

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №6**  
**по дисциплине «Организация ЭВМ и систем»**  
**Тема: Организация связи Ассемблера с ЯВУ на примере**  
**программы построения частотного**  
**распределение попаданий псевдослучайных целых чисел в**  
**заданные интервалы.**

Студент гр. 1303

Самохин К.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

### **Цель работы.**

Разработать программу, генерирующую набор псевдослучайных чисел и подсчитывающую, сколько из этих чисел попадают в заданные интервалы.

### **Задание.**

На языке высокого уровня (Pascal или C) генерируется массив псевдослучайных целых чисел, изменяющихся в заданном диапазоне и имеющих равномерное распределение.

Далее должен вызываться ассемблерный модуль(модули) для формирования распределения количества попаданий псевдослучайных целых чисел в заданные интервалы. В общем случае интервалы разбиения диапазона изменения псевдослучайных чисел могут иметь различную длину.

Результирующий массив частотного распределения чисел по интервалам, сформированный на ассемблерном уровне, возвращается в программу, реализованную на ЯВУ, и затем сохраняется в файле и выводится на экран средствами ЯВУ.

Исходные данные:

1. Длина массива псевдослучайных целых чисел - NumRanDat ( $\leq 16K$ ,  $K=1024$ );
2. Диапазон изменения массива псевдослучайных целых чисел  $[X_{\min}, X_{\max}]$ , значения могут быть биполярные;
3. Количество интервалов, на которые разбивается диапазон изменения массива псевдослучайных целых чисел - NInt ( $\leq 24$ );
4. Массив левых границ интервалов разбиения LGrInt (должны принадлежать интервалу  $[X_{\min}, X_{\max}]$ ).

Результаты:

1. Текстовый файл, строка которого содержит:
  - номер интервала,
  - левую границу интервала,

- количество псевдослучайных чисел, попавших в интервал.

Количество строк равно числу интервалов разбиения.

2. График, отражающий распределение чисел по интервалам.

(необязательный результат)

Подпрограмма формирования распределения количества попаданий псевдослучайных целых чисел в заданные интервалы реализуется в виде одного ассемблерного модуля, сразу формирующего требуемое распределение и возвращающего его в главную программу, написанную на ЯВУ.

### **Выполнение работы.**

Считывание исходных данных реализована на ЯВУ C++. Сначала считывается размер массива чисел, потом его границы, результат генерации псевдослучайных чисел записывается в массив `p_arr`. После этого проводится считывание количества интервалов и их левые границы, которые записываются в массив `int_arr`.

В ассемблерный модуль в процедуру `function` передаются указатель на массив чисел `Digits`, размер этого массива `len`, указатель на массив левых границ `LBInt`, размер этого массива `NInt` и указатель на результирующий массив `result`, все элементы которого изначально равны 0. В теле процедуры для каждого элемента массива `Digits` находится интервал, которому элемент принадлежит, после чего счётчик, находящийся по индексу найденного интервала в массиве `result`, увеличивается. Если же число не принадлежит ни одному из интервалов, оно пропускается.

Результат работы выводится в консоль и файл `out.txt`.

Исходный код программы см. в приложении А.

## Тестирование.

Работа программы с заданными условиями представлена на Рисунке 1.

```
Enter random number array length
6
Enter min value
1
Enter max value
8
Enter number of intervals
4
Enter array of left borders
Left border1 = 2
Left border2 = 4
Left border3 = 6
Left border4 = 8
Generated numbers:
8 6 3 8 8 1

Results:
1. left border:2 amount of numbers - 1
2. left border:4 amount of numbers - 0
3. left border:6 amount of numbers - 1
4. left border:8 amount of numbers - 3
```

Рисунок 1.

## Выводы.

В ходе лабораторной работы разработана программа, совмещающая в себе модули на языке Ассемблера и ЯВУ C++. Программа генерирует псевдослучайные числа и фиксирует их попадание в заданные интервалы.

