**Задачи для самостоятельной работы (Машина Поста):**

**Задача1:** увеличить число 3 на единицу (изменить значение в памяти с 3 на 4). Целое положительное число на ленте машины Поста представимо идущими подряд метками, которых на одну больше, чем кодируемое число. Это связано с тем, что одна метка обозначает ноль, а уже две – единицу, и т.д. Каретка стоит слева от меток и обозревает пустую ячейку.

**Задача2:** Пусть задано исходное состояние каретки и требуется на пустой ленте написать две метки: одну в секцию под кареткой, вторую справа от нее.

**Задача3:.** На ленте имеется некоторое множество меток (общее количество меток не менее 1). Между метками множества могут быть пропуски, длина которых составляет одну ячейку. Заполнить все пропуски метками. Каретка находится слева под ячейкой с меткой.

**Задача4:**.На ленте проставлена метка в одной-единственной ячейке. Каретка стоит на некотором расстоянии левее этой ячейки. Необходимо подвести каретку к ячейке, стереть метку и остановить каретку слева от этой ячейки.

**Задача5:**На ленте задан массив меток. Увеличить длину массива на 2 метки. Каретка находится либо слева от массива, либо над одной из ячеек самого массива.

**Задача6:** На ленте на некотором расстоянии справа от каретки, стоящей под пустой клеткой, находится непрерывный массив меток. Требуется присоединить справа к массиву одну метку.

**Задача7:** На ленте имеется массив из *n* отмеченных ячеек. Каретка обозревает крайнюю левую метку. Справа от данного массива на расстоянии в *m* ячеек находится еще одна метка. Составьте для машины Поста программу, придвигающую данный массив к данной ячейке.

**Задача8:** Даны два массива меток, которые находятся на некотором расстоянии друг от друга. Требуется соединить их в один массив. Каретка находится над крайней левой меткой первого массива.

**Задача9:** На ленте заданы два массива — *m* и *n*, *m* > *n*. Вычислить разность этих массивов. Каретка располагается над левой ячейкой правого массива.

**Решение №1** *На доске разбирается пример.*

1 -> 2

2 ? 1;3

3 <- 4

4 V 5

5 !

- Исходное состояние:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | v | v | v | v |  |  |  |

↑

- Результат:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | v | v | v | v | v |  |  |  |

↑

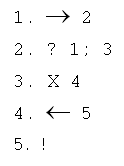
**Решение №2***:*

* 1. 1 v 2
  2. 2 → 3
  3. 3 v 4
  4. 4 !

**Решение №3**

1. → 2
2. ? 3, 1
3. → 4
4. ? 5, 6
5. !
6. ← 7
7. v 1

**Решение №4**



**Решение №5**1. ? 2; 3 (команды 1 и 2 — передвигаем каретку к массиву)

2. –> 1

3. –> 4 (команды 3 и 4 — передвигаем каретку к концу массива)

4. ? 5; 3

5. V 6 (команды 5–7 — ставим 2 метки в конце массива)

6. –> 7

7. V 8

8. !

**Решение №6** 1→2 2?1,3 3→4 4?5, 3 5v6 6!

**Решение №7** 1. X 2 (удаляем левую метку массива)

2. –> 3

3. ? 4; 2 (передвигаем каретку к концу массива)

4. V 5 (ставим справа от массива метку, раннее нами была удалена самая левая метка)

5. –> 6

6. ? 7; 10 (проверяем, передвинули ли мы уже наш массив к заданной метке)

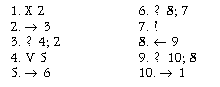
7. <– 8

8. ? 9; 7 (идем к левой метке массива)

9. –> 1 (и начинаем все сначала)

10. !

**Решение №8**



**Решение №9** Запишем решение алгоритма в словесной форме.

1. Ищем правый край массива *m*, двигаясь слева направо.

2. Стираем правую метку массива *m*.

3. Ищем правый край массива *n*, двигаясь слева направо.

4. Стираем левую метку массива *n*.

5. Проверяем, мы стерли последнюю метку в массиве *n* (в этом случае следующая справа ячейка должна быть пустой)?

6. Если стерли последнюю метку, то конец алгоритма.

7. Иначе ищем правый конец массива *m*, двигаясь справа налево.

8. Переход на шаг 2.

1. –> 2 (команды 1–3: ищем левую метку массива *m*)

2. ? 3; 1

3. <– 4

4. X 5 (стираем левую метку массива *m*)

5. ? 6; 7

6. –> 5

7. X 8 (стираем левую метку массива *n*)

8. –> 9

9. ? 12; 10 (стерли последнюю метку в массиве *n*?)

10. <– 11 (ищем левый край массива *m*)

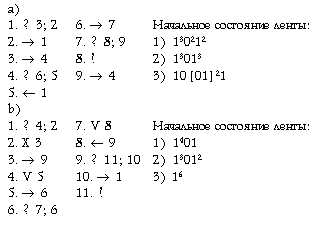
11. ? 10; 4

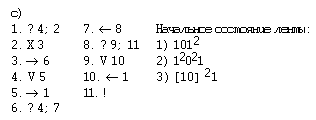
12. !

**Определение результата выполнения программ**

В задачах при описании начального состояния ленты будем указывать то, что записано начиная с самой левой непустой ячейки и заканчивая самой правой непустой ячейкой. При этом будем использовать следующие обозначения: *n* подряд идущих меток будем обозначать 1*n*, а *m* пустых ячеек — 0*m*. При обозначении одной заполненной или пустой ячейки будем писать просто 1 или 0, соответственно.

**1.** Выяснить, применимы ли программы к заданным состояниям машины Поста, указать результат работы машины Поста для каждого состояния.





*Ответы:*

a) 1) 1110011000

    2) зацикливание

    3) 1001011000

b) 1) зацикливание

    2) 010011

    3) 01010110

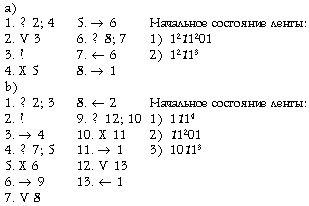
c) 1) зацикливание (…111)

    2) зацикливание (…1111001)

    3) зацикливание (1010111…)

**2.** Определить состояние, в котором окажется машина Поста в результате выполнения программы при заданном начальном состоянии ленты.

*Пояснение*: выделенная цифра, например ***1***, означает, что эту ячейку каретка обозревает в начальный момент времени.



*Решение*. Выделенная цифра показывает, на какой ячейке остановится машина.

a) 1) 11000000***1***

    2) 1100000***1***

b) 1) 11001***0***1

     2) 100***0***1

     3) (***1***11111?) 10100***1***1