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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Гасилов А.С.

(подпись)

Руководитель

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

(подпись)

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

Санкт-Петербург   
2021

**Задача**

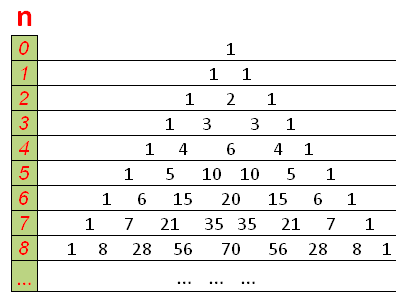
1. Разработать программу для EDSAC, реализующую определенную вариантом задания функциональность, и предполагающую загрузчик Initial Orders 1. Массив (массивы) данных и другие параметры (преобразуемое число, длина массива, параметр статистики и пр.) располагаются в памяти по фиксированным адресам.

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**

Треугольник Паскаля – способ определения биноминальных коэффициентов, где боковые числа всегда равны единицам, а любое равно сумме двух чисел находящихся над ним  
  
 Для определения биноминальных коэффициентов показателя n необходим массив из n+1 чисел (Начальное состояние 1 и n нулей или две единицы и n-1 нулей), который будет заполняться итеративно начиная с 0 или 1 показателя.

В своей программе я начинал с нулевого показателя(1 и n нулей) и рассматривал показатель n = 5. Суть алгоритма заключается в том, чтобы проходить по массиву c конца, складывать значения двух соседних элементов и записывать их в ячейку правого числа.   
По массиву мы проходим n раз, до тех пор, пока не будут попределены все значения биноминальных коэффициентов.

Данная версия программы не является оптимальной, т.к мы на каждой итерации проходим по всем элементам массива (n+1), в то время как на k-той итерации нам нужно рассматривать только первые k+1 эл-тов

Код программы Initial Orders 1 с комментариями:  
  
T 102 S

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

A 87 [<с5>] S [загрузка в аккумулятор начального числа итераций ]

T 0 S [запись этого значения в ячейку]

A 87 [<с5>] S [ загрузка в аккумулятор показателя числа ]

T 1 S [ запись этого значения в ячейку]

A 90 [<addr>] S [загрузка в аккумулятор адреса Предпоследнего элемента массива ]

L 0 L [ сдвиг аккумулятора на 1 разряд влево]

A 57 [<r1>] S [ прибавляем код инструкции с нулевым полем адреса]

T 57 [<r1>] S [ запись сформированной инструкции, обнуление аккумулятора]

A 90 [<addr>] S [ загрузка в аккумулятор адреса Предпоследнего элемента массива]

L 0 L [ сдвиг аккумулятора на 1 разряд влево ]

A 58 [<r2>] S [ прибавляем код инструкции с нулевым полем адреса]

T 58 [<r2>] S [запись сформированной инструкции, обнуление аккумулятора]

A 90 [<addr>] S [загрузка в аккумулятор адреса Предпоследнего элемента массива]

L 0 L [сдвиг аккумулятора на 1 разряд влево]

A 59 [<w2>] S [прибавляем код инструкции с полем адреса 1]

T 59 [<w2>] S [запись сформированной инструкции, обнуление аккумулятора]

[loop1:] A 1 S [ загружаем счетчик порядка числа ]

S 88 [<с1>] S [ уменьшаем на 1 ]

G 86[<exit1>]S [ecли результат меньше 0,завершаем работу]

T 1 S [обновляем значение счетчика и обнуляем аккумулятор]

[loop2:]A 0 S [загружаем счетчик необработанных элементов массива]

S 88 [<с1>] S [ уменьшаем на 1 ]

G 70[<exit2>]S [если счетчик необработанных элементов <0 - покидаем внутренинй цикл]

T 0 S [обновляем значение счетчика и обнуляем аккумулятор]

[r1:] A 0 S [ загрузка в аккумулятор значения из ячейки N ]

