

1. ПОСЛЕДОВАТЕЛЬНОСТЬ

20 очков

10 секунд

Рассмотрим последовательность, которая составляется следующим образом: начинают с 1-элементной последовательности, единственный элемент которой 0; на каждом следующем шагу берётся имеющаяся последовательность $a_1..a_N$, вычисляется её дополнение $b_1..b_N$ (где b_i равен 1, если a_i равен 0 и наоборот) и добавляется в конец последовательности. Очевидно длина последовательности возрастает на каждом шагу в два раза.

Первые шаги создания последовательности выглядят так:

```

0
0 1
0 1 1 0
0 1 1 0 1 0 0 1
0 1 1 0 1 0 0 1 1 0 0 1 0 1 1 0

```

Написать программу для нахождения элементов последовательности.

Входные данные. На первой строке файла JADA.SIS расположены два разделённых пробелом числа N и M ($1 \leq N \leq 1\ 000\ 000\ 000$, $1 \leq M \leq 100$) – позиция первого элемента и длина искомой подпоследовательности.

Выходные данные. На первую строку файла JADA.VAL вывести M элементов искомой подпоследовательности разделённые пробелами.

Пример.

JADA.SIS	JADA.VAL
4 6	0 1 0 0 1 1

2. ВОЛНА

30 очков

10 секунд

Для обработки сигнала в компьютере периодически измеряют его уровень и представляют полученный результат в виде последовательности a_1, a_2, \dots, a_N . Возрастающим фронтом называют подпоследовательность $a_{i_1}..a_{i_2}$, в которой значения сигнала не убывают ($a_{i_1} \leq a_{i_1+1} \leq \dots \leq a_{i_2}$), падающим фронтом называют подпоследовательность $a_{i_1}..a_{i_2}$, в которой значения сигнала не возрастают. Выпуклой волной называют подпоследовательность $a_{i_1}..a_{i_2}..a_{i_3}$, где $a_{i_1}..a_{i_2}$ – возрастающий, а $a_{i_2}..a_{i_3}$ падающий фронт ($a_{i_1} \leq a_{i_1+1} \leq \dots \leq a_{i_2} \geq a_{i_2+1} \geq \dots \geq a_{i_3}$), впуклой волной называют подпоследовательность состоящую из падающего и возрастающего фронтов.

Написать программу, которая находит в заданном сигнале самые длинные возрастающий и спадающий фронт, а также выпуклую и впуклую волны (длиной подпоследовательности считать количество элементов в ней).

Входные данные. На первой строке файла LAINE.SIS записано количество элементов в последовательности N ($1 \leq N \leq 1000$) и на следующих N строках целочисленные значения элементов ($-1000 \leq a_i \leq 1000$).

Выходные данные. На первую строку файла LAINE.VAL вывести длины самых длинных возрастающего и падающего фронтов разделённые пробелом, на вторую строчку длины самых длинных выпуклой и впуклой волн также разделённые пробелом.

Пример.

LAINE.SIS	LAINE.VAL
6	3 3
1	5 4
2	
3	
2	
1	
2	

Оценивание. В этом задании длины фронтов и волн оценивается отдельно. Если программа не решает одну из двух подзадач (длины фронтов или волн), то вывести в соответствующую строчку два –1 разделённые пробелом.

3. ТАЙНЫЙ ЯЗЫК

30 очков

10 секунд

Юля и Рита пользуются тайным языком, в котором каждое слово кодируется отдельно: слово разделяют на две части, которые обмениваются местами. Например фразу “homme kohtume pargis” можно закодировать “mehom tumekoh isparg” или “mmeho ohtumek gisparg” или ещё 118 способами.

Написать программу, которая подсчитывает сколько из фраз от S_1 до S_N могут быть кодом фразы S . Требуется чтобы при кодировании менялось каждое слово.

