МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Череповецкий государственный университет»

**Лабораторная работа № 6**

**«Подпрограммы и стек»**

**Выполнил:**

**студент гр.** 1ИВТпб-01-31оп

Климов А.Г.  
**Проверил: преподаватель**

Виноградова Л.Н.  
**Отметка о зачете:**

Череповец

2018 год

**Задание**

Составить и отладить программу учебной ЭВМ для решения следующей задачи. Три массива в памяти заданы начальными адресами и длинами. Вычислить и вывести на устройство вывода среднее арифметическое параметров этих массивов. Параметры определяются заданием к предыдущей лабораторной работе (см. табл. 10.9), причем соответствие между номерами вариантов заданий к лабораторным работам № 5 и 6 устанавливается по табл. 10.11.

|  |  |
| --- | --- |
| **Номер варианта**  **задания к л. р. № 6** | **Характеристика последовательности чисел C1, C2, ..., Cn** |
| 5 (9) | Минимальное положительное число |

Даны три массива чисел. Требуется вычислить среднее арифметическое их минимальных элементов. Каждый массив задается двумя параметрами: адресом первого элемента и длиной.

Очевидно, в программе трижды необходимо выполнить поиск максимального элемента массива, поэтому следует написать соответствующую подпрограмму.

Параметры в подпрограмму будем передавать через регистры: R1 — начальный адрес массива, R2 — длина массива.

Рассмотрим конкретную реализацию этой задачи. Пусть первый массив начинается с адреса 085 и имеет длину 14 элементов, второй — 100 и 4, третий — 110 и 9. Программа будет состоять из основной части и подпрограммы.

Основная программа задает параметры подпрограмме, вызывает ее и сохраняет результаты работы подпрограммы в рабочих ячейках. Затем осуществляет вычисление среднего арифметического и выводит результат на устройство вывода. В качестве рабочих ячеек используются регистры общего назначения R6 и R7 — для хранения максимальных элементов массивов. Подпрограмма получает параметры через регистры R1 (начальный адрес массива) и R2 (длина массива). Эти регистры используются подпрограммой в качестве регистра текущего адреса и счетчика цикла соответственно. Кроме того, R3 используется для хранения текущего максимума, а R4 — для временного хранения текущего элемента. Подпрограмма возвращает результат через аккумулятор. В табл. 10.10 приведен текст основной программы и подпрограммы. Обратите внимание, цикл в подпрограмме организован с помощью команды JRNZ, а модификация текущего адреса — средствами постинкрементной адресации.

