

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

Отчет по лабораторной работе №2

Тема: Машина EDSAC

Вариант 16

Выполнил студент гр. 3530901/90002 _____ П. В. Рубинова
(подпись)

Руководитель _____ Д. С. Степанов
(подпись)

“ ____ ” _____ 2021 г.

Санкт-Петербург

2021

Постановка задачи:

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.

Формулировка задачи:

Определение максимального простого числа, не превышающего заданное число N , методом решета Эратосфена (используется рабочий массив длины не более N).

Теория:

Решето Эратосфена — алгоритм нахождения всех простых чисел до некоторого целого числа N , который приписывают древнегреческому математику Эратосфену Киренскому.

В данном случае решето подразумевает фильтрацию всех чисел за исключением простых. По мере обработки массива чисел нужные числа (простые) остаются, а ненужные (составные) исключаются.

Программа, предполагающая загрузчик Initial Orders 1

В ячейках [97] – [106] располагается массив, заполненный единицами.

Проход по массиву осуществляется во внешнем цикле 1 в ячейках [42] – [86].

Внутренний цикл 2, в котором помечаем числа кратные встреченному простому числу, располагается в ячейках [69] – [84].

Результат работы программы располагается в ячейке [96].

Алгоритм работы:

Изначально все числа в массиве равны единице, то есть помечены как простые. Проходя по массиву и встречая простое число N, мы записываем его в результат, а после заходим во внутренний цикл, в котором помечаем кратные N числа как составные. В конечном итоге результатом будет последнее встреченное простое число.

Текст программы:

T 107 K

[32]Z 0 D [Точка останова для отладки]

[33]T 0 S [Очищаем аккумулятор]

[34]A 94 S [Загружаем в аккумулятор длину массива]

[35]T 0 S [Кладем ее в нулевую ячейку]

[36]A 95 S [Загружаем в аккумулятор адрес нулевого элемента массива]

[37]L 0 L [Сдвигаем его на 1 разряд влево, для того, чтобы загрузить в него инструкцию]

[38]T 1 S [Кладем его в первую ячейку]

[39]A 1 S [Загружаем в аккумулятор адрес нулевого элемента массива]

[40]A 92 [A 0 S] S [Прибавляем код инструкции с нулевым полем адреса]

[41]T 54 [r2] S [Запись сформированной инструкции]

[Цикл 1:-----]

[42]A 0 S [Загружаем число элементов массива]

[43]U 91 S [Обновляем число необработанных элементов во внутреннем цикле]

[44]S 88 S [Вычитаем значение счетчика]

[45]G 87 S [Если в аккумуляторе отрицательное значение, массив прочитан -> конец]

[46]T 1023 S [Очищаем аккумулятор]

[47]A 88 S [Загружаем значение счетчика в аккумулятор \ Увеличиваем]

[48]A 89 S [Прибавляем единицу в аккумулятор | значение счетчика]

[49]T 88 S [Загружаем увеличенный на 1 счетчик обратно / на единицу]

[50]A 89 S [Загружаем единицу в аккумулятор \]

[51]L 0 L [Сдвигаем на 1 разряд влево | Перемещаемся на следующий]

[52]A 54 S [Загружаем команду r2 в аккумулятор | элемент массива]

[53]T 54 S [Загружаем обновленную команду r2 /]

[54 <r2>]A 1 S [Загружаем в аккумулятор значения из ячейки N]

[55]S 89 S [Вычитаем 1]

[56]G 84 S [Если в аккумуляторе отрицательное значение, то число составное]

[57]T 1023 S [Очищаем аккумулятор]

[58]A 88 S [Загружаем значение счетчика в аккумулятор]

[59]U 96 S [Кладем его в Результат, так как значение массива >0]

[60]S 89 S [Вычитаем 1 \]

[61]L 0 L [Сдвигаем на 1 разряд влево | Формируем]

[62]A 1 S [Загружаем расположение 1 элемента массива в аккумулятор | текущий адрес]

[63]U 90 S [Загружаем получившееся значение в текущий адрес /]

[64]A 92 [A 0 S] S [Прибавляем код инструкции с нулевым полем адреса]

[65]T 73 [r1] S [Записываем получившуюся инструкцию]