[r2:] A 1 S [Прибавление к значению аккумулятора значения в ячейке N+1 ]

[w2:] T 1 S [ запись этого значения в ячейку с адресом N+1, обнуление аккумулятора]

A 57[<r1>] S [загрузка в аккумулятор инструкции]

S 89[<c2>] S [Уменьшение адресса ячейки массива на 1 (2 - сдвиннутая влево единица)]

T 57[<r1>] S [Запись обновленной инструкции]

A 58[<r2>] S [загрузка в аккумулятор инструкции]

S 89[<c2>] S [Уменьшение адресса ячейки массива на 1 (2 - сдвиннутая влево единица)]

T 58[<r2>] S [Запись обновленной инструкции]

A 59[<w2>] S [загрузка в аккумулятор инструкции]

S 89[<c2>] S [Уменьшение адресса ячейки массива на 1 (2 - сдвиннутая влево единица)]

T 59[<w2>] S [Запись обновленной инструкции]

E 53 [<loop2>] S [Если счетчик >= 0 - возвращаемся в начало цикла]

[exit2:] T 0 S [Очистка аккумулятора. Значение счетчика будет обновленно в следующем шаге]

A 87 [<с5>] S [загрузка константы 5 для возвращения счетчика необработанных элементов в начальное положение]

T 0 S [запись этого значения в ячейку]

A 87 S [<с5>] [Загрузка константы 5 для возвращания инструкции в начальное положение]

L 0 L [сдвиг аккумулятора на 1 разряд влево]

A 57 S[<r1>] [прибавляем код инструкции с нулевым полем адреса]

T 57 S[<r1>] [Запись обновленной инструкции]

A 87 [<с5>] S [Загрузка константы 5 для возвращания инструкции в начальное положение]

L 0 L [ сдвиг аккумулятора на 1 разряд влево]

A 58 [<r2>] S [ прибавляем код инструкции с нулевым полем адреса]

T 58 [<r2>] S [Запись обновленной инструкции]

A 87 [<с5>] S [Загрузка константы 5 для возвращания инструкции в начальное положение]

L 0 L [ сдвиг аккумулятора на 1 разряд влево]

A 59 [<w2>] S [прибавляем код инструкции с нулевым полем адреса]

T 59 [<w2>] S [Запись обновленной инструкции]

E 49 [<loop1>] S

[exit1:]Z 0 S [ останов ]

[с5:] P 2 L [ константа 5 (показатель числа)]

[с1:] P 0 L [ константа 1]

[c2:] P 1 S [ константа 2]

[addr:] P 50 S [ Предпоследний элемент массива ]

P 0 S [ Отступ для более наглядного отображения массива ]

P 0 S [ ]

P 0 S [ ]

P 0 S [ ]

P 0 S [ ]

P 0 L [ 1 ]

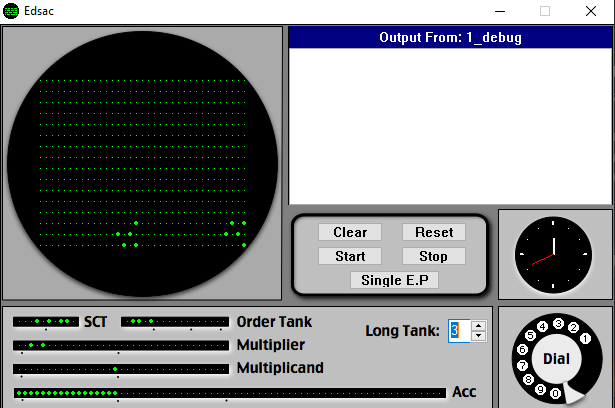
P 0 S [ 0 ]

P 0 S [ 0 ]

P 0 S [ 0 ]

P 0 S [ 0 ]

P 0 S [ 0 ]



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.   
  
