

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. 1381

Васильева О. М.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2022

Задание (Вариант 2)

На языке С программируется ввод с клавиатуры и контроль исходных данных, а также генерируется массив псевдослучайных целых чисел, изменяющихся в заданном диапазоне и имеющих заданный закон распределения. Необходимые датчики псевдослучайных чисел находятся в каталоге RAND_GEN (при его отсутствии получить у преподавателя).

Следует привести числа к целому виду с учетом диапазона изменения.

Далее должны вызываться 1 или 2 ассемблерных процедуры для формирования распределения количества попаданий псевдослучайных целых чисел в заданные интервалы. Ассемблерные процедуры должны вызываться как независимо скомпилированные модули. Передача параметров в процедуру должна выполняться через кадр стека.

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

В общем случае интервалы разбиения диапазона изменения псевдослучайных чисел могут иметь различную длину, левые границы могут задаваться в произвольном порядке и иметь произвольные значения. Если $X_{min} < LGrInt(1)$, то часть данных не будет участвовать в формировании распределения. Каждый интервал, кроме последнего, следует

интерпретировать как $[LGrInt(i), LGrInt(i+1))$. Если у последнего интервала правая граница меньше X_{max} , то часть данных не будет участвовать в формировании распределения.

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.

Задание на разработку программы выбирается из таблицы 1 в зависимости от номера студента в группе.

Задание в соответствии с номером(вариантом):

- Вид распределения – нормальный;
- Число ассемблерных процедур -1;
- Количество интервалов должно быть больше или равно разности границ диапазона;
- Первая левая граница должна быть больше минимальной границы диапазона;
- Правая граница последнего интервала должна быть больше максимальной границы диапазона.

Выполнение работы.

Исходный код программы см. в приложении А

1. Начало программы прописано в файле main.cpp.
2. Setlocale и system дают возможность работать с кириллицей.
3. Объявляются переменные, отвечающие за количество псевдослучайных чисел, диапазон, количество интервалов. Затем происходит запись пользователем в соответствующие переменные.
4. После записывается массив левых границ в соответствие с

количеством интервалов, также запрашивается правая граница диапазона.

5. Во время всех этих осуществляется проверка на удовлетворение условиям, прописанным в задании.

6. Далее генерируются случайные числа из заданного диапазона в массив numbers.

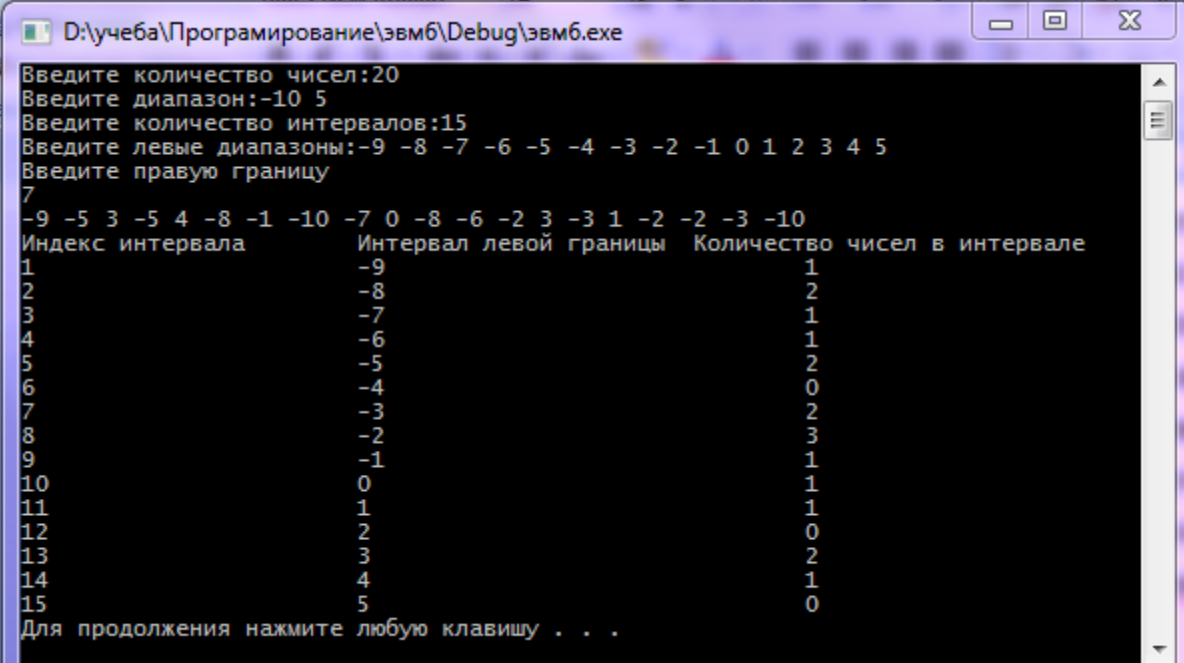
7. Создается результирующий массив final_answer, а далее работу осуществляет функция, написанная на ассемблере.

