Для работы нашего интерпритатора мы схраним два массива: array, command. Command хранит в себе входной набор чередующихся 0 и 1, array же хранит массив, в который записываются данные. Кроме того мы храним два указателя: current\_array, который указывает на текущий элемент в массиве и current command, который указывает на текущую ячейку с коммандой. И еще имеем список result\_te, в который мы каждый раз при встрече открывающуюся скобку добавляем текущую позицию corrent\_command, из которого, при встрече закрывающейся скобки, извлекается эта позиция и, если неоходимо повторить цикл, то возвращается к ней. Прохождение циклов устроено следующим образом: когда мы встречаем открывающуюся скобку и нам необходимо войти в цикл, то мы просто продолжаем работу программы (do = True), если необходимо не входить в цикл, то мы проходим по всем комндам до закрывающейся соответсвующей скобки, не исполняя команды (do = False) (нужной закрывающуюся скобку мы находим по переменной count, которая хранит разность между количеством уже пройденных открывающихся и закрывающихся скобок). Если мы дошли до конца цикла, исполняя его, то, если необходимо войти в него снова, то возвращаемся к комманде соответствующей открывающейся скобки, если же не нужно, то просто продолжаем исполнять команды. В результате мы проходимся по всем коммандам из command от первой до последней (иногда с повторением), которая получает по веткам if соответсвующие команды и исполняет их, только если do == True:

```
if command[current_command] == '1': # '1'
    значение текущей ячейки массива с данными увеличиваем на 1
else:
    переход к следующей ячейке в команде
    if command[current command] == '1': # '01'
         переход к следующей ячейке в команде
        if command[current_command] == '1': # '011
             сдвигаем указатель current_array на одну ячейку массива с данными назад
        else: # '010'
             добавляем в конец массива с данными элемент 0
             двигаем указатель current_array на одну ячейку массива с данными вперед
    else: # '00'
        переход к следующей ячейке в команде
        if command[current command] == '0': # '000'
             значение текущей ячейки массива с данными уменьшаем на 1
        else: # '001'
             переход к следующей ячейке в команде
             if command[current_command] == '1': # '0011'
                 извлекаем из return te позицию начала соответсвующего цикла
                 если мы сейчас в исполняемой части программы и
                   int(array[current_array]) > 0, то переходим к началу цикла и
                   исполняем внутренний цикл
                  если же мы не в исполняемой части, то уменьшаем число count на 1,
                   если оно обнулилось, то мы вышли из неисполняемой части, поэтому
                   делаем do = True
             else: # '0010'
                 переход к следующей ячейке в команде
                 if command[current command] == '0': # '00100'
                      добавляем в return te текущую позицию
                     если мы сейчас в исполняемой части программы и
                          int(array[current_array]) > 0, то исполняем внутренний цикл, д
                          делая do = True, иначе проходим насквозь do = False
                      если же мы не в исполняемой части, то увеличиваем число count на 1
                 else:
                      переход к следующей ячейке в команде
                      if command[current command] == '0': # '001010'
                          печатаем текущую ячейку массива с данными
                      else: # '001011'
                          переход к следующей ячейке в команде # '0010110'
                          вводим знчение в текущую ячейку массива с данными, если оно
                          есть на входе, иначе обнуляем его
```