Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

**Лабораторная работа № 2**

Программирование EDSAC

по дисциплине «Низкоуровневое программирование»

Выполнил

студент гр. 3530901/90004

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Остапчук А.С.

(подпись)

Руководитель

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Алексюк А.О.

(подпись)

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Санкт-Петербург   
2021

**Задача**

1. Разработать программу для EDSAC, реализующую определенную вариантом задания функциональность, и предполагающую загрузчик Initial Orders 1. Массив (массивы) данных и другие параметры (преобразуемое число, длина массива, параметр статистики и пр.) располагаются в памяти по фиксированным адресам.

2. Выделить определенную вариантом задания функциональность в замкнутую (closed) подпрограмму, разработать вызывающую ее тестовую программу. Использовать возможности загрузчика Initial Orders 2. Адрес обрабатываемого массива данных и другие параметры передавать через ячейки памяти с фиксированными адресами.

**Вариант задания**

Вариант: 9 - Расчет биномиальных коэффициентов для данного показателя по треугольнику Паскаля.

**Выполнение работы**

Адрес массива IO1 – 100, IO2 - 9

Адрес показателя IO1 – 96, IO2 - 8

Изначально мы имеем массив из 2 элементов, равных 1. Суть алгоритма заключается в том, чтобы проходить по массиву, складывать значения двух соседних элементов и перезаписывать их и увеличивать массив в размерах. По массиву мы проходим столько раз, сколько необходимо для вычисления коэффициентов для заданного показателя.

Код программы Initial Orders 1 с комментариями:  
  
[31:] T 103 S

[32:] Z 0 S [останов для отладки]

[33:] T 0 S [запись аккумулятора в рабочую ячейку 0, обнуление аккумулятор]

[34:] A 95 [c1] S [загружаем счетчик текущего показателя]

[35:] T 2 S [обновляем значение счетчика и обнуляем аккумулятор]

[36:] A 100 [addr] S [загрузка в аккумулятор адреса 0-го элемента массива]

[37:] L 0 L [сдвиг аккумулятора на 1 разряд влево]

[38:] A 52 S [прибавляем код инструкции с полем адреса 0]

[39:] T 52 S [запись сформированной инструкции, обнуление аккумулятора]

[40:] A 100 [addr] S [загрузка в аккумулятор адреса 0-го элемента массива]

[41:] L 0 L [сдвиг аккумулятора на 1 разряд влево]

[42:] A 53 S [прибавляем код инструкции с полем адреса 1]

[43:] T 53 S [запись сформированной инструкции, обнуление аккумулятора]

[44: loop1] A 2 S [загружаем счетчик текущего показателя]

[45:] S 96 [pokaz] S [уменьшаем на необходимый показатель]

[46:] E 93 [exit] S [если результат больше 0, завершаем работу]

[47:] A 95 [c1] S [увеличиваем на 1]

[48:] A 96 [pokaz] S [увеличиваем на необходимый показатель]

[49:] T 2 S [обновляем значение счетчика и обнуляем аккумулятор]

[50:] A 2 S

[51:] T 3 S [обновляем значение счетчика для внутреннего цикла]

[52:] A 0 S [загрузка в аккумулятор значения из ячейки 0-го элемента массива]

[53:] A 1 S [загрузка в аккумулятор значения из ячейки 1-го элемента массива]

[54:] T 0 S [запись этого значения в рабочую ячейку, обнуление аккумулятора]

[55:] A 100 [addr] S [загрузка в аккумулятор адреса 0-го элемента массива]

[56:] L 0 L [сдвиг аккумулятора на 1 разряд влево]

[57:] A 97 [r1] S [прибавляем код инструкции с полем адреса 1]

[58:] T 71 [r1] S [запись сформированной инструкции, обнуление аккумулятора]

[59:] A 100 [addr] S [загрузка в аккумулятор адреса 0-го элемента массива]

[60:] L 0 L [сдвиг аккумулятора на 1 разряд влево]

