

**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 распределения попаданий**  
**псевдослучайных целых чисел в заданные интервалы.**

Студентка гр. 0383

Александрович В.П.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

### **Цель работы.**

Реализовать программу построения частотного распределения попаданий псевдослучайных целых чисел в заданные интервалы, связав модуль на Ассемблере с файлом на ЯВУ.

### **Задание.**

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

Необходимые датчики псевдослучайных чисел находятся в каталоге Tasks\RAND\_GEN (при его отсутствии программу датчика получить у преподавателя).

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)

В зависимости от номера бригады формирование частотного распределения должно производиться по одному из двух вариантов.

### **Вариант 1**

Один ассемблерный модуль.

### **Выполнение работы.**

На языке C++ реализовано считывание начальных данных (количество чисел, минимальное и максимальное значения, количество интервалов и их левые границы). Левые границы заносятся в массив `left_borders`, а генерируемые числа добавляются в массив `arr`. Отдельно создается массив, который будет хранить результат работы.

В ассемблерный модуль в процедуру `FUNC` передаются указатель на массив сгенерированных чисел, его размер, указатель на массив левых границ интервалов и его размер, указатель на массив, хранящий результат работы. В процедуре для каждого числа находится интервал, в который он попадает, и результат записывается в результирующий массив.

После этого результат работы выводится в консоль и записывается в файл `out.txt`.

Разработанный программный код см. в приложении А.

### Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

Входные данные	Выходные данные	Комментарий
amount: 25 min: 2 max: 50 borders: 2 7 19	Generated numbers: 45 11 48 32 13 39 31 24 36 38 45 15 15 21 39 42 35 48 14 50 24 25 13 5 24 Interval index   Interval left border   Amount of numbers in interval 1                      2                      1 2                      7                      6 3                      19                     18	ВЕРНО
amount: 15 min: 10 max: 100 borders: 30 55	Generated numbers: 65 14 23 87 55 94 40 38 31 57 66 20 17 81 89 Interval index   Interval left border   Amount of numbers in interval 1                      30                      3 2                      55                      8	ВЕРНО

### Выводы.

В ходе выполнения данной лабораторной работы мы изучили организацию связи модулей на языке ассемблер с ЯВУ. Была реализована программа построения частотного распределения попаданий псевдослучайных целых чисел в заданные интервалы.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.cpp

```
#include <iostream>
#include <fstream>
#include <random>

extern "C" void FUNC(int* array, int array_size, int*
left_borders, int intervals_size, int* result_array);

int main()
{
    std::ofstream file("out.txt");
    int NumRanDat;

    std::cout << "Enter the amount of numbers" << std::endl;
    std::cin >> NumRanDat;

    int Xmin, Xmax;
    std::cout << "Enter the minimum number" << std::endl;
    std::cin >> Xmin;
    std::cout << "Enter the maximum number" << std::endl;
    std::cin >> Xmax;

    if (Xmax < Xmin)
    {
        std::cout << "Error: wrong minimum or maximum number" <<
std::endl;
        return 0;
    }

    int interval_amount;
    std::cout << "Enter the amount of intervals" << std::endl;
    std::cin >> interval_amount;

    if (interval_amount <= 0)
    {
        std::cout << "Error: the amount of intervals should be
positive" << std::endl;
        return 0;
    }

    int* left_borders = new int[interval_amount];
    std::cout << "Enter left borders" << std::endl;
    for (int i = 0; i < interval_amount; i++)
        std::cin >> left_borders[i];

    for (int i = 0; i < interval_amount - 1; i++)
    {
        for (int j = i + 1; j < interval_amount; j++)
        {
            if (left_borders[j] < left_borders[i])
                std::swap(left_borders[j], left_borders[i]);
        }
    }
}
```

```

std::random_device rand;
std::mt19937 gen(rand());
std::uniform_int_distribution<> dis(Xmin, Xmax);
int* arr = new int[NumRanDat];

for (int i = 0; i < NumRanDat; i++)
    arr[i] = dis(gen);

file << "Generated numbers: ";
for (int i = 0; i < NumRanDat; i++)
    file << arr[i] << ' ';
file << '\n';

int* result_array = new int[interval_amount];
for (int i = 0; i < interval_amount; i++)
    result_array[i] = 0;

FUNC(arr,      NumRanDat,      left_borders,      interval_amount,
result_array);

std::cout << "Interval index \tInterval left border \tAmount
of numbers in interval" << '\n';
file << "Interval index \tInterval left border \tAmount of
numbers in interval" << '\n';

for (int i = 0; i < interval_amount; i++)
{
    std::cout << "\t" << i + 1 << "\t\t" << left_borders[i]
<< "\t\t\t" << result_array[i] << '\n';
    file << "\t" << i + 1 << "\t\t" << left_borders[i] << "\t\t\t" << result_array[i] << '\n';
}

delete[] left_borders;
delete[] arr;
delete[] result_array;
}

```

Название файла: module.asm

```

.586
.MODEL FLAT, C
.CODE
FUNC PROC C array:dword, array_size:dword, left_borders:dword,
interval_amount:dword, result_array:dword
push ecx
push esi
push edi
push eax
push ebx

mov ecx, array_size
mov esi, array
mov edi, left_borders
mov eax, 0

```

```

l1:
    mov ebx, 0
    borders:
        cmp ebx, interval_amount
        jge borders_exit
        push eax
        mov eax, [esi+4*eax]
        cmp eax, [edi+4*ebx]
        pop eax
        jl borders_exit
        inc ebx
        jmp borders
    borders_exit:
    dec ebx

    cmp ebx, -1
    je skip
    mov edi, result_array
    push eax
    mov eax, [edi+4*ebx]
    inc eax
    mov [edi+4*ebx], eax
    pop eax
    mov edi, left_borders
    skip:
    inc eax
loop l1

pop ebx
pop eax
pop edi
pop esi
pop ecx
ret
FUNC ENDP
END

```