Входные данные. На первой строке файла SALA.SIS записана фраза S . На второй строке файла записано число N ($1 \leq N \leq 100$) и на следующих N строчках - фразы S_1, S_2, \dots, S_N . Все фразы состоят из маленьких латинских букв и пробелов, длина фраз не превышает 80 символов.

Выходные данные. Вывести в файл SALA.VAL количество фраз которые являются кодом фразы S .

<u>Пример.</u>	SALA.SIS	SALA.VAL	Пояснения
	kohtume pargis	2	
	5		
	tumekoh pargis		второе слово не изменилось
	mekohtu gisparg		порядок
	tukohme isparg		первое слово разбито на три части
	tumekoh argisp		порядок
	umekoht gaspar		во втором слове изменена буква

4. НОВОСТИ

40 очков

10 секунд

В новостных группах Интернета у каждого сообщения есть уникальный идентификатор. Если письмо является ответом на другое письмо, то оно также содержит идентификатор того письма. Пользователь хочет видеть содержание группы в виде дерева, где заголовок каждого ответа на письмо находится под заголовком исходного письма и сдвинут вправо на один пробел.

Например, если на сообщение с заголовком “aaa” ответили письмом с заголовком “bbb” и на это сообщение ответили письмами с заголовками “ccc” ja “ddd”, то в результате дерево должно было бы выглядеть так:

```
aaa
  bbb
    ccc
    ddd
```

Написать программу для составления такого дерева.

Входные данные. На первой строке файла UUDIS.SIS записано число сообщений в группе N ($1 \leq N \leq 1000$) и на следующих $2 \cdot N$ строках расположены 2-строчные описания сообщений: на первой строке расположен идентификатор письма (положительное число, не более 5 знаков) и идентификатор исходного письма, если такой есть, иначе -1; на второй строке записан заголовок сообщения (до 40 символов).

Выходные данные. Вывести в файл UUDIS.VAL заголовки всех сообщений в виде дерева. На каждом уровне иерархии упорядочить заголовки в порядке возрастания идентификаторов.

<u>Пример.</u>	UUDIS.SIS	UUDIS.VAL
	5	a
	1 -1	b
	a	c
	2 4	e
	e	d
	4 -1	
	c	
	5 4	
	d	
	3 1	
	b	

5. АЛГОРИТМ

30 очков

Рассмотрите предложенную программу записанную на псевдоязыке (псевдоязык – это упрощённый для понимания язык программирования) и попытайтесь понять, что эта программа делает.

```

1.  input N
2.  for i := 1 .. N
3.      input A[i]
4.  end for
5.  F := true
6.  for i := 1 .. N-1
7.      if A[i] > A[i + 1]
8.          x := A[i]
9.          A[i] := A[i + 1]
10.         A[i + 1] := x
11.         F := false
12.     end if
13. end for
14. if F == false
15.     goto 5
16. end if
17. print N
18. for i := 1 .. N
19.     print A[i]
20. end for

```

Решением данной задачи должен быть текстовый файл, в котором записаны ответы на следующие вопросы. Текст должен быть на эстонском или русском языках:

1. Сформулируйте задачу, которую решает эта программа (2-3 предложения).
2. Обоснуйте, что если эта программа заканчивает свою работу, то результат соответствует ответу на предыдущий вопрос. Достаточно нескольких предложений.
3. Обоснуйте, что эта программа заканчивает свою работу при всех корректных входных данных. Достаточно нескольких предложений.
4. Как минимум сколько раз выполняется 5-ая строка программы (F:=true)? Ответ должен быть формулой зависящей от N. При каких входных данных эта строка выполняется именно такое количество раз?
5. Как максимум сколько раз выполняется 5-ая строка программы (F:=true)? Ответ должен быть формулой зависящей от N. При каких входных данных эта строка выполняется именно такое количество раз?
6. Как максимум сколько раз выполняется 11-ая строка программы (F:=false)? Ответ должен быть формулой зависящей от N. При каких входных данных эта строка выполняется именно такое количество раз?