[61:] A 99 [w1] S [прибавляем код инструкции с нулевым полем адреса]

[62:] T 75 [w1] S [запись сформированной инструкции, обнуление аккумулятора]

[63:] A 100 [addr] S [загрузка в аккумулятор адреса 0-го элемента массива]

[64:] L 0 L [сдвиг аккумулятора на 1 разряд влево]

[65:] A 98 [r2] S [прибавляем код инструкции с полем адреса 2]

[66:] T 72 [r2] S [запись сформированной инструкции, обнуление аккумулятора]

[67: loop2] A 3 S [загружаем счетчик необработанных элементов массива]

[68:] S 95 [c1] S [уменьшаем на 1]

[69:] G 91 S [если результат меньше 0, завершаем вложенный цикл]

[70:] T 3 S [обновляем значение счетчика и обнуляем аккумулятор]

[71:] [r1:] A 1 S [загрузка в аккумулятор значения из ячейки N+1]

[72:] [r2:] A 2 S [загрузка в аккумулятор значения из ячейки N+2]

[73:] T 1 S [запись этого значения в рабочую ячейку 1, обнуление аккумулятора]

[74:] A 0 S [загрузка в аккумулятор значения из ячейки 0]

[75:] [w1:] T 1 S [запись этого значения в ячейку с адресом N+1, обнуление аккумулятора]

[76:] A 1 S [загрузка в аккумулятор значения из ячейки 1]

[77:] T 0 S [запись этого значения в рабочую ячейку, обнуление аккумулятора]

[78:] A 95 [c1] S [загрузка в аккумулятор константы 1]

[79:] L 0 L [сдвиг на 1 разряд влево]

[80:] A 71 [r1] S [прибавляем код инструкции, исполненной на предыдущем шаге]

[81:] T 71 [r1] S [записываем сформированную инструкцию в память]

[82:] A 95 [c1] S [загрузка в аккумулятор константы 1]

[83:] L 0 L [сдвиг на 1 разряд влево]

[84:] A 75 [w1] S [прибавляем код инструкции, исполненной на предыдущем шаге]

[85:] T 75 [w1] S [записываем сформированную инструкцию в память]

[86:] A 95 [c1] S [загрузка в аккумулятор константы 1]

[87:] L 0 L [сдвиг на 1 разряд влево]

[88:] A 72 [r2] S [прибавляем код инструкции, исполненной на предыдущем шаге]

[89:] T 72 [r2] S [записываем сформированную инструкцию в память]

[90:] E 67 [loop2] S

[91:] T 4 S [обнуление аккумулятора]

[92:] E 44 [loop1] S [повторяем все операции; аккумулятор обнулен]

[93:] A 100 S

[94: exit] Z 0 S [останов]

[95: с1] P 0 L [константа 1 (0…0|0|0…00|1 = 1)]

[96: pokaz] P 2 L [ 5 (0…0|0|0…010|1 = 5) (данный показатель) ]

[97: r1 const] A 1 S [загрузка в аккумулятор значения из ячейки N+1]

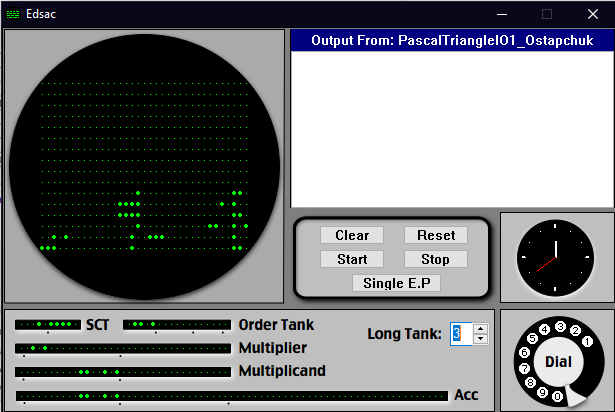
[98: r2 const] A 2 S [загрузка в аккумулятор значения из ячейки N+2]