[66]A 90 S [Загружаем адрес текущего элемента массива]

[67]A 93 [T 0 S] S [Прибавляем код инструкции с нулевым полем адреса]

[68]T 75 [w1] S [Записываем получившуюся инструкцию]

[Цикл2:-----]

[69]A 91 S [Загружаем число необработанных элементов во внутреннем цикле]

[70]S 88 S [Вычитаем значение счетчика]

[71]G 85 S [Если в аккумуляторе отрицательное значение, значит при данном значении счетчика массив обработан-> конец внутреннего цикла]

[72]T 91 S [Если значение не отрицательное, то загружаем измененное число необработанных элементов во внутреннем цикле]

[73 <r1>]A 1 S [Загружаем в аккумулятор значения из ячейки N \ Все элементы массива, индексы]

[74]S 89 S [Вычитаем 1 | которых кратны значению счетчика,]

[75 <w1>]T 0 S [Записываем это значения в ячейку N, обнуление аккумулятора / уменьшаются на единицу]

[76]A 88 S [Загружаем значение счетчика в аккумулятор \]

[77]L 0 L [Сдвигаем его на 1 разряд влево | Изменяем инструкцию r1]

[78]A 73 S [Прибавляем код инструкции, исполненной на предыдущем шаге | на значение счетчика]

[79]T 73 S [Записываем получившуюся инструкцию /]

[80]A 88 S [Загружаем значение счетчика в аккумулятор \]

[81]L 0 L [Сдвигаем его на 1 разряд влево | Изменяем инструкцию w1]

[82]A 75 S [Прибавляем код инструкции, исполненной на предыдущем шаге | на значение счетчика]

[83]T 75 S [Записываем получившуюся инструкцию /]

[84]E 69[Цикл2] S [Переход в начало внутреннего цикла]

[-----]

[85]T 1023 S [Очищаем аккумулятор]

[86]E 42[Цикл1] S [Переход в начало внешнего цикла]

[-----]

[87]Z S [Остановка программы]

[88]P 0 L [Счетчик]

[89]P 0 L [const = 1]

[90]P 0 S [Адрес текущего элемента в массиве]

[91]P 0 S [Число необработанных элементов во внутреннем цикле]

[92]A 0 S [Основа для формирования инструкций r1 и r2]

[93]T 0 S [Основа для формирования инструкции w1]

[94]P 5 S [Длина массива = 20]

[95]P 48 L [Адрес массива = 97]

[96]P 0 S [Результат]

[Массив, заполненный единицами:]

[97]P 0 L

[98]P 0 L

[99]P 0 L

[100]P 0 L

[101]P 0 L

[102]P 0 L

[103]P 0 L

[104]P 0 L

[105]P 0 L

[106]P 0 L

Руководство:

В ячейках [32] – [45] идет подготовка к работе цикла:

- Очистка аккумулятора перед работой (ячейка [33])
- Перенос длины исходного массива в нулевую ячейку (ячейки [34] – [35])
- Перенос и сдвиг адреса нулевого элемента в первую ячейку (ячейки [36] – [38])
- Формируем инструкцию r2, которая используется во внешнем цикле для загрузки следующего элемента массива и проверки, является ли соответствующее число простым (ячейки [39] – [41]).

Во внешнем цикле (ячейки [42] – [86]) происходит следующее:

1. Обновление числа необработанных элементов во внутреннем цикле (ячейка [43]).
2. Из числа элементов в массиве вычитается значение счетчика. Это значение сравнивается с нулем (если значение отрицательное, то цикл прерывается, поскольку весь массив уже прочитан) (ячейки [42] – [46]).
3. Увеличиваем значение счетчика на единицу (ячейки [47] – [49]).
4. Перемещаемся на следующий элемент массива при помощи сформированной инструкции r2 (ячейки [50] – [53]).
5. Проверяем на простоту число в массиве: вычитаем единицу и если значение отрицательное, значит число составное -> переходим к следующему элементу массива (ячейки [54] – [57]).
6. Иначе – загружаем значение счетчика в ячейку Результат (ячейки [58] – [59]).
7. Формируем текущий адрес и загружаем его в ячейку [90] (ячейки [58] – [63]).
8. Формируем инструкции r1 (ячейка [73]) и w1 (ячейка [75]), которые используются во внутреннем цикле для перемещения по массиву и уменьшения элементов массива на единицу соответственно (ячейки [58] – [68]).
9. По окончании внутреннего цикла в ячейке [86] находится команда «E 42 S», которая возвращает нас в начало внешнего цикла.

