

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Отчёт по лабораторной работе № 2

Дисциплина: Низкоуровневое программирование

Тема: Программирование EDSAC

Вариант: 9

Выполнил студент гр. 3530901/00002 _____ М.А. Разин
(подпись)

Принял преподаватель _____ Д.С. Степанов
(подпись)

“ ____ ” _____ 2021 г.

Санкт-Петербург
2021

Задача:

Разработать программу для EDSAC, реализующую согласно варианту задания функциональность, и предполагающую загрузчик Initial Orders 1. Параметры (показатель, параметр статистики и пр.) располагаются в памяти по фиксированным адресам.

Выделить определенное вариантом задания выполнение функции в замкнутую (closed) подпрограмму, разработать вызывающую ее тестовую программу. Использовать возможности загрузчика Initial Orders 2. Параметры передавать через ячейки памяти с фиксированными адресами.

Вариант задания:

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

Алгоритм:

```
fun BinCoeff(n: Int): List<Int> {  
    var a = 1  
    val res = mutableListOf<Int>()  
    for (t in 1..n + 1) {  
        res += a  
        a = a * (n + 1 - t) / t  
    }  
    return res  
}
```

Рис. 1 Расчет n-ой строки Треугольника Паскаля, Kotlin.

Выполнение работы:

Ю1: Нужные данные хранятся в 33-39 ячейки, вывод коэффициентов производится, начиная с 100 ячейки.

Ю2: Вывод коэффициентов производится, начиная с 128 ячейки.

Код программы Initial Orders 1 с комментариями:

[31] T 90 S

[32] E 40 S

[Const]

[33] P 0 L [1]

[34] P 2 L [Показатель n=5]

[35] P 0 L [Current coefficient a = 1]

[36] P 50 L [address Num = 100]

[37] PS [EndCounter k = n]

[38] XS

[39] P 0 L [CycleCounter t]

[ProgramBody]

[40] A 34 S [Выгружаю степень для установления счетчика]

[41] T 37 S [Запись подготовленного счетчика k]

[42] A 35 S [Выгружаю первый коэффициент]

[43] T 100 S [Запись первого коэффициента]

[44] A 36 S [Загружаю адрес 2-го коэффициента]

[45] L 0 L [Коррекция]

[46] A 75 S [Добавляю код инструкции с нулевым полем адреса]

[47] T 75 S [Запись сформированной инструкции]

[Внешний цикл]

[48] T 0 S [Чищу акк/Начало цикла]

[49] A 34 S [Выгружаю показатель n]

[50] A 33 S [Прибавляю 1]

[51] S 39 S [Отнимаю t]

[52] T 25 S [Делимое хранится в 25 ячейке]

[$a*((n+1)-t)$]

[53] H 35 S [Выгружаю текущий коэфф в умножающий регистр]

[54] V 25 S [Умножение текущего значения результата на x]

[55] L 64 S [Коррекция]

[56] L 64 S [Коррекция]

[57] T 25 S [Запись Полученного делимого]

[58] X S

[Получение нового коэффициента/Внутренний цикл]

[59] T 26 S [Обнуляю частное, хранящееся в 26 ячейке]

[60] A 25 S [Загружаю делимое в акк]

[61] S 39 S [Вычитаю делитель]

[62] T 25 S [Записываю новое значение делимого]

[63] A 26 S [Загружаю частное]

[64] A 33 S [Частное + 1]

[65] T 26 S [Записываю увеличенное частное]

[66] A 25 S [Выгружаю в аккумулятор делимое]
[67] G 70 S [Если делимое < 0 , выход из цикла]
[68] T 25 S [Чищу содержимое аккумулятора]

[69] E 60 S [Иначе повторяю цикл]
[70] T 0 S [Чищу содержимое аккумулятора]
[71] A 26 S [Выгружаю частное]
[72] S 33 S [Вычитаю 1 для коррекции]
[73] U 26 S [Загружаю частное обратно]
[74] U 35 S [Заменяю коэффициент]
[cout]
[75] T 0 S [Запись коэффициента]

[76] A 39 S [Выгружаю t]
[77] A 33 S [t+=1]
[78] T 39 S [Записью полученного t]
[79] A 37 S [Выгружаю k]
[80] S 33 S [k-=1]
[81] T 37 S [Запись полученного k]

