работает,

\* если один из аргументов равен 0.

\*

\* Например:

\* Функция 2 принимает аргументы(по значению)

\* Введите два числа

\* -1 0

\* Проверить, имеют ли два аргумента разные знаки : Нет

\*/

return (a < 0 || b < 0) && !(a < 0 && b < 0);

}

void byReference(int& a, int b, int c)

{

int max{ a };

if (b > max)

{

max = b;

}

if (c > max)

{

max = c;

}

a = max;

}

int\* byAddressStaticArray(int \*array, int arraySize)

{

int \*pMin = &array[0];

for (int i = 0; i < arraySize; i++)

{

if (array[i] < \*pMin) pMin = &array[i];

}

return pMin ? pMin : NULL;

}

int\* byAddressDynamicArray(int \*\*array, int arraySize)

{

// Одномерный массив — это минимумы строк двухмерного массива

int \*result = new int[arraySize];

for (int i = 0; i < arraySize; i++)

{

result[i] = array[i][0];

for (int j = 0; j < arraySize; j++)

{

if (array[i][j] < result[i])

{

result[i] = array[i][j];

}

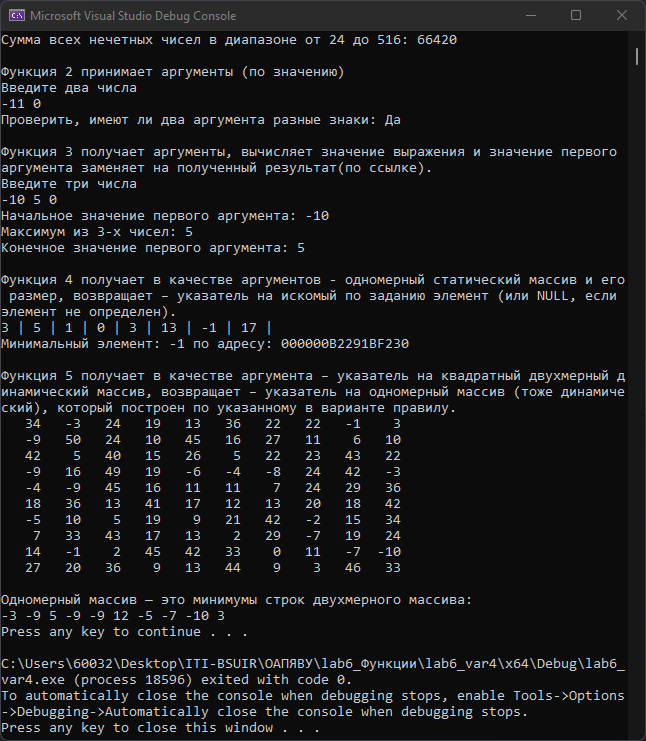
}

}

return result;

}

**Результат:**



Выводы

Изучены способы создания собственных функций, методы передачи аргументов в функции (их преимущества и недостатки), возможности возврата различных типов значений из функций.