Во внутреннем цикле (ячейки [69] – [84]) происходит следующее:

1. Проверяем: обработан ли весь массив: из числа необработанных элементов вычитаем значение счетчика, и если в аккумуляторе отрицательное значение, значит при данном значении счетчика массив обработан -> прерываем внутренний цикл. Если же число не отрицательное -> обновляем число необработанных элементов (ячейки [69] – [72]).
2. Используя сформированные инструкции r1 и w1, загружаем в аккумулятор следующее значение массива и вычитаем из него единицу. Таким образом

все элементы массива, индексы которых кратны значению счетчика, уменьшаются на единицу (ячейки [73] – [75]).

3. Изменяем инструкции r1 и w1 на значение счетчика (ячейки [76] – [83]).
4. В конце цикла в ячейке [84] находится команда «E 69 S», которая возвращает нас в начало внутреннего цикла.

В конце содержатся счетчик, константа, адрес текущего элемента в массиве, число необработанных элементов во внутреннем цикле, основы для формирования инструкций, длина массива, адрес нулевого элемента массива и массив, заполненный единицами.

Программа, предполагающая загрузчик Initial Orders 2

Функциональность программы, предполагающей загрузчик Initial Orders 1 была выделена в подпрограмму.

В ячейках [111] – [120] располагается массив, заполненный единицами.

Проход по массиву осуществляется во внешнем цикле 1 в ячейках [49] – [93].

Внутренний цикл 2, в котором помечаем числа кратные встреченному простому числу, располагается в ячейках [76] – [91].

Результат работы программы располагается в ячейке [101].

Алгоритм работы:

Изначально все числа в массиве равны единице, то есть помечены как простые. Проходя по массиву и встречая простое число N, мы записываем его в Результат, а после заходим во внутренний цикл, в котором помечаем кратные N числа как составные. В конечном итоге результатом будет последнее встреченное простое число.

Текст программы:

Т 44 К [Директива - установка адреса загрузки]

[Подпрограмма:]

GK[Директива - фиксация начального адреса подпрограммы]

[0]A 3 F [Формирование инструкции возврата]

[1]T 50 @ [Запись сформированной инструкции]

[2]A 1 F [Загружаем в аккумулятор адрес нулевого элемента массива]

[3]A 55 @ [Прибавляем код инструкции с нулевым полем адреса]

[4]T 17 @ [Запись сформированной инструкции]

[Цикл 1:-----]

[5]A 0 F [Загружаем число элементов массива]

[6]U 54 @ [Обновляем число необработанных элементов во внутреннем цикле]

[7]S 51 @ [Вычитаем значение счетчика]

[8]G 50 @ [Если в аккумуляторе отрицательное значение, массив прочитан -> конец]

[9]T 1023 F [Очищаем аккумулятор]

[10]A 51 @ [Загружаем значение счетчика в аккумулятор \ Увеличиваем]

[11]A 52 @ [Прибавляем единицу в аккумулятор | значение счетчика]

[12]T 51 @ [Загружаем увеличенный на 1 счетчик обратно / на единицу]

[13]A 52 @ [Загружаем единицу в аккумулятор \]

[14]L 0 D [Сдвигаем на 1 разряд влево | Перемещаемся на следующий]

[15]A 17 @ [Загружаем команду r2 в аккумулятор | элемент массива]

[16]T 17 @ [Загружаем обновленную команду r2 /]

[17 <r2>] A 0 F [Загружаем в аккумулятор значения из ячейки N]

[18]S 52 @ [Вычитаем 1]

[19]G 48 @ [Если в аккумуляторе отрицательное значение, то число составное]

[20]T 1023 F [Очищаем аккумулятор]

[21]A 51 @ [Загружаем значение счетчика в аккумулятор]

[22]U 57 @ [Кладем его в Результат, так как значение массива >0]

[23]S 52 @ [Вычитаем 1 \]