8. В файле module_.asm прописана функция func, в которой происходит поиск количества чисел, вошедших в каждый интервал.

9. С помощью двух циклов (главных по метке start, а второй по метке border) мы сначала перебираем массив чисел сгенерированных, а во внутреннем цикле перебираем левые границы, для нахождения совпадений.

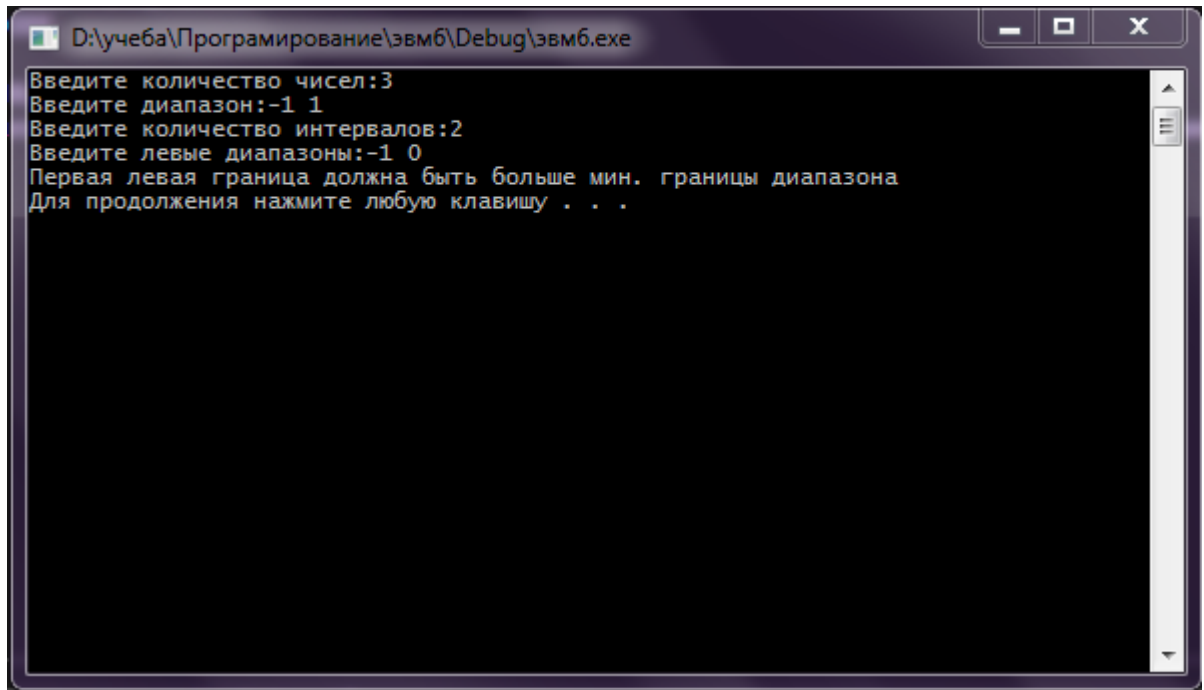
10. По метке print мы записываем число, соответствующее данному диапазону (из итогового массива берем это число, увеличиваем на один и кладем обратно).

