QUEUE

1. Min của xâu

Cho xâu ký tự S. Ta gọi giá trị của xâu S là tổng bình phương số lần xuất hiện mỗi ký tự trong S. Hãy tìm giá trị nhỏ nhất của xâu S sau khi thực hiện K lần loại bỏ ký tự.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất là số K; phần thứ hai là một xâu ký tự S được viết trên một dòng.
* T, S, K thỏa mãn ràng buộc: 1≤T≤100;  1≤length(S)≤10000; 1≤K≤1000.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 2  ABCCBC 2  AAAB | 6 2 |

1. Số nhị phân từ 1 đến N

Cho số tự nhiên n. Hãy in ra tất cả các số nhị phân từ 1 đến n.

**Input:**

* Dòng đầu tiên ghi lại số lượng test T (T≤100).
* Mỗi test là một số tự nhiên n được ghi trên một dòng (n≤10000).

**Output:**

* Đưa ra kết quả mỗi test trên một dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  2  5 | 1 10  1 10 11 100 101 |

1. BDN 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ta gọi số nguyên dương K là một số BDN nếu các chữ số trong K chỉ bao gồm các 0 hoặc 1 có nghĩa. Ví dụ số K = 1, 10, 101. Cho số tự nhiên N (N<231). Hãy cho biết có bao nhiêu số BDN không lớn hơn N. Ví dụ N=100 ta có 4 số BDN bao gồm các số: 1, 10, 11, 100.  **Dữ liệu vào:**Dòng đầu tiên ghi lại số tự nhiên T (T ≤ 50) là số lượng bộ test.  T dòng kế tiếp mỗi dòng là một số tự nhiên N (1 ≤ N ≤ 231).  **Kết quả:**Với mỗi số N in ra kết quả trên một dòng là số các số BDN không lớn hơn N. | |  |  | | --- | --- | | **Input** | **Output** | | 3  10  100  200 | 2  4  7 | |

1. BDN 2

Ta gọi số nguyên dương K là một số BDN nếu các chữ số trong K chỉ bao gồm các 0 hoặc 1 có nghĩa. Ví dụ số K =  101 là số BDN, k=102 không phải là số BDN.

Số BDN của N là số P =M´N sao cho P là số BDN. Cho số tự nhiên N (N<1000), hãy tìm số BDN nhỏ nhất của N.

**Ví dụ**. Với N=2, ta tìm được số BDN của N là P = 5´2=10. N = 17 ta tìm được số BDN của 17 là P = 653´17=11101.

**Input:**

* Dòng đầu tiên ghi lại số tự nhiên T là số lượng Test;
* T dòng kế tiếp mỗi dòng ghi lại một bộ Test. Mỗi test là một số tự nhiên N.

**Output:**

                        Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  2  12  17 | 10  11100  11101 |

1. Số 0 và 9

Cho số tự nhiên N. Hãy tìm số nguyên dương X nhỏ nhất được tạo bởi số 9 và số 0 chia hết cho N. Ví dụ với N = 5 ta sẽ tìm ra  X = 90.

**Input:**

* Dòng đầu tiên ghi lại số lượng test T (T≤100).
* Những dòng kế tiếp mỗi dòng ghi lại một test. Mỗi test là một số tự nhiên N được ghi trên một dòng (N≤100).

**Output:**

* Đưa ra theo từng dòng số X nhỏ nhất chia hết cho N tìm được .

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  5  7 | 90  9009 |

1. Biến đổi ST

Cho hai số nguyên dương S và T (S, T<10000) và hai thao tác (a), (b) dưới đây:

**Thao tác (a)**: Trừ S đi 1  (S = S-1) ;

**Thao tác (b)**: Nhân S với 2 ( S = S\*2);

Hãy dịch chuyển S thành T sao cho số lần thực hiện các thao tác (a), (b) là ít nhất. Ví dụ với    S =2, T=5 thì số các bước ít nhất để dịch chuyển S thành T thông qua 4 thao tác sau:

**Thao tác (a)**: 2\*2 = 4;

**Thao tác (b)**: 4-1 = 3;

**Thao tác (a)**: 3\*2 = 6;

**Thao tác (b)**: 6-1 = 5;

**Input:**

* Dòng đầu tiên ghi lại số tự nhiên T là số lượng Test;
* T dòng kế tiếp mỗi dòng ghi lại một bộ Test. Mỗi test là một bộ đôi S và T.

**Output:** Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  2 5  3 7  7 4 | 4  4  3 |

1. Số lộc phát đối xứng chẵn

Trong quan điểm người Việt, một số nguyên dương được gọi là “lộc phát” nếu chỉ có hai chữ số là 6 và/hoặc 8. Một số lộc phát X được gọi là đối xứng chẵn nếu:

* Số chữ số của X là chẵn
* X thỏa mãn tính chất đối xứng.

Hãy liệt kê N số lộc phát đối xứng chẵn đầu tiên theo thứ tự từ nhỏ đến lớn.