[99: w1 const] T 1 S [запись этого значения в ячейку с адресом N+1, обнуление аккумулятора]

[100: addr] P 50 L [ 101 адресс первого элемента массива]

[101:] P 0 L [ 1 ]

[102:] P 0 L [ 1 ]



Аналогичный алгоритм используется для Initial Orders 2 и в целом часть подпрограмма базируется на изменённой версии кода для Initial Orders 1. Так же разработана вызывающая ее тестовую программа. Адрес обрабатываемого массива данных и показатель передаются через ячейки памяти с фиксированными адресами (5 и 6 ячейки соответственно).  
  
Код программы Initial Orders 2 с комментариями:

T 56 K

G K [ директива IO2, фиксация начального адреса подпрограммы ]

[0: ] A 3 F [ пролог: формирование кода инструкции возврата в Acc ]

[1: ] T 56 [ ret ] @ [ пролог: запись инструкции возврата ]

[2:] A 57 [c1] @ [загружаем счетчик текущего показателя]

[3:] T 2 F [обновляем значение счетчика и обнуляем аккумулятор]

[4:] A 5 [addr] F [загрузка в аккумулятор адреса 0-го элемента массива]

[5:] A 18 @ [прибавляем код инструкции с полем адреса 0]

[6:] T 18 @ [запись сформированной инструкции, обнуление аккумулятора]

[7:] A 5 [addr] F [загрузка в аккумулятор адреса 0-го элемента массива]

[8:] A 19 @ [прибавляем код инструкции с полем адреса 1]

[9:] T 19 @ [запись сформированной инструкции, обнуление аккумулятора]

[10: loop1] A 2 F [загружаем счетчик текущего показателя]

[11:] S 6 [pokaz] F [уменьшаем на необходимый показатель]

[12:] E 56 [exit] @ [если результат больше 0, завершаем работу]

[13:] A 57 [c1] @ [увеличиваем на 1]

[14:] A 6 [pokaz] F [увеличиваем на необходимый показатель]

[15:] U 2 F [обновляем значение счетчика и обнуляем аккумулятор]

[16:] Z 0 F

[17:] T 3 F [обновляем значение счетчика для внутреннего цикла]

[18:] A 0 F [загрузка в аккумулятор значения из ячейки 0-го элемента массива]

[19:] A 1 F [загрузка в аккумулятор значения из ячейки 1-го элемента массива]

[20:] T 0 F [запись этого значения в рабочую ячейку, обнуление аккумулятора]

[21:] A 5 [addr] F [загрузка в аккумулятор адреса 0-го элемента массива]

[22:] A 58 [r1] @ [прибавляем код инструкции с полем адреса 1]

[23:] T 34 [r1] @ [запись сформированной инструкции, обнуление аккумулятора]

[24:] A 5 [addr] F [загрузка в аккумулятор адреса 0-го элемента массива]

[25:] A 60 [w1] @ [прибавляем код инструкции с нулевым полем адреса]

[26:] T 38 [w1] @ [запись сформированной инструкции, обнуление аккумулятора]

[27:] A 5 [addr] F [загрузка в аккумулятор адреса 0-го элемента массива]

[28:] A 59 [r2] @ [прибавляем код инструкции с полем адреса 2]

[29:] T 35 [r2] @ [запись сформированной инструкции, обнуление аккумулятора]

[30: loop2] A 3 F [загружаем счетчик необработанных элементов массива]

[31:] S 57 [c1] @ [уменьшаем на 1]

[32:] G 54 @ [если результат меньше 0, завершаем вложенный цикл]

[33:] T 3 F [обновляем значение счетчика и обнуляем аккумулятор]

[34:] [r1:] A 1 F [загрузка в аккумулятор значения из ячейки N+1]

[35:] [r2:] A 2 F [загрузка в аккумулятор значения из ячейки N+2]

[36:] T 1 F [запись этого значения в рабочую ячейку 1, обнуление аккумулятора]