Тестирование программы.

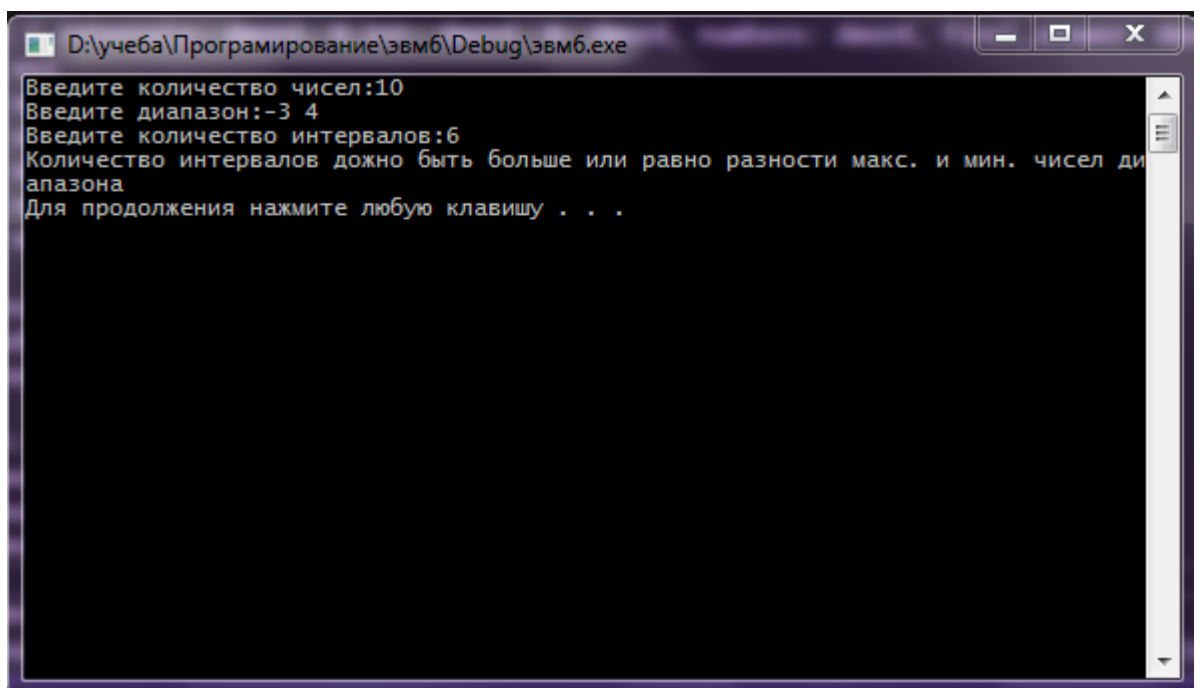


```
D:\учеба\Программирование\эвмб\Debug\эвмб.exe
Введите количество чисел:20
Введите диапазон:-10 5
Введите количество интервалов:15
Введите левые диапазоны:-9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5
Введите правую границу
7
-9 -5 3 -5 4 -8 -1 -10 -7 0 -8 -6 -2 3 -3 1 -2 -2 -3 -10
Индекс интервала    Интервал левой границы    Количество чисел в интервале
1                    -9                        1
2                    -8                        2
3                    -7                        1
4                    -6                        1
5                    -5                        2
6                    -4                        0
7                    -3                        2
8                    -2                        3
9                    -1                        1
10                   0                        1
11                   1                        1
12                   2                        0
13                   3                        2
14                   4                        1
15                   5                        0
Для продолжения нажмите любую клавишу . . .
```

1.



2.



3.

Вывод.

В ходе лабораторной работы был рассмотрен способ организации Ассемблера с ЯВУ. Написана программа построения частотного распределения попаданий псевдослучайных чисел в заданный диапазон.

ПРИЛОЖЕНИЕ А. Код программ.

