

**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**(РУТ (МИИТ))**

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

**Отчет по практической работе**

«Арифметические команды»

**по дисциплине**

«Машинно-ориентированные языки программирования»

Вариант № 14

**Выполнил:** студент группы ТКИ-342

Фещенко В. А.

**Проверил:** доцент кафедры УиЗИ, к.т.н. Логинова Л. Н.

**Москва 2022 г.**

**Цель работы**

Изучение команд передачи управления и получения навыка работы с ними.

**Задание**

1. С помощью процедуры *Random* сформировать случайное целое число в диапазоне от 0 до 65535;
2. Четное число (четное с точки зрения математики) записать в ЯП (ячейку памяти), начиная с нулевого элемента массива *data*. Нечетное число (нечетное с точки зрения математики) записать в ЯП, начиная с 128 элемента массива *data*. Все числа, которые больше или равны 50000 записать в ЯП, начиная с 256 элемента массива *data*. Все числа, которые меньше 10000 записать в ЯП, начиная с 384 элемента массива *data*;
3. Повторять п.1-п.2, пока число распределенных данных по любому из признаков не превысит 127.

**Код программы**

#include <iostream>

#include <random>

using namespace std;

const int DATA\_SIZE = 512;

const uint16\_t MIN = 0x0;

const uint16\_t MAX = 0xFFFF;

uint16\_t random()

{

random\_device rd;

mt19937 rng(rd());

uniform\_int\_distribution<uint16\_t> uni(MIN, MAX);

return uni(rng);

}

int main()

{

uint16\_t arr[DATA\_SIZE];

for (int i = 0; i < DATA\_SIZE; i++) {

arr[i] = 0;

}

\_\_asm

{

pushad

lea esi, arr // 1) 0 - 127

mov edi, esi // 2) 128 - 255

add edi, 256

mov edx, edi // 3) 256 - 383

add edx, 256

mov ecx, edx // 4) 384 - 511

add ecx, 256

calling:

push ecx

push edx

call random

movzx eax, ax

pop edx

pop ecx

push edx

mov edx, 0

mov ebx, 2

push eax

div ebx

cmp edx, 0

pop eax

pop edx

jne add\_odd

mov[esi], eax

add esi, 2

lea ebx, arr

add ebx, 256

cmp esi, ebx

jge go\_to\_exit

jmp is\_bigger

add\_odd:

mov[edi], eax

add edi, 2

lea ebx, arr

add ebx, 512

cmp edi, ebx

jge go\_to\_exit

is\_bigger:

cmp eax, 50000

jnge is\_smaller

mov[edx], eax

add edx, 2

lea ebx, arr

add ebx, 768

cmp edx, ebx

jge go\_to\_exit

jmp calling

is\_smaller:

cmp eax, 10000

jnle calling

mov[ecx], eax

add ecx, 2

lea ebx, arr

add ebx, 1024

cmp ecx, ebx

jge go\_to\_exit

jmp calling

go\_to\_exit:

popad

}

for (size\_t i = 0; i < DATA\_SIZE; i++)

{

switch (i)

{

case 0:

cout << "even: " << "\n";

break;

case 128:

cout << "\n" << "odd: " << "\n";

break;

case 256:

cout << "\n" << "> 50000: " << "\n";

break;

case 384:

cout << "\n" << "< 10000: " << "\n";

break;

}

cout << arr[i] << " ";

}

return 0;

}

**Блок-схема алгоритма.**



Рисунок - Блок-схема алгоритма

**Результат выполнения программы**

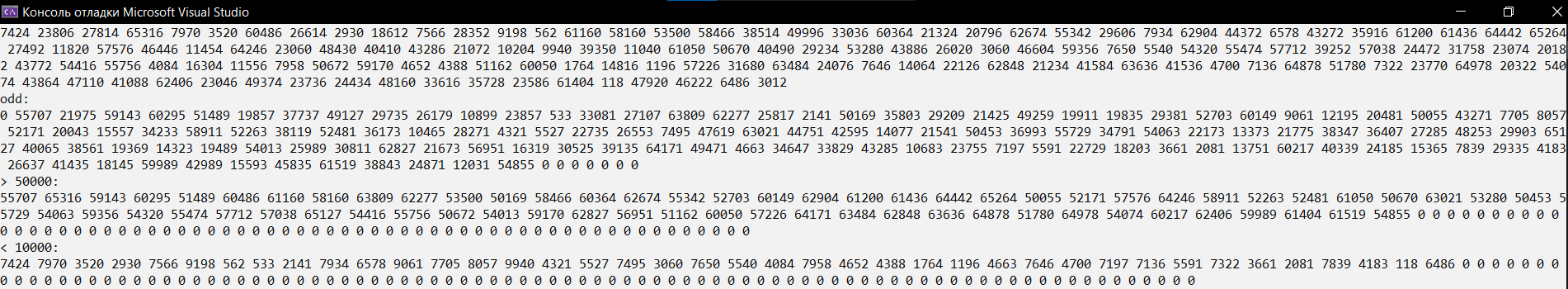
****

Рисунок - Результат выполнения программы

**Контрольные вопросы**

1. Влияет ли результат команды LOOPE на флаги?

Ответ: не влияет.

1. Какой регистр выступает в роли счетчика при организации циклов с помощью команды LOOP?

Ответ: .

1. Какой флаг учитывает команда LOOPNZ при переходе к началу цикла?

Ответ: учитывается флаг (для перехода необходимо ).

1. Чем отличается условный переход от безусловного? Какие команды осуществляют условный переход?

Ответ: безусловный переход осуществится в любом случае, условный переход осуществляется при соответствии условиям. Условный переход записывается в виде , где отвечает за конкретное условие.

1. Какой тип переходов (ближний или дальний) реализуются командами LOOP,LOOPE,LOOPNE?

Ответ: реализуется ближний переход.

1. Сохраняет ли команда безусловного перехода JMP информацию о точке возврата?

Ответ: не сохраняет.

1. Какие значения примут регистры EAX, ECX после выполнения следующего фрагмента программы:

MOV EAX,$2

MOV ECX, 10

@LOOP\_START:

ADD EAX, ECX

LOOP @LOOP\_START

Ответ: 00000039, 00000000.

**Вывод**

В ходе выполнения лабораторной работы были получены знания по использованию циклов, условных и безусловных переходов, вызовов процедур.