

**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.**

Студентка гр. 0382

Морева Е.С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

## **Цель работы.**

Написать программу построения частного распределения попаданий псевдослучайных целых чисел в заданные интервалы.

## **Задание.**

Вид распределения: равномерный

Число ассем. процедур: 1

$N_{int} < D_x$

$L_{g1} > X_{min}$

Исходные данные:

1. Длина массива псевдослучайных целых чисел - NumRandat ( $\leq 16K$ )
2. Диапазон изменения массива псевдослучайных целых чисел  
[Xmin, Xmax] (м.б. биполярный, например, [-100, 100])
3. Массив псевдослучайных целых чисел {Xi}.
4. Количество интервалов, на которые разбивается диапазон  
изменения массива псевдослучайных целых чисел - NInt ( $\leq 24$ )
5. Массив левых границ интервалов разбиения LGrInt.

Результаты:

Текстовая таблица, строка которой содержит:

- номер интервала,
- левую границу интервала,
- количество псевдослучайных чисел, попавших в интервал.

Количество строк должно быть равно числу интервалов разбиения.

Таблица должна выводиться на экран и сохраняться в файле.

## **Выполнение работы.**

На языке ЯВУ C++ написана программа принимающая исходные данные (и производящая проверку введенных границ на соответствие требованиям

задания), генерирующая псевдослучайные числа и проводящая все необходимые сортировки. Также в файле с расширением `asm` реализована ассемблеровская программа, она обрабатывает созданный массив псевдослучайных чисел. Для каждого числа из массива `numbers` находится нужный интервал(если текущее число больше левой границы, то берется следующая граница, пока число не будет меньше границы, тогда ее интервал – предыдущая граница - переходим по метке `1b1`, где соответствующий результат увеличивается на единицу.)

## **Вывод.**

Написана программа построения частного распределения попаданий псевдослучайных целых чисел в заданные интервалы.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab6.cpp

```
#include <iostream>

#include <random>
#include <fstream>

using namespace std;

extern "C" void FUNC(int* numbers, int num_cnt, int* lgrint, int n_int, int* result);

int main()
{
    setlocale(0, "");
    int n, x_min, x_max, n_int, Dx;

    cout << "Введите количество чисел, которые необходимо сгенерировать:" << endl;
    cin >> n;

    cout << "Введите Xmin:" << endl;
    cin >> x_min;

    cout << "Введите Xmax:" << endl;
    cin >> x_max;

    if (x_max < x_min)
    {
        cout << "Неверное введенное значение Xmax" << endl;
        return 0;
    }

    Dx = x_max - x_min;

    cout << "Введите количество интервалов: < " << Dx << ":" << endl;
    cin >> n_int;

    int* lGrInt = new int[n_int];

    cout << "Введите первую левую границу LG1 > " << x_min << endl;

    for (int i = 0; i < n_int; i++)
    {
        cin >> lGrInt[i];

        if (lGrInt[i] <= x_min)
        {
            cout << "Введенная граница некорректна" << endl;
            return 0;
        }
    }
}
```

```

int j, buffer;
for (int i = 1; i < n_int; ++i)
{
    buffer = lGrInt[i];
    j = i - 1;
    while (j >= 0 && lGrInt[j] > buffer)
    {
        lGrInt[j + 1] = lGrInt[j];
        lGrInt[j] = buffer;
        --j;
    }
}

lGrInt[n_int] = x_max;
random_device rand;
mt19937 gen(rand());
uniform_int_distribution<> numb(x_min, x_max);
int* numbers = new int[n];

for (int i = 0; i < n; i++)
    numbers[i] = numb(gen);

cout << "Сгенерированные числа:" << endl;
for (int i = 0; i < n; i++)
    cout << numbers[i] << " ";
cout << endl;

int* res = new int[n];

for (int i = 0; i < n_int; i++)
    res[i] = 0;
cout << endl;

FUNC(numbers, n, lGrInt, n_int, res);

ofstream fout("ResOut.txt");

cout << "N  Lg  Res" << endl;
fout << "N  Lg  Res" << endl;

for (int i = 0; i < n_int; i++)
{
    cout << i << "    " << lGrInt[i] << "    " << res[i] << endl;
    fout << i << "    " << lGrInt[i] << "    " << res[i] << endl;
}

fout.close();
return 0;

}

}

```

## Название файла:lb6.asm

```
..586
.MODEL FLAT, C
.CODE
FUNC PROC C numbers:dword, num_count:dword, lGrInt:dword, n_int:dword, res:dword

; помещаем данный в стек для сохранения состояния
push edi
push esi
push eax
push ebx
push ecx

; передаем переданные данные
mov esi, numbers
mov edi, lGrInt
mov ecx, num_count
xor eax, eax

next_rand_number:

    xor ebx, ebx

    next_lg:

        cmp ebx, n_int
        jge lb1 ;X!=Y
        push eax
        mov eax, [esi + 4 * eax]
        cmp eax, [edi + 4 * ebx]
        pop eax
        jl lb1
        inc ebx
        jmp next_lg

lb1:

    dec ebx
    mov edi, res
    push eax
    ; увеличение счётчика в текущем интервале
    mov eax, [edi + 4 * ebx]
    inc eax
    mov [edi + 4 * ebx], eax
```

```
    pop eax
    mov edi, lGrInt
    inc eax
```

```
loop next_rand_number;inc ecx
```

```
pop ecx
pop ebx
pop eax
pop edi
pop esi
ret
```

```
FUNC ENDP
```

```
END
```