[82] A 33 S [Загружаю 1]
[83] L 0 L [Сдвигаю влево на 1]
[84] A 75 S [Прибавляю код инструкции, исполненной ранее]
[85] T 75 S [Записываю сформированную инструкцию в память]

[86] A 37 S [Выгружаю счетчик k]
[87] S 33 S [k-1]
[88] E 48 S [K<0 стоп, иначе вернуться в начало внешнего цикла]
[89] Z 0 S

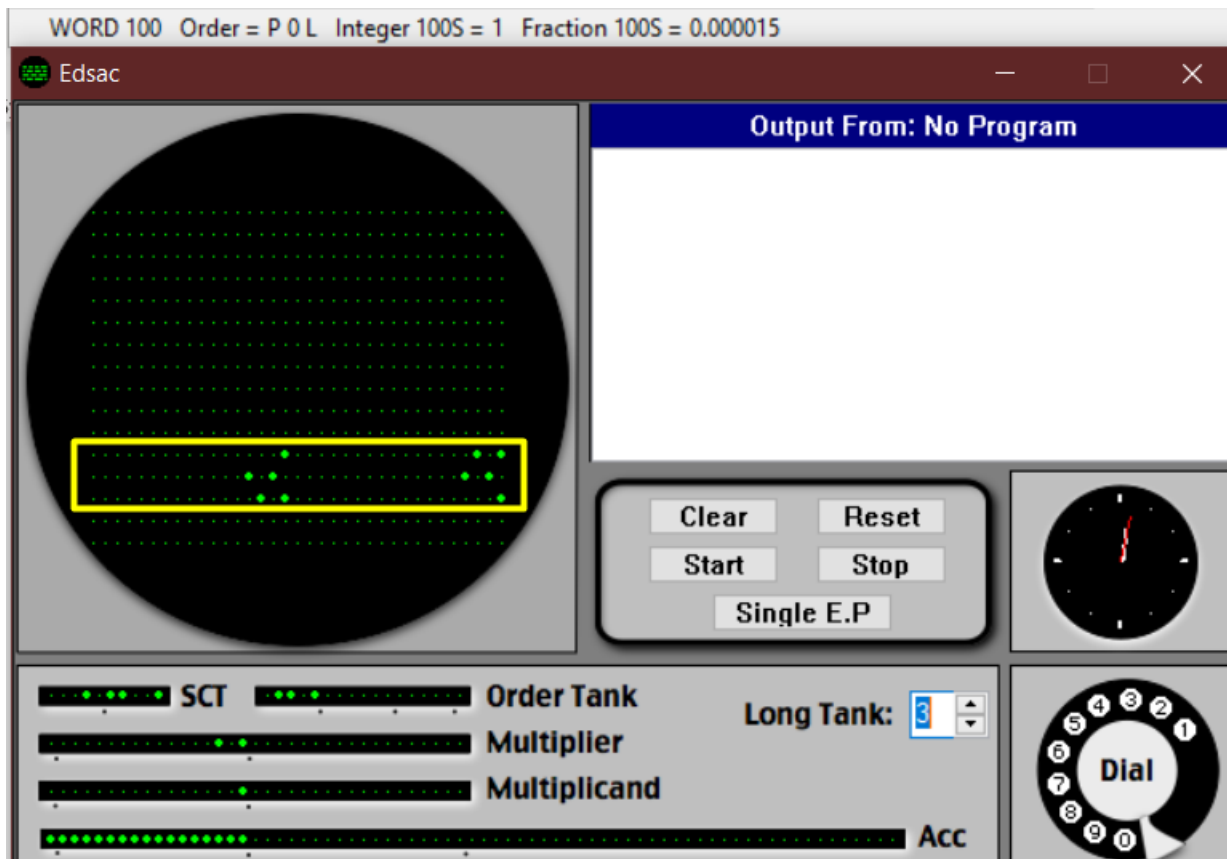


Рис. 2 Результаты работы программы для IO1

На рис. 2 представлены результаты работы программы для показателя $n=5$. Последовательно выведены коэффициенты (1, 5, 10, 10, 5, 1) в 100, 101...105 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 акк]