**Input**

* Dòng đầu tiên đưa vào số lượng test T.
* Dòng tiếp theo đưa vào các bộ test. Mỗi bộ test được là một số N được viết trên một dòng.
* T, N thỏa mãn ràng buộc: 1≤T≤100; 1≤N≤104.

**Output**

Đưa ra kết quả mỗi test theo từng dòng. Các số cách nhau đúng một khoảng trống.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  4  10 | 66 88 6666 6886  66 88 6666 6886 8668 8888 666666 668866 686686 688886 |

1. Lộc phát\_QUEUE

Một số được gọi là lộc phát nếu chỉ có 2 chữ số 6 và 8. Cho số tự nhiên N. Hãy liệt kê các số lộc phát có không quá N chữ số.

**Input:**

* Dòng đầu tiên ghi lại số tự nhiên T là số lượng bộ test (T<10);
* T dòng kế tiếp mỗi dòng ghi số N (1<N<15).

**Output:**

* Dòng đầu tiên là số lượng số lộc phát tìm được. Dòng thứ hai in ra đáp án **theo thứ tự giảm dần**.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  2  3 | 6  88 86 68 66 8 6  14  888 886 868 866 688 686 668 666 88 86 68 66 8 6 |

1. Số nguyên thủy

Cho số nguyên N. Nhiệm vụ của bạn hãy đưa ra N số nguyên thủy đầu tiên theo thứ tự từ nhỏ đến lớn.

Số K được gọi là số nguyên thủy nếu số đó thỏa mãn tất cả các điều kiện:

* Số các chữ số của K là một số chẵn.
* Tất cả các chữ số của K chỉ bao gồm số 4 hoặc 5.
* K là một số đối xứng.

**Input:**

* Dòng đầu tiên đưa vào số lượng test T.
* Dòng tiếp theo đưa vào các bộ test. Mỗi bộ test được là một số N được viết trên một dòng.
* T, N thỏa mãn ràng buộc: 1≤T≤100; 1≤N≤104.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  4  10 | 44 55 4444 4554  44 55 4444 4554 5445 5555 444444 445544 454454 455554 |

STACK

1. Biểu thức đúng

Cho một mảng S gồm 2×N ký tự, trong đó có N ký tự ‘[’ và N ký tự ‘]’. Xâu S được gọi là viết đúng nếu S có dạng S2[S1] trong đó S, S2 là các xâu viết đúng. Nhiệm vụ của bạn là tìm số các phép đổi chỗ ít nhất các ký tự kề nhau của xâu S viết sai để  S trở thành viết đúng. Ví dụ với xâu S =”[]][][” ta có số phép đổi chỗ kề nhau ít nhất là 2.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một xâu S viết sai theo nguyên tắc kể trên.
* T, S thòa mãn ràng buộc: 1≤T≤100; 1≤length(S)≤100000.

**Output:**

* Đưa kết quả trên một dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  []][][  [][][] | 2 0 |

1. Đảo từ

Cho một xâu ký tự str bao gồm nhiều từ trong xâu. Hãy đảo ngược từng từ trong xâu?

**Input**:

* Dòng đầu tiên đưa vào số lượng bộ test T;
* Những dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một dòng ghi lại nhiều từ trong xâu str.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

**Ràng buộc**:

* T, str thỏa mãn ràng buộc: 1≤T≤100; 2≤length(str)≤106.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  ABC DEF  123 456 | CBA FED  321 654 |

1. Kiểm tra dãy ngoặc đúng

Cho một xâu chỉ gồm các kí tự ‘(‘, ‘)’, ‘[‘, ‘]’, ‘{‘, ‘}’. Một dãy ngoặc đúng được định nghĩa như sau:

-     Xâu rỗng là 1 dãy ngoặc đúng.

-     Nếu A là 1 dãy ngoặc đúng thì (A), [A], {A} là 1 dãy ngoặc đúng.

-     Nếu A và B là 2 dãy ngoặc đúng thì AB là 1 dãy ngoặc đúng.

Cho một xâu S. Nhiệm vụ của bạn là xác định xâu S có là dãy ngoặc đúng hay không?

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 20).

Mỗi test gồm 1 xâu S có độ dài không vượt quá 100 000.

**Output:**

Với mỗi test, in ra “YES” nếu như S là dãy ngoặc đúng, in ra “NO” trong trường hợp ngược lại.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output** |
| 2  [()]{}{[()()]()}  [(]) | YES  NO |

1. Dãy ngoặc đúng dài nhất

Cho một xâu chỉ gồm các kí tự ‘(‘ và ‘)’. Một dãy ngoặc đúng được định nghĩa như sau:

-     Xâu rỗng là 1 dãy ngoặc đúng.

-     Nếu A là 1 dãy ngoặc đúng thì (A) là 1 dãy ngoặc đúng.

-     Nếu A và B là 2 dãy ngoặc đúng thì AB là 1 dãy ngoặc đúng.

Cho một xâu S. Nhiệm vụ của bạn là hãy tìm dãy ngoặc đúng dài nhất xuất hiện trong xâu đã cho.

