

Учреждение образования
“Белорусский государственный университет информатики и
радиоэлектроники”

ОТЧЕТ
«Функции»

Выполнил:
Студент ФКП
Гр.114302
Верховодко Н.В.

Проверил:
Кабариха В. А.

Минск 2021

У меня было задние: найти количество встреч числа в массиве.

Для решения этой задачи я разбил ее на маленькие подзадачи: создать массив, создать строку путем перевода элементов массива в строку и найти повторяющиеся элементы. Для каждой задачи я создал отдельную функцию.

```
void numberOfMeetingsNumbers(int number, string line)
{
    int counterOfRecElements = 0;
    string numberStr = toString(number), buildingNumber;
    for (int i = 0; i < line.size(); i++)
    {
        if (line[i] != ' ')
            buildingNumber += line[i];
        else
        {
            if (buildingNumber == numberStr)
                counterOfRecElements++;
            buildingNumber = "";
        }
    }
    if (counterOfRecElements == 0)
    {
        cout << "\nThere is no " << numberStr << "\n\n";
    }
    else
        cout << endl << "Number of recurring " << numberStr << " = " << counterOfRecElements << endl;
}
```

```
int* buildArray(int& amount)
{
    int code;
    do {
        cout << "Введите количество элементов массива: ";
        amount = inputNumber();
        if (amount < 1) cout << "Размер массива не может быть отрицательным или равным нулю!\n";
        if (amount > 9999) cout << "Размер массива не может быть больше 9999!\n";
    } while (amount < 1 || amount > 9999);
    int* array = new int[amount];
    do {
        cout << "Выберите способ задания массива:\n" <<
            "-1- ввод с клавиатуры\n" <<
            "-2- задание случайными числами на выбранном отрезке\n";
        cout << "Введите код вашего выбора: ";
        code = inputNumber();
        switch (code)
        {
            case 1: byKeyboard(amount, array); break;
            case 2: byRandom(amount, array); break;
            default: cout << "Вы можете использовать только '1' и '2'\n";
        }
    } while (code < 1 || code > 2);
    return array;
}
```

```
void byKeyboard(int amount, int* array)
{
    for (int i = 0; i < amount; ++i)
    {
        cout << "Введите элемент массива: ";
        array[i] = inputNumber();
    }
}
```

```
void showArray(int amount, int* array)
{
    for (int i = 0; i < amount; ++i)
    {
        cout << array[i] << " ";
    }
    cout << endl;
}
```

```
string buildString(int* array, int amount)
{
    string line;
    for (int i = 0; i < amount; i++)
    {
        line += toString(array[i]);
        line += " ";
    }
    return line;
}
```

```
int inputNumber()
{
    int number;
    while (true)
    {
        cin >> number;
        if (isNumberNumeric())
        {
            return number;
        }
        else
        {
            cout << "Incorrect input!\n" << "Try again: ";
        }
    }
}
```

```
void byRandom(int amount, int* array)
{
    int left_value, right_value;
    cout << "Введите начало отрезка: ";
    left_value = inputNumber();
    do {
        cout << "Введите конец отрезка: ";
        right_value = inputNumber();
        if (left_value >= right_value) cout << "Конец отрезка не может быть меньше начала или равен ему!\n";
    } while (left_value >= right_value || right_value < 0);

    for (int i = 0; i < amount; ++i)
    {
        int z = rand() % 3;
        if (z == 1)
            z = -1;
        else
            z = 1;
        array[i] = (rand() % (right_value - left_value + 1) + left_value) * z;
    }
}
```

И вызвал все эти функции в `numberOfRecurringByString()`

```
void numberOfRecurringByString()
{
    cout << "Enter a number: ";
    int number = inputNumber();
    int amount, counterRec;
    int* array = buildArray(amount);
    string line = buildString(array, amount);
    cout << endl << line;
    numberOfMeetingsNumbers(number, line);
}
```

Если бы я не поделил задачу на функции мой код выглядел бы так:

```
void numberOfRecurringByString()
{
    cout << "Enter a number: ";
    int number;
    while (true)
    {
        cin >> number;
        if (if (cin.get() == '\n')
            return true;
        else
        {
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            return false;
        }
        return number;
    }
    else
    {
        cout << "Incorrect input!\n" << "Try again: ";
    }
}

int amount, counterRec;
int* array = buildArray(amount);
int code;
do {
    cout << "Введите количество элементов массива: ";
    amount = inputNumber();
    if (amount < 1) cout << "Размер массива не может быть отрицательным или равным нулю!\n";
    if (amount > 9999) cout << "Размер массива не может быть больше 9999!\n";
} while (amount < 1 || amount > 9999);
int* array = new int[amount];
do {
    cout << "Выберите способ задания массива:\n" <<
        "-1- ввод с клавиатуры\n" <<
        "-2- задание случайными числами на выбранном отрезке\n";
    cout << "Введите код нашего выбора: ";
    code = inputNumber();
    switch (code)
    {
        case 1: for (int i = 0; i < amount; ++i)
        {
            cout << "Введите элемент массива: ";
            array[i] = inputNumber();
        }; break;
        case 2: int left_value, right_value;
            cout << "Введите начало отрезка: ";
            left_value = inputNumber();
            do {
                cout << "Введите конец отрезка: ";
                right_value = inputNumber();
                if (left_value >= right_value) cout << "Конец отрезка не может быть меньше начала или равен\n";
            } while (left_value >= right_value || right_value < 0);
            for (int i = 0; i < amount; ++i)
            {
                int x = rand() % 3;
                if (x == 1)
                    x = -1;
                else
                    x = 1;
                array[i] = (rand() % (right_value - left_value + 1) + left_value) * x;
            }; break;
        default: cout << "Вы можете использовать только '1' и '2'\n";
    }
} while (code < 1 || code > 2);
return array;
cout << endl << line;
int counterOfReclements = 0;
string numberStr = toString(number, buildingNumber);
for (int i = 0; i < line.size(); i++)
{
    if (line[i] != ' ')
        buildingNumber += line[i];
    else
    {
        if (buildingNumber == numberStr)
            counterOfReclements++;
        buildingNumber = "";
    }
}
if (counterOfReclements == 0)
{
    cout << "\nThere is no " << numberStr << "\n\n";
}
else
    cout << endl << "Number of recurring " << numberStr << " = " << counterOfReclements << endl;
```

Разобраться в такой программе было бы довольно сложно, не говоря уже про поиски ошибок.

Плюсы использования функций

Возможность повторного использования

Читабельность кода

Простота отладки

