

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

ОТЧЕТ
по лабораторной работе №5
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. 0382

Деткова А.С

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Научиться строить частотное распределение псевдослучайных чисел в заданные интегралы. Научиться создавать проекты, комбинированные из языков C++ и Ассемблера.

Задание.

Вариант 3.

Распределение — равномерное, число процедур — 2, $N_{int} \geq D_x$ -, $N_{int} < D_x$ +, $L_{gi} \leq X_{min}$ +, $L_{gi} > X_{min}$ -, $П_{Госл} \leq X_{max}$ +, $П_{Госл} > X_{max}$ -.

На языке C 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

$[X_{\min}, X_{\max}]$ (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 - $N_{\text{Int}} (\leq 24)$
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. Варианты заданий различаются:

- 1) видом распределения псевдослучайных чисел: равномерное или нормальное (гаусовское);
- 2) количеством ассемблерных модулей, формирующих требуемое распределение:

- если указан 1 модуль, то он сразу формирует распределение по заданным интервалам и возвращает его в главную программу, написанную на ЯВУ;

- если указаны 2 модуля, то первый из них формирует распределение исходных чисел по интервалам единичной длины и возвращает его в вызывающую программу на ЯВУ как промежуточный результат (это распределение должно выводиться на экран для контроля); затем вызывается второй модуль который по этому промежуточному распределению формирует окончательное распределение псевдослучайных целых чисел по интервалам произвольной длины (с заданными границами).

Это распределение возвращается в главную программу и выдается как основной результат в виде текстового файла.

3) условием – может ли число интервалов быть больше-равно ($N_{int} \geq D_x$) или меньше ($N_{int} < D_x$) диапазона изменения входных чисел;

4) условием – может ли первая левая граница быть больше X_{min} ($Lg1 > X_{min}$) или могут ли какие-то левые границы быть меньше X_{min} ($Lgi \leq X_{min}$);

5) условием – может ли правая граница последнего интервала быть больше X_{max} ($ПГ_{посл} > X_{max}$) или меньше-равна X_{max} ($ПГ_{посл} \leq X_{max}$)

Выполнение работы.

В части программы, написанной на ЯВУ, происходит считывание входных данных и генерация псевдослучайной последовательности заданной длины в заданном интервале. Также происходит вызов функций func1 и func2, написанных на языке Ассемблера. Вывод результатов в файл и консоль.

Func1 — находит частоту встречаемых символов в интервалах длины один.

func2 — находит частоту встречаемости символов в заданных интервалах на базе функции 1.

Разработанный программный код см. в приложении А.

Тестирование.

Введите количество псевдослучайных чисел:

15

Введите границы распределения сначала X_{\min} , затем X_{\max} через пробел:

2 15

Введите количество интервалов разбиения:

3

Введите левые границы:

5 10 7

Массив псевдослучайных чисел:

3 12 9 3 4 7 7 9 15 8 10 2 7 11 13

Распределение исходных чисел по интервалам длины 1:

2: 1; 3: 2; 4: 1; 5: 0; 6: 0; 7: 3; 8: 1; 9: 2; 10: 1; 11: 1; 12: 1; 13: 1; 14: 0; 15:

1;

Результат:

Распределение чисел по интервалам:

Номер интервала Левая граница Количество чисел

0	5	0
1	7	6
2	10	4

Выводы.

Были изучены способы обработки строк и символов на языке Ассемблера. Была разработана программа, которая из строки длиной не более 80 символов составляет новую строку, в которой остаются только буквы русского и английского алфавита.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab6.cpp

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

using namespace std;

extern "C" void func1(int* X, int n, int* res1, int x_min);
extern "C" void func2(int* res1, int* GrInt, int* res2, int
x_max, int x_min, int n);

int main(){

    setlocale(LC_ALL, "ru");

    int NumRamDat;
    cout << "Введите количество псевдослучайных чисел:\n";
    cin >> NumRamDat;
    if (NumRamDat <= 0 || NumRamDat > 16000) {
        cout << "Количество чисел не может быть меньше нуля и
больше 16к\n";
        return 1;
    }

    int Xmax, Xmin;
    cout << "Введите границы распределения сначала Xmin, затем
Xmax через пробел:\n";
    cin >> Xmin >> Xmax;
    if (Xmax <= Xmin) {
        cout << "Недопустимые границы\n";
        return 1;
    }