## ПРИЛОЖЕНИЕ А

Название файла: main.cpp

```
#include <cstdio>
#include <cstdlib>
#include <ctime>
#include <iostream>
#include <fstream>

std::ofstream file("out.txt");

extern "C" {void function(int* Digits, int len, int* LBInt, int
NInt, int* result); }

int main() {
    srand(static_cast<unsigned int>(time(nullptr)));
    int NumLen = 0;
    int MinX = 0;
    int MaxX = 0;
    int NInt = 0;
    std::cout << "Enter random number array length\n";
    std::cin >> NumLen;
    if (NumLen <= 0 || NumLen > 16 * 1024) {
        std::cout << "Invalid number array length\n";
        return 1;
    }
    std::cout << "Enter min value\n";
    std::cin >> MinX;
    std::cout << "Enter max value\n";
    std::cin >> MaxX;
    if (MinX >= MaxX) {
        std::cout << "Invalid min and max values\n";
        return 1;
    }
    std::cout << "Enter number of intervals\n";
    std::cin >> NInt;
    if (NInt <= 0 || NInt > 24) {
        std::cout << "Invalid number of intervals\n";
        return 1;
    }
    int* n_arr = new int[NumLen];
    int* int_arr = new int[NInt];
    std::cout << "Enter array of left borders\n";
    for (int i = 0; i < NInt; ++i) {
        std::cout << "Left border" << i + 1 << " = ";
```

```

        std::cin >> int_arr[i];
        if ((int_arr[i] < MinX || int_arr[i] > MaxX) ||
            (i > 0 && int_arr[i] <= int_arr[i - 1])) {
            printf("Invalid left border!\n");
            delete[] n_arr;
            delete[] int_arr;
            return 1;
        }
    }
    int range = MaxX - MinX + 1;
    for (int i = 0; i < NumLen; ++i) {
        n_arr[i] = MinX + rand() % range;
    }
    int* res_arr = new int[NInt] {0};
    function(n_arr, NumLen, int_arr, NInt, res_arr);
    std::cout << "Generated numbers:\n";
    file << "Generated numbers:\n";
    for (int i = 0; i < NumLen; ++i) {
        std::cout << n_arr[i] << " ";
        file << n_arr[i] << " ";
    }
    std::cout << "\n\nResults:\n";
    file << "\n\nResults:\n";
    for (int i = 0; i < NInt; ++i) {
        std::cout << i + 1 << ". left border:" << int_arr[i] <<
" amount of numbers - " << res_arr[i] << "\n";
        file << i + 1 << ". left border:" << int_arr[i] << "
amount of numbers - " << res_arr[i] << "\n";
    }
    file.close();
    delete[] n_arr;
    delete[] int_arr;
    delete[] res_arr;

    return 0;
};

```

**Название файла: lb6.asm**

```

.586p
.MODEL FLAT, C
.CODE
function PROC C USES EDI ESI, Digits:dword, len:dword,
LBInt:dword, NInt:dword, result:dword

```

```

    push eax
    push ebx
    push ecx
    push edi
    push esi

    mov ecx, len
    mov esi, Digits
    mov edi, LBIInt
    mov eax, 0

cycle:
    mov ebx, 0
iter:
    cmp ebx, NInt
    jge out_iter
    push eax
    mov eax, [esi + 4 * eax]
    cmp eax, [edi + 4 * ebx]
    pop eax
    jl out_iter
    inc ebx
    jmp iter

out_iter:
    dec ebx
    cmp ebx, -1
    je next_num
    mov edi, result
    push eax
    mov eax, [edi + 4 * ebx]
    inc eax
    mov [edi + 4 * ebx], eax
    pop eax
    mov edi, LBIInt

next_num:
    inc eax

loop cycle

pop esi
pop edi
pop ecx
pop ebx
pop eax

```

```
ret
```

```
function ENDP  
    END
```