

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. 1303

Королева П.А

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Разработать программу, считывающую исходные данные и генерирующая массив случайных чисел через ЯВУ и подсчитывающую количество чисел в интервалах через модуль ассемблера.

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.

Выполнение работы.

В модуле ЯВУ в начале программы объявляется функция

```
extern "C" {void function (int* Array, int len, int* LGrInt, int NInt, int* answer); }
```

Таким образом компилятор будет понимать, что function определена в другом модуле и он связывается с основной программой согласно порядку для C.

Далее считываются исходные данные – кол-во чисел в массиве, который надо будет сгенерировать случайным образом, пределы разброса значений в этом массиве, кол-во интервалов и их левые границы (правой границей интервала по умолчанию считается следующая левая граница).

Генерируется массив и передается в function, определенную в ассемблерном модуле.

В нем реализован цикл, по всем элементам массива чисел. Каждое число сравнивается со всеми левыми границами последовательно, до тех пор, пока число не окажется меньше текущей левой границы. Тогда в массиве result в ячейке с индексом, соответствующим этой левой границе, значение увеличивается на 1 и цикл повторяется для уже следующего числа.

Результаты с помощью ЯВУ выводятся в файл.

Разработанный программный код см. в приложении А.

Вывод.

Разработана программа, состоящая из двух модулей с++ и ассемблера. Программа генерирует массив случайных чисел и подсчитывает количество чисел в каждом интервале.

ПРИЛОЖЕНИЕ А

Название файла: lr6.asm

```
.586p
.MODEL FLAT, C
.CODE
function PROC C USES EDI ESI, array:dword, len:dword, leftBord:dword,
n_int:dword, result:dword

    push eax
    push ebx
    push ecx
    push edi
    push esi

    mov ecx, len
    mov esi, array
    mov edi, leftBord
    mov eax, 0

lp:
    mov ebx, 0 ; индекс текущего интервала
iter:
    cmp ebx, n_int ; while (i < n_int)
    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 to_next_num
    mov edi, result
    push eax
    mov eax, [edi + 4 * ebx]
    inc eax
    mov [edi + 4 * ebx], eax
    pop eax
    mov edi, leftBord

to_next_num:
    inc eax

loop lp

pop esi
pop edi
```

```

pop ecx
pop ebx
pop eax

ret

function ENDP
    END

```

Название файла: main.cpp

```

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

using namespace std;

extern "C" {void function(int* Array, int len, int* LGrInt, int
NInt, int* answer); }

int main() {
    int N, X_min, X_max, N_int;
    cout << "Enter array length:\n";
    cin >> N;
    if (N <= 0 || N > 16 * 1024) {
        std::cout << "Invalid array lenght\n";
        return 1;
    }
    cout << "Enter x-min and x-max:\n";
    cin >> X_min >> X_max;
    if (X_min >= X_max) {
        std::cout << "Invalid Xmin and Xmax values\n";
        return 1;
    }
    cout << "Enter number of intervals\n";
    cin >> N_int;

    if (N_int <= 0 || N_int > 24) {
        cout << "The number of intervals must be between 0 and 24\n";
        return 1;
    }

    cout << "Enter left borders:\n";
    auto intervals = new int[N_int];
    for (int i = 0; i < N_int; ++i) {
        cin >> intervals[i];
        if ((intervals[i] < X_min || intervals[i] > X_max) ||
            (i > 0 && intervals[i] <= intervals[i - 1])) {
            printf("Invalid left border!\n");
            delete[] intervals;
            return 1;
        }
    }
}

```

```

auto numbers = new int[N];
random_device rd;
mt19937 generator(rd());
uniform_int_distribution<> dist(X_min, X_max);
for (int i = 0; i < N; i++) {
    numbers[i] = dist(generator);
}

auto result = new int[N_int];

for (int i = 0; i < N_int; i++) {
    result[i] = 0;
}

function(numbers, N, intervals, N_int, result);

ofstream file("result.txt");
file << "Generated numbers:\n";
for (int i = 0; i < N; ++i) {
    file << numbers[i] << " ";
}
file << "\n\nResults:\n";
for (int i = 0; i < N_int-1; ++i) {
    file << "borders: " << intervals[i] << " - " << intervals[i +
1] - 1 << " amount of numbers - " << result[i] << "\n";
}
file << "borders: " << intervals[N_int-1] << " - " << X_max << "
amount of numbers - " << result[N_int - 1] << "\n";
file.close();
delete[] intervals;
delete[] result;
delete[] numbers;
return 0;
}

```