[24]L 0 D [Сдвигаем на 1 разряд влево | Формируем]

[25]A 1 F [Загружаем расположение 1 элемента массива в аккумулятор | текущий адрес]

[26]U 53 @ [Загружаем получившееся значение в текущий адрес /]

[27]A 55 [A 0 F] @ [Прибавляем код инструкции с нулевым полем адреса]

[28]T 36 [r1] @ [Записываем получившуюся инструкцию]

[29]A 53 @ [Загружаем адрес текущего элемента массива]

[30]A 56 [T 0 F] @ [Прибавляем код инструкции с нулевым полем адреса]

[31]T 38 [w1] @ [Записываем получившуюся инструкцию]

[Цикл 2:-----]

[32]A 54 @ [Загружаем число необработанных элементов во внутреннем цикле]

[33]S 51 @ [Вычитаем значение счетчика]

[34]G 48 @ [Если в аккумуляторе отрицательное значение, значит при данном значении счетчика массив обработан-> конец внутреннего цикла]

[35]T 54 @ [Если значение не отрицательное, то загружаем измененное число необработанных элементов во внутреннем цикле]

[36 <r1>]A 0 F [Загружаем в аккумулятор значения из ячейки N \ Все элементы массива индексы]

[37]S 52 @ [Вычитаем 1 | которых кратны значению счетчика]

[38 <w1>]T 0 F [Записываем это значения в ячейку N, обнуление аккумулятора / уменьшаются на единицу]

[39]A 51 @ [Загружаем значение счетчика в аккумулятор \]

[40]L 0 D [Сдвигаем его на 1 разряд влево | Изменяем инструкцию r1]

[41]A 36 @ [Прибавляем код инструкции, исполненной на предыдущем шаге | на значение счетчика]

[42]T 36 @ [Записываем получившуюся инструкцию /]

[43]A 51 @ [Загружаем значение счетчика в аккумулятор \]

[44]L 0 D [Сдвигаем его на 1 разряд влево | Изменяем инструкцию r1]

[45]A 38 @ [Прибавляем код инструкции, исполненной на предыдущем шаге | на значение счетчика]

[46]T 38 @ [Записываем получившуюся инструкцию /]

[47]E 32[Цикл2] @ [Переход в начало внутреннего цикла]

[-----]

[48]T 1023 F [Очищаем аккумулятор]

[49]E 5[Цикл1] @ [Переход в начало внешнего цикла]

[-----]

[50]E 0 F [Инструкция возврата из подпрограммы]

[51]P 0 D [Счетчик]

[52]P 0 D [const = 1]

[53]P 0 F [Адрес текущего элемента в массиве]

[54]P 0 F [Число необработанных элементов во внутреннем цикле]

[55]A 0 F [Основа для формирования инструкций r1 и r2]

[56]T 0 F [Основа для формирования инструкции w1]

[57]P 0 F [Результат]

GK[Директива - фиксация начального адреса подпрограммы]

[0]Z 0 F [Точка останова для отладки]

[1]A 7 @ [Запись в аккумулятор длины массива]

[2]T 0 F [Запись длины массива в 1 ячейку]

[3]A 8 @ [Запись в аккумулятор адреса массива]

[4]T 1 F [Запись адреса массива в 0 ячейку]

[5]A 5 @ [Добавление в аккумулятор кода, необходимого для формирования инструкции возврата]

[6]G 44 F [Переход в подпрограмму]

[7]P 5 F [Длина = 10]

[8]P 9 @ [Адрес массива = 9]

[Массив, заполненный единицами:]

[9]P 0 D

[10]P 0 D

[11]P 0 D

[12]P 0 D

[13]P 0 D

[14]P 0 D

[15]P 0 D

[16]P 0 D

[17]P 0 D

[18]P 0 D

EZ PF

Руководство:

Никаких существенных изменений нет. Основной и внутренний рабочие циклы остались такими же, принцип их работы тот же. Главное отличие в относительной адресации.

Вывод:

В ходе данной лабораторной работы я познакомилась с принципом работы EDSAC и общими правилами реализации алгоритмов на ней на примере определения максимального простого числа, не превышающего заданное число N , методом решета Эратосфена.