    int NInt;
    cout << "Введите количество интервалов разбиения:\n";
    cin >> NInt;
    if (NInt <= 0 || NInt > 24 || NInt >= (Xmax - Xmin)) {
        cout << "Количество интервалов не может быть меньше или
равно 0, больше 24, больше или ";
        cout << "равно, чем разность между максимальным и
минимальным значением в диапазоне псевдослучайных чисел\n";
        return 1;
    }

    int* LGrInt = new int[NInt + 1];
    cout << "Введите левые границы:\n";
    for (int i = 0; i < NInt; i++) {
        cin >> LGrInt[i];

        if (LGrInt[i] < Xmin || LGrInt[i] > Xmax) {
```

```

        cout << "Недопустимое значение интервала\n";
        return 1;
    }

    int ind = i;
    while (ind && LGrInt[ind] < LGrInt[ind - 1]) {
        swap(LGrInt[ind--], LGrInt[ind]);
    }
}
LGrInt[NInt] = Xmax;

int* X = new int[NumRamDat];
random_device rd;
mt19937 gen(rd());
uniform_int_distribution<> distrib(Xmin, Xmax);

cout << "Массив псевдослучайных чисел:\n";
for (int i = 0; i < NumRamDat; i++) {
    X[i] = distrib(gen);
    cout << X[i] << ' ';
}
cout << '\n';

int* res_1 = new int[Xmax - Xmin + 1];
for (int i = 0; i < (Xmax - Xmin + 1); i++)
    res_1[i] = 0;

int* res_2 = new int[NInt];
for (int i = 0; i < NInt; i++)
    res_2[i] = 0;

func1(X, NumRamDat, res_1, Xmin);
    cout << "Распределение исходных чисел по интервалам длины
1:\n";
for (int i = 0; i < (Xmax - Xmin + 1); i++)
    cout << i + Xmin << ": " << res_1[i] << "; ";
cout << "\n";

func2(res_1, LGrInt, res_2, Xmax, Xmin, NInt);
ofstream out;
out.open("C:\\Result.txt");
cout << "Распределение чисел по интервалам:\n";
out << "Распределение чисел по интервалам:\n";
cout << "Номер интервала Левая граница Количество чисел\n";
out << "Номер интервала Левая граница Количество чисел\n";
for (int i = 0; i < NInt; i++) {
    cout << setw(8) << i << setw(14) << LGrInt[i] << setw(16)
<< res_2[i] << "\n";
    out << setw(8) << i << setw(14) << LGrInt[i] << setw(16)
<< res_2[i] << "\n";
}
out.close();

return 0;
}

```


Название файла: func1.asm

586

.MODEL FLAT, C

.CODE

PUBLIC C func1

func1 PROC C X:dword, n: dword, res1: dword, x_min: dword

push esi

push edi

mov esi, X

mov edi, res1

mov ecx, n

change:

mov eax, [esi]

sub eax, x_min

mov ebx, [edi + 4*eax]

inc ebx

mov [edi + 4*eax], ebx

add esi, 4

loop change

pop edi

pop esi

ret

func1 ENDP

END

Название файла: func2.asm

.586

.MODEL FLAT, C

.CODE

PUBLIC C func2

func2 PROC C res1:dword, GrInt: dword, res2: dword, x_max: dword,
x_min: dword, n: dword

push esi

push edi

mov esi, GrInt

mov edi, res2

mov ecx, n

lp:

mov eax, [esi]

mov ebx, [esi + 4]

l2:

sub ebx, eax

cmp ebx, 0

jz l1

```

        push ecx
        push esi

        mov ecx, ebx
        sub eax, x_min
        mov esi, res1
        mov ebx, 0

lp2:
        add ebx, [esi + 4*eax]
        inc eax
        loop lp2

pop esi
pop ecx

mov [edi], ebx

cmp ecx, 2
je 13
jmp 11

13:
        add edi, 4
        add esi, 4
        mov eax, [esi]
        mov ebx, x_max
        mov ecx, 1
        jmp 12

11:
        add edi, 4
        add esi, 4

loop lp

pop edi
pop esi

ret
func2 ENDP
END

```