**Input:**Dòng đầu tiên là số lượng bộ test T (T ≤ 20).

Mỗi test gồm một xâu S có độ dài không vượt quá 105 kí tự.

**Output:**Với mỗi test in ra một số nguyên là độ dài dãy ngoặc đúng dài nhất tìm được.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output** |
| 3  ((()  )()())  ()(())))) | 2  4  6 |

1. Kiểm tra biểu thức số học

Cho biểu thức số học, hãy cho biết biểu thức số học có dư thừa các cặp ký hiệu ‘(’,’) ‘ hay không?

**Input**:

* Dòng đầu tiên đưa vào số lượng bộ test T;
* Những dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một biểu thức tiền tố exp.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

**Ràng buộc**:

* T, exp thỏa mãn ràng buộc: 1≤T≤100; 2≤length(exp)≤20.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  ((a+b))  (a + (b)/c)  (a + b\*(c-d)) | Yes  Yes  No |

1. Sửa lại dãy ngoặc

Cho một xâu chỉ gồm các kí tự ‘(‘, ‘) và có độ dài chẵn. Hãy đếm số lượng dấu ngoặc cần phải đổi chiều ít nhất, sao cho xâu mới thu được là một dãy ngoặc đúng.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 20).

Mỗi test gồm 1 xâu S có độ dài không vượt quá 100 000, chỉ gồm dấu ( và ).

**Output:**

Với mỗi test, in ra đáp án tìm được trên một dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input: | Output |
| 4  ))((  ((((  (((())  )(())((( | 2  2  1  3 |

1. Phần tử đầu tiên bên phải lớn hơn

Cho dãy số A[] gồm N phần tử. Với mỗi A[i], bạn cần tìm phần tử bên phải đầu tiên lớn hơn nó. Nếu không tồn tại, in ra -1.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 20).

Mỗi test bắt đầu bởi số nguyên N (1 ≤ N ≤ 100000).

Dòng tiếp theo gồm N số nguyên A[i] (0 ≤ A[i] ≤ 109).

**Output:**

Với mỗi test, in ra trên một dòng N số R[i], với R[i] là giá trị phần tử đầu tiên lớn hơn A[i].

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  4  4 5 2 25  3  2 2 2  4  4 4 5 5 | 5 25 25 -1  -1 -1 -1  5 5 -1 -1 |

1. Đánh dấu số thứ tự cặp dấu ngoặc

Cho một biểu thức đúng, tức là các dấu ngoặc đơn đều đầy đủ mở và đóng, đảm bảo đúng thứ tự. Hãy viết chương trình đánh số các cặp dấu ngoặc theo thứ tự xuất hiện, tính từ 1.

Ví dụ với biểu thức                                              (a + (b \*c) ) + (d/e)

ta có thứ tự của các cặp ‘(‘, ‘)’ là                         1  2  2  1  3  3

**Input**:

* Dòng đầu tiên đưa vào số lượng bộ test T (không quá 100).
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một biểu thức số học được đưa vào trên một dòng, độ dài không quá 106.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  (a + (b \*c) ) + (d/e)  ( ( () ) ( () ) ) | 1  2  2  1  3  3  1 2  3  3  2  4  5  5  4  1 |

1. Số lộc phát\_STACK

Một số được gọi là lộc phát nếu chỉ có 2 chữ số 6 và 8. Cho số tự nhiên N. Hãy liệt kê các số lộc phát có không quá N chữ số.

**Input:**

* Dòng đầu tiên ghi lại số tự nhiên T là số lượng bộ test (T<10);
* T dòng kế tiếp mỗi dòng ghi số N (1<N<15).

**Output:**

* Dòng đầu tiên là số lượng số lộc phát tìm được. Dòng thứ hai in ra đáp án **theo thứ tự giảm dần**.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  2  3 | 6  88 86 68 66 8 6  14  888 886 868 866 688 686 668 666 88 86 68 66 8 6 |

1. Kiểm tra câu viết đúng quy tắc

Cho một câu văn bản kết thúc với dấu chấm, trong đó chỉ có các chữ cái Tiếng Anh và các dấu ngoặc kiểu ngoặc đơn (,) hoặc ngoặc vuông [,].

Nếu chỉ xét các cặp dấu ngoặc, bạn hãy kiểm tra xem câu văn bản trên có viết đúng quy tắc hay không. Tức là các cặp dấu ngoặc đều viết  mở và đóng đẩy đủ, đúng thứ tự.

**Input**

Dòng đầu ghi số bộ test (không quá 4000). Mỗi test viết trên một dòng câu văn bản có độ dài không quá 100, luôn kết thúc bởi dấu chấm.

**Output**

Ghi ra YES hoặc NO tùy thuộc kết quả kiểm tra.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  [ graph ] ( binary tree ).  Dynamic programming )( devide and conquer.  ([ (([( [ ] ) ( ) (( ))] )) ]). | YES  NO  YES |