[1] T 49 [<ret>] @ [пролог: запись инструкции возврата]

[2] A 51 @ [Загружаю адрес 2-го коэффициента]

[3] L 0 D [Коррекция]

[4] A 34 @ [Добавляю код инструкции с нулевым полем адреса]

[5] T 34 @ [Запись сформированной инструкции]

[6] A 1 F [Загружаю в акк первый коэффициент]

[7] T 128 F [Запись в 52 ячейку]

[Цикл : получение нового коэффициента]

[8] T 6 F [Чищу акк]

[9] A 0 F
[10] A 50 @ [n+1-t]
[11] S 3 F
[12] T 5 F

[13] H 1 F [$a*((n+1)-t)$]
[14] V 5 F
[15] L 64 F
[16] L 64 F
[17] T 5 F

[Деление, получение нового a]
[18] T 4 F [Обнуляю частное]
[19] A 5 F [Загружаю делимое полученное с 6-14]
[20] S 3 F [Вычитаю делитель t]
[21] T 5 F [Записываю новое значение делимого]
[22] A 4 F [Загружаю частное]
[23] A 50 @ [Частное + 1]
[24] T 4 F [Записываю увеличенное частное]
[25] A 5 F [Выгружаю в аккумулятор делимое]
[26] G 29 @ [Если делимое < 0, стоп]
[27] T 5 F [Чищу акк]

[28] E 19 @ [Иначе повторяю цикл]
[29] T 6 F [Чищу акк]
[30] A 4 F [Выгружаю частное]
[31] S 50 @ [Вычитаю 1 для коррекции]
[32] U 4 F [Загружаю частное обратно]
[33] U 1 F [Загружаю как новый коэффициент]
[cout]
[34] T 0 F [Вывожу коэффициент в след ячейку]

[35] A 3 F
[36] A 50 @ [t+=1]
[37] T 3 F
[38] A 2 F
[39] S 50 @ [k-=1]
[40] T 2 F

[41] A 50 @ [Загружаю 1]
[42] L 0 D [Сдвигаю влево на 1]
[43] A 34 @ [Прибавляю код инструкции, исполненной ранее]
[44] T 34 @ [Записываю сформированную инструкцию в память]

[45] A 2 F [Выгружаю в акк k]
[46] S 50 @ [K -1]
[47] E 8 @ [K<0 - выход]
[48] T 6 F [Чищу акк]
[49] E 0 F [Эпилог: инструкция возврата из подпрограммы]
[50] P 0 D [const 1]
[51] P 64 D [address 53]

G K [директива IO2, фиксация начального адреса программы]

[0] X F
[1] A 15 [<power>] @ [степень n]
[2] T 0 F [запись степени в ячейку 0, обнуление акк]
[3] A 16 [<coefficient>] @ [текущий коэффициент a]
[4] T 1 F [запись коэффициента в ячейку 1, обнуление акк]
[5] A 15 @ [степень n]
[6] T 2 F [k = n]
[7] A 16 @ [Выгружаю в акк 1]
[8] T 3 F [t = 1]
[9] T 4 F [Промеж результат частного/будущий коэф]
[10] T 5 F [Будущий $a*((n+1)-t)$]
[11] T 6 F

[12] A 12 @ [\ вызов]
[13] G 56 F [/ подпрограммы]
[14] Z 0 F [останов]

[Power]
[15] P 4 F [n = 8]
[16] P 0 D [const 1]

EZ PF [директива IO2, переход к исполнению]

