**LXI H, 8100** ;задание адреса первого элемента (HL:=8100H)

**MVI D, 40**  ;задание кол-ва элементов

**XRA A**  ;обнуление аккумулятора

**MVI C, 00**  ;обнуление регистра C (счетчик переполнений)

**X1: ADD M** ;суммирование (A:=A+M(HL))

**JNC X2**  ;перейти на Х2 если нет переполнения

**INR C**  ;произошло переполнение → прибавить 1 к старшему байту

результата суммирования

**X2: INX H**  ;присвоить HL адрес следующей ячейки (HL:=HL+1)

**DCR D**  ;уменьшение счётчика кол-ва элементов на 1 (D:=D-1)

**JNZ X1**  ;если элемент не последний - продолжить суммирование

**MOV B, A** ;пересылка младшего байта результата суммирования в регистр В

Сумма 64х элементов находится в паре BC

**MVI D, 06**  ;задание кол-ва сдвигов (6 сдвигов равносильно делению на 64)

**CALL DIV** ;вызов подпрограммы деления

После DIV: Среднее значение находится в регистре В

**LXI H, 8100** ;задание адреса первого элемента (HL:=8100H)

**MVI D, 40**  ;задание кол-ва элементов

**MOV** **A, M ;**будем считать первый элемент минимальным (A:=M(HL))

**X3: CMP M** ;сравниваем минимальный элемент с текущим (A-М(HL))

**JM X4** ;при S=1 (A-M(HL)<0 → A<M(HL)) переход на Х4

**MOV A, M** ;если же S=0 (A-M(HL)≥0 → A≥M(HL)), сделать текущий элемент

наименьшим

**X4: INX H**  ;присвоить HL адрес следующей ячейки (HL:=HL+1)

**DCR D**  ;уменьшение счётчика кол-ва элементов на 1 (D:=D-1)

**JNZ X3**  ;если элемент не последний - продолжить

**MOV C, A** ;пересылка значения AMIN в регистр C

Минимальное значение находится в регистре С

**MOV A, C** ;пересылаем минимальное значение в аккумулятор (C→A)

**RAR** ;циклический сдвиг вправо через перенос (деление на 2)

**ORA A** ;обнуление флага переполнения (ТС:=0)

**MVI C, 00**  ;обнуление регистра C (счетчик переполнений)

**ADD B** ;суммирование: АСР + АMIN/2

**JNC X5** ;перейти на Х5 если нет переполнения

**INR C ;**произошло переполнение → прибавить 1 к старшему байту

результата Y

**X5: MOV B, A** ;пересылаем младший байт из аккумулятора в регистр B

Значение (Аср + Amin/2) находится в паре ВС

**MVI D, 02**  ;задание кол-ва сдвигов (2 сдвига равносильно делению на 4)

**CALL DIV**  ;вызов подпрограммы DIV (деление на 4)

После DIV: Y находится в регистре В

**MOV A, B** ;пересылаем значение Y в регистр аккумулятор (B→A)

**STA 8150 ;**пересылаем значение Y (окончательный результат) в ячейку 8150H

**RST 1**  ;выход из программы

**DIV:** **MOV A, C** ;пересылаем старший байт в аккумулятор (C→A)

**RAR** ;циклический сдвиг вправо через ТС

**MOV C, A** ;возврат в C старшего байта

**MOV A, B** ;пересылаем младший байт в аккумулятор (B→A)

**RAR** ;циклический сдвиг вправо через ТС

**MOV B, A** ;возврат в B младшего байта

**ORA A** ;обнуление флага переполнения (ТС:=0)

**DCR D**  ;уменьшение счётчика кол-ва сдвигов на 1 (D:=D-1)

**JNZ DIV**  ;если сдвиг не последний – продолжить

**RET**  ;выход из подпрограммы DIV