Код программы Initial Orders 2 с комментариями:

T 56 K [ директива IO2, установка адреса загрузки ]

G K [ директива IO2, фиксация начального адреса подпрограммы ]

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

[ 1:]T 51 [<ret>] @ [ пролог: запись инструкции возврата ]

[ 2:]A 0 [<addr>] F [загрузка в аккумулятор адреса Предпоследнего элемента массива ]

[ 3:]A 55 [<r2init>] @ [ прибавляем код инструкции с нулевым полем адреса ]

[ 4:]T 21 [<r1>] @ [ запись сформированной инструкции, обнуление аккумулятора]

[ 5:]A 0 [<addr>] F [ загрузка в аккумулятор адреса Предпоследнего элемента массива]

[ 6:]A 56 [<r2init>] @ [ прибавляем код инструкции с нулевым полем адреса ]

[ 7:]T 22 [<r2>] @ [запись сформированной инструкции, обнуление аккумулятора]

[ 8:]A 0 [<addr>] F [загрузка в аккумулятор адреса Предпоследнего элемента массива]

[ 9:]A 57 [<w2init>] @ [ прибавляем код инструкции с нулевым полем адреса ]

[ 10:]T 23 [<w2>] @ [запись сформированной инструкции, обнуление аккумулятора]

[ 11:]A 52 [<с5>] @ [загрузка в аккумулятор начального числа итераций ]

[ 12:]T 0 F [запись этого значения в ячейку]

[ 13: loop1:] A 1 F [ загружаем счетчик порядка числа ]

[ 14:]S 53 [<с1>] @ [ уменьшаем на 1 ]

[ 15:]G 50[<exit1>]@ [ecли результат меньше 0,завершаем работу]

[ 16:]T 1 F [обновляем значение счетчика и обнуляем аккумулятор]

[ 17: loop2:]A 0 F [загружаем счетчик необработанных элементов массива]

[ 18:]S 53 [<с1>] @ [ уменьшаем на 1 ]

[ 19:]G 34[<exit2>]@ [если счетчик необработанных элементов <0 - покидаем внутренинй цикл]

[ 20:]T 0 F [обновляем значение счетчика и обнуляем аккумулятор]

[ 21: r1:] A 0 F[ загрузка в аккумулятор значения из ячейки N ]

[ 22: r2:] A 1 F [Прибавление к значению аккумулятора значения в ячейке N+1 ]

[ 23: w2:] T 1 F [ запись этого значения в ячейку с адресом N+1, обнуление аккумулятора]

[ 24:]A 21[<r1>] @ [загрузка в аккумулятор инструкции]

[ 25:]S 54[<c2>] @ [Уменьшение адресса ячейки массива на 1 (2 - сдвиннутая влево единица)]

[ 26:]T 21[<r1>] @ [Запись обновленной инструкции]

[ 27:]A 22[<r2>] @ [загрузка в аккумулятор инструкции]

[ 28:]S 54[<c2>] @ [Уменьшение адресса ячейки массива на 1 (2 - сдвиннутая влево единица)]

[ 29:]T 22[<r2>] @ [Запись обновленной инструкции]

[ 30:]A 23[<w2>] @ [загрузка в аккумулятор инструкции]

[ 31:]S 54[<c2>] @ [Уменьшение адресса ячейки массива на 1 (2 - сдвиннутая влево единица)]

[ 32:]T 23[<w2>] @ [Запись обновленной инструкции]

[ 33:]E 17 [<loop2>] @ [Если счетчик >= 0 - возвращаемся в начало цикла]

[ 34: exit2:] T 0 F [Очистка аккумулятора. Значение счетчика будет обновленно на следующем шаге]

[ 35:]A 52 [<с5>]@ [загрузка константы 5 для возвращения счетчика необработанных элементов в начальное положение]

[ 36:]T 0 F [запись этого значения в ячейку]

[ 37:]A 52 [<с5>]@ [Загрузка константы 5 для возвращания инструкции в начальное положение]