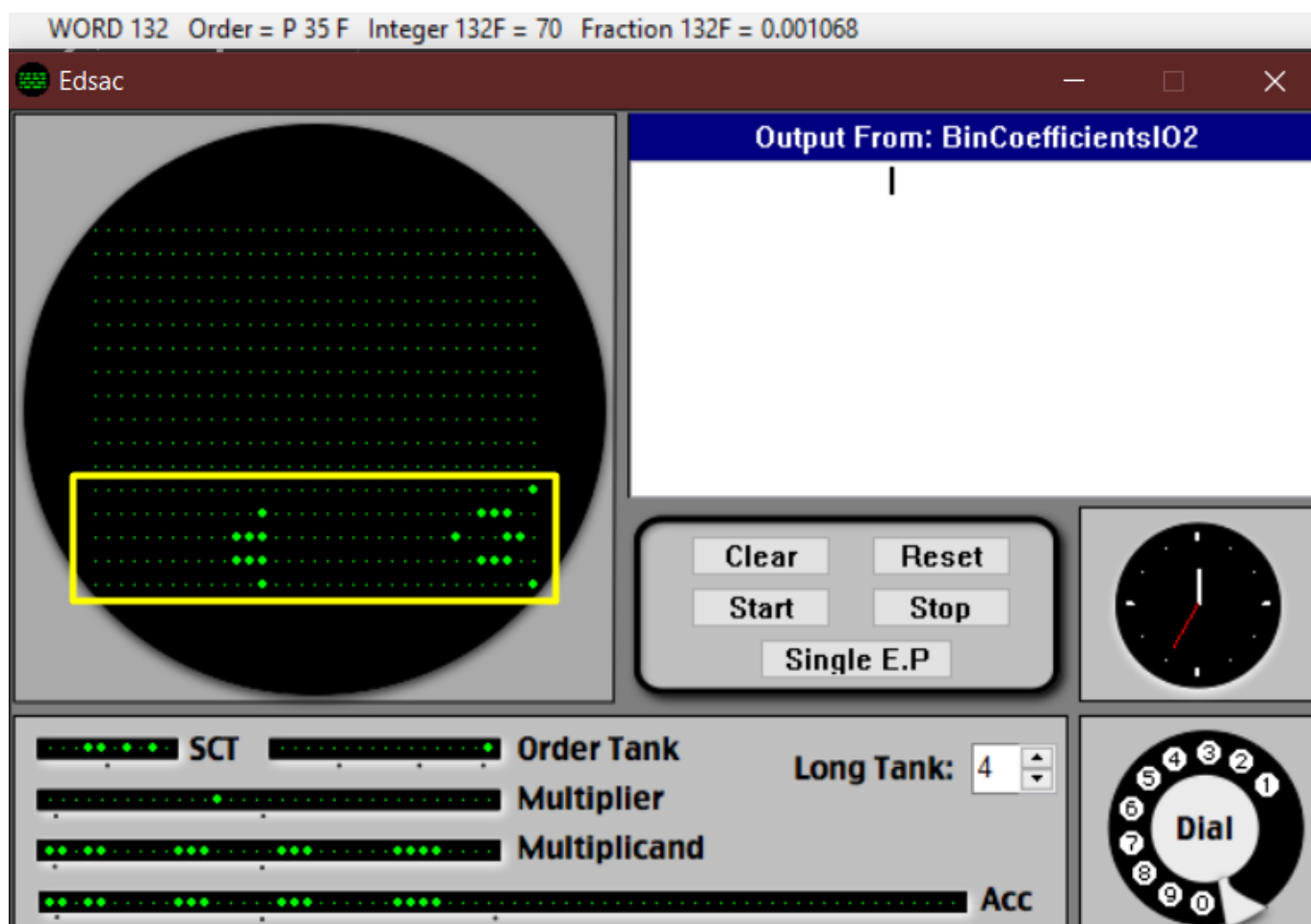


Рис. 3 Результаты работы программы для IO2

На рис. 3 представлены результаты работы программы для показателя $n=8$. Последовательно выведены коэффициенты (1, 8, 28, 56, 70,...,1) в 128, 129...136 ячейки соответственно.

Вывод:

В ходе выполнения работы были успешно написаны две программы (IO1 и IO2) для EDSAC, реализующие расчет биномиальных коэффициентов для данного показателя по треугольнику Паскаля.

Список использованных источников:

http://kspt.icc.spbstu.ru/media/files/2019/lowlevelprog/edsac_io.pdf

http://kspt.icc.spbstu.ru/media/files/2019/lowlevelprog/edsac_prg.pdf

http://kspt.icc.spbstu.ru/media/files/2020/lowlevelprog/edsac_sub.pdf

http://kspt.icc.spbstu.ru/media/files/2019/lowlevelprog/edsac_subprg_c.pdf