[37:] A 0 F [загрузка в аккумулятор значения из ячейки 0]

[38:] [w1:] T 1 F [запись этого значения в ячейку с адресом N+1, обнуление аккумулятора]

[39:] A 1 F [загрузка в аккумулятор значения из ячейки 1]

[40:] T 0 F [запись этого значения в рабочую ячейку, обнуление аккумулятора]

[41:] A 57 [c1] @ [загрузка в аккумулятор константы 1]

[42:] L 0 D [сдвиг на 1 разряд влево]

[43:] A 34 [r1] @ [прибавляем код инструкции, исполненной на предыдущем шаге]

[44:] T 34 [r1] @ [записываем сформированную инструкцию в память]

[45:] A 57 [c1] @ [загрузка в аккумулятор константы 1]

[46:] L 0 D [сдвиг на 1 разряд влево]

[47:] A 38 [w1] @ [прибавляем код инструкции, исполненной на предыдущем шаге]

[48:] T 38 [w1] @ [записываем сформированную инструкцию в память]

[49:] A 57 [c1] @ [загрузка в аккумулятор константы 1]

[50:] L 0 D [сдвиг на 1 разряд влево]

[51:] A 35 [r2] @ [прибавляем код инструкции, исполненной на предыдущем шаге]

[52:] T 35 [r2] @ [записываем сформированную инструкцию в память]

[53:] E 30 [loop2] @

[54:] T 4 F [обнуление аккумулятора]

[55:] E 10 [loop1] @ [повторяем все операции; аккумулятор обнулен]

[56: ret ] E 0 F [ эпилог: инструкция возврата из подпрограммы ]

[57: с1] P 0 D [константа 1 (0…0|0|0…00|1 = 1)]

[58: r1 const] A 1 F [загрузка в аккумулятор значения из ячейки N+1]

[59: r2 const] A 2 F [загрузка в аккумулятор значения из ячейки N+2]

[60: w1 const] T 1 F [запись этого значения в ячейку с адресом N+1, обнуление аккумулятора]

G K [ директива IO2, фиксация начального адреса программы ]

[1:] A 8 [addr] @ [ загружаем адрес массива ]

[2:] T 5 F [ запись адреса массива в ячейку 5, обнуление аккумулятора ]

[3:] A 7 [pokaz] @ [ загружаем показатель ]

[4:] T 6 F [ запись показателя в ячейку 6, обнуление аккумулятора ]

[5:] A 5 @ [\ вызов ]

[6:] G 56 F [/ подпрограммы ]

[7:] Z 0 F [ останов ]

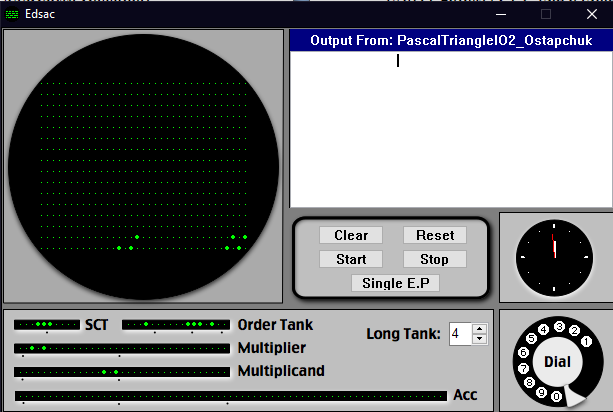
[8: pokaz] P 2 D [ 5 (0…0|0|0…010|1 = 5) (данный показатель) ]

[9: addr ] P 9 @ [ адрес массива ]

[10:] P 0 D

[11:] P 0 D

[12:] EZ PF



**Вывод**

В ходе выполнения лабораторной работы была разработана программа для EDSAC, выполняющая расчет биномиальных коэффициентов для данного показателя по треугольнику Паскаля предполагающую загрузчики Initial Orders 1 и Initial Orders 2.