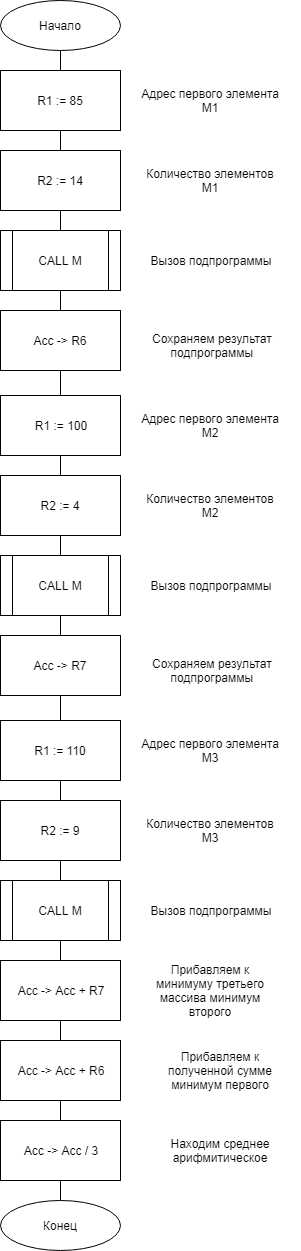


Рис. 1. Граф-схема основной программы

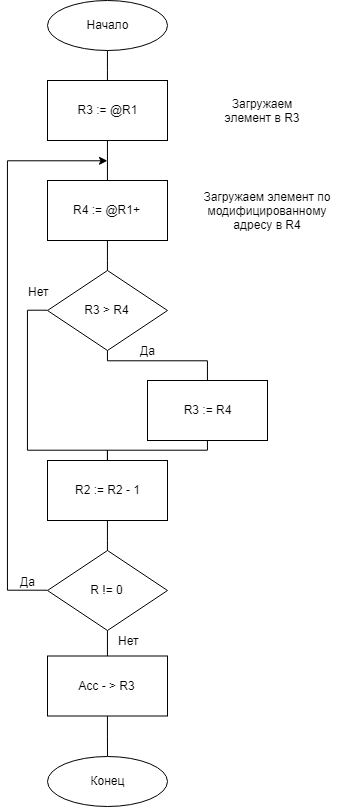


Рис. 2. Граф-схема подпрограммы MIN

Таблица 1

Текст программы с комментариями

|  |  |
| --- | --- |
| **Команда** | **Примечания** |
| **Основная программа** | |
| RD #85 (адрес первого элемента массива) | Загрузка |
| WR R1 | параметров |
| RD #14 (количество элементов) | первого |
| WR R2 | массива |
| CALL M | Вызов подпрограммы |
| WR R6 | Сохранение результата |
| RD #100 | Загрузка |
| WR R1 | параметров |
| RD #4 | второго |
| WR R2 | массива |
| CALL M | Вызов подпрограммы |
| WR R7 | Сохранение результата |
| RD #110 | Загрузка |
| WR R1 | параметров |
| RD #9 | третьего |
| WR R2 | массива |
| CALL M | Вызов подпрограммы |
| ADD R7 | Вычисление |
| ADD R6 | среднего |
| DIV #3 | арифметического |
| OUT | Вывод результата |
| HLT | Стоп |
| **Подпрограмма MIN** | |
| M: RD @R1 | Загрузка |
| WR R3 | первого элемента в R3 |
| L2: RD @R1+ | Чтение элемента и модификация адреса |
| WR R4 | Сравнение |
| SUB R3 | и замена, |
| JNS L1 | если R3 >= R4 |
| MOV R3,R4 |  |
| L1: JRNZ R2,L2 | Цикл |
| RD R3 | Чтение результата в Асс |
| RET | Возврат |
|  |  |
|  |  |

R1 - адрес первого элемента массива

R2 - количество элементов

R3 - адрес первого элемента массива

R4 – следующий элемент массива

10 5

1. Загружаем в R3 из R1 адрес первого элемента массива
2. Считывает элемент из R1 и модифицируем адрес (получаем следующий элемент), а затем записываем его в R4
3. Вычитаем из следующего элемента текущий (R4 – R3)
4. Если R3> =R4, то в R3 записываем R4
5. Выполняем цикл

Текст программы:

RD #85

WR R1

RD #14

WR R2

CALL M

WR R6

RD #100

WR R1

RD #4

WR R2

CALL M

WR R7

RD #110

WR R1

RD #9

WR R2

CALL M

ADD R7

ADD R6

DIV #3

OUT

HLT

M: RD @R1

WR R3

L2: RD @R1+

WR R4

SUB R3

JNS L1

MOV R3,R4

L1: JRNZ R2,L2

RD R3

RET

Пример:

|  |  |
| --- | --- |
| **Команда** | **Примечания** |
| **Основная программа** | |
| RD #85 | Загрузка |
| WR R1 | параметров |
| RD #14 | первого |
| WR R2 | массива |
| CALL M | Вызов подпрограммы |
| WR R6 | Сохранение результата |
| RD #100 | Загрузка |
| WR R1 | параметров |
| RD #4 | второго |
| WR R2 | массива |
| CALL M | Вызов подпрограммы |
| WR R7 | Сохранение результата |
| RD #110 | Загрузка |
| WR R1 | параметров |
| RD #9 | третьего |
| WR R2 | массива |
| CALL M | Вызов подпрограммы |
| ADD R7 | Вычисление |
| ADD R6 | среднего |
| DIV #3 | арифметического |
| OUT | Вывод результата |
| HLT | Стоп |
| **Подпрограмма MAX** | |
| M: RD @R1 | Загрузка |
| WR R3 | первого элемента в R3 |
| L2: RD @R1+ | Чтение элемента и модификация адреса |
| WR R4 | Сравнение |
| SUB R3 | и замена, |
| JS L1 | если R3 < R4 |
| MOV R3,R4 |  |
| L1: JRNZ R2,L2 | Цикл |
| RD R3 | Чтение результата в Асс |
| RET | Возврат |
|  |  |
|  |  |

RD #85

WR R1

RD #14

WR R2

CALL M

WR R6

RD #100

WR R1

RD #4

WR R2

CALL M

WR R7

RD #110

WR R1

RD #9

WR R2

CALL M

ADD R7

ADD R6

DIV #3

OUT

HLT

M: RD @R1

WR R3

L2: RD @R1+

WR R4

SUB R3

JS L1

MOV R3,R4

L1: JRNZ R2,L2

RD R3

RET