[ 38:]L 0 D [сдвиг аккумулятора на 1 разряд влево]

[ 39:]A 21 [<r1>]@ [прибавляем код инструкции с нулевым полем адреса]

[ 40:]T 21 [<r1>]@ [Запись обновленной инструкции]

[ 41:]A 52 [<с5>] @ [Загрузка константы 5 для возвращания инструкции в начальное положение]

[ 42:]L 0 D [ сдвиг аккумулятора на 1 разряд влево]

[ 43:]A 22 [<r2>] @ [ прибавляем код инструкции с нулевым полем адреса]

[ 44:]T 22 [<r2>] @ [Запись обновленной инструкции]

[ 45:]A 52 [<с5>] @ [Загрузка константы 5 для возвращания инструкции в начальное положение]

[ 46:]L 0 D [ сдвиг аккумулятора на 1 разряд влево]

[ 47:]A 23 [<w2>] @ [прибавляем код инструкции с нулевым полем адреса]

[ 48:]T 23 [<w2>] @ [Запись обновленной инструкции]

[ 49:]E 13 [<loop1>] @

[ 50: exit1:] T 0 F [ обнуление аккумулятора ]

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

[ 52: с5:] P 2 D [ константа 5 (показатель числа)]

[ 53: с1:] P 0 D [ константа 1]

[ 54: c2:] P 1 F [ константа 2]

[ 55:] [r1init:] A 0 F [ основа для формирования инструкции с меткой r1 ]

[ 56:] [r2init:] A 1 F [ основа для формирования инструкции с меткой r2 ]

[ 57:] [w2init:] T 1 F [ основа для формирования инструкции с меткой w2 ]

G K [ директива IO2, фиксация начального адреса программы ]

[ 0:]X 0 F [ для пошаговой отладки использовать Z 0 F ]

[ 1:] A 8 [<addr>] @ [ адрес массива (в начале работы обнуление аккумулятора не требуется) ]

[ 2:] T 0 F [ запись адреса массива в ячейку 0, обнуление аккумулятора ]

[ 3:] A 9 [<Pok>] @ [ Показатель числа ]

[ 4:] T 1 F [ запись длины массива в ячейку 1, обнуление аккумулятора ]

[ 5:] A 5 @ [\ вызов ]

[ 6:] G 56 F[/ подпрограммы ]

[ 7:] Z 0 F [ останов ]

[ 8: addr:] P 18 @ [ адрес предпоследнего элемента массива ]

[ 9: pok:] P 2 D [ показатель числа - 5 ]

[ 10:] P 0 F [ Отступ для более наглядного отображения массива ]

[ 11:] P 0 F [ ]

[ 12:] P 0 F [ ]

[ 13:] P 0 F [ ]

[ 14: array:] P 0 D [ 1 ]

[ 15:] P 0 F [ 0 ]

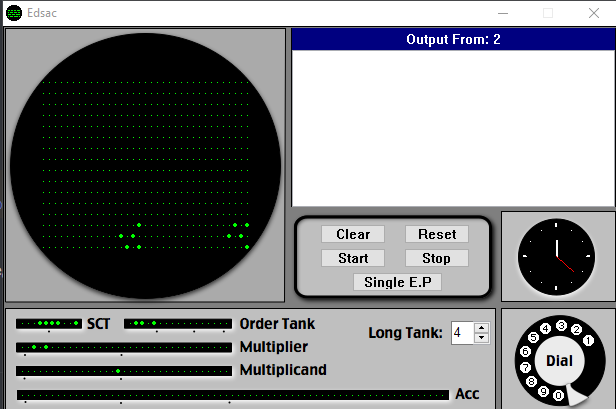
[ 16:] P 0 F [ 0 ]

[ 17:] P 0 F [ 0 ]

[ 18:] P 0 F [ 0 ]

[ 19:] P 0 F [ 0 ]

EZ PF [ директива IO2, переход к исполнению ]



**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.