Имя файла: lab6.cpp

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

using namespace std;

extern "C" void func(int* intervals, int N_int, int N, int* numbers, int*
final_answer);

int main() {
    system("chcp 1251 > nul");
    setlocale(LC_CTYPE, "rus");

    int N, X_min, X_max, N_int;

    cout << "Введите количество чисел:";
    cin >> N;
    cout << "Введите диапазон:";
    cin >> X_min >> X_max;
    cout << "Введите количество интервалов:";
    cin >> N_int;

    if (N_int <= 0 || N_int > 24) {
        cout << "Количество интервалов не соответствует\n";
        system("Pause");
        return 0;
    }

    if (N_int < abs(X_max - X_min)) {
        cout << "Количество интервалов должно быть больше или равно разно-
сти макс. и мин. чисел диапазона\n";
        system("Pause");
        return 0;
    }

    cout << "Введите левые диапазоны:";
    auto intervals = new int[N_int + 1];
    for (int i = 0; i < N_int; i++)
        cin >> intervals[i];

    if (intervals[0] <= X_min) {
        cout << "Первая левая граница должна быть больше мин. границы
диапазона\n";
        system("Pause");
        return 0;
    }

    cout << "Введите правую границу\n";
    cin >> intervals[N_int];

    if (intervals[N_int] < intervals[N_int - 1]){
```



```

        cout << "Правая граница последнего интервала должна быть больше
левой границы последнего интервала\n";
        system("Pause");
        return 0;
    }

    if (intervals[N_int] <= X_max) {
        cout << "Правая граница последнего интервала должна быть больше
макс. границы диапазона\n";
        system("Pause");
        return 0;
    }

    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);
        cout << numbers[i] << " ";
    }

    cout << endl;

    auto final_answer = new int[N_int];

    for (int i = 0; i < N_int; i++)
        final_answer[i] = 0;

    func(intervals, N_int, N, numbers, final_answer);

    ofstream file("output.txt");
    auto str = "Индекс интервала\tИнтервал левой границы\tКоличество чисел
в интервале";
    file << str << endl;
    cout << str << endl;

    for (int i = 0; i < N_int; i++) {
        auto str_result = to_string(i + 1) + "\t\t\t" +
to_string(intervals[i]) + "\t\t\t\t" + to_string(final_answer[i]) + "\n";
        file << str_result;
        cout << str_result;
    }

    system("Pause");

    return 0;
}

```

Имя файла: modile_.asm

.686

.MODEL FLAT, C

.CODE

PUBLIC C func

func PROC C intervals: dword, N_int: dword, N: dword, numbers: dword, fi-
nal_answer: dword

```

push esi
push edi
push eax
push ebx
push ecx
push edx

mov esi, numbers ;массив сгенерированных чисел
mov edi, final_answer ;итоговый массив
mov eax, 0 ;количество пройденных символов в сгенерированном массиве

start:
    mov ebx, [esi + 4*eax] ;текущее число из массива сгенерированных
чисел
    push esi
    mov ecx, N_int ;количество, просмотренных диапазонов

    mov esi, intervals
    cmp [esi + 4*ecx], ebx ;если число находится между правой грани-
цей и макс границей диапазона
    jl finish
    dec ecx

border: ;цикл проверки для нахождения нужного диапазо-
на (выполняется пока не будет найден нужный)
    cmp ebx, [esi + 4*ecx] ; если взятое число больше следующей
левой границы
    jge print ;переход к записи
    dec ecx ; вычитаем от количества интервалов один
    jmp border

print: ;запись в итоговый массив
    mov esi, final_answer
    mov ebx, [esi + 4*ecx] ;берем число из итогового массива

    inc ebx ;увеличиваем на один число из итогового массива
    mov [edi + 4*ecx], ebx ;кладем обратно

finish:
    pop esi
    inc eax ;увеличиваем на один
    cmp eax, N ;если весь массив чисел обработан (количество чи-
сел)
    jne start ;если нет продолжаем проходить по массиву чисел

    pop edx
    pop ecx
    pop ebx
    pop eax
    pop edi
    pop esi
ret
func ENDP
END

```