Các câu hỏi có điểm lần lượt 3, 3, 3, 1.

Đề 1

Bài 1.

Cho một dãy số, một chuỗi con không giảm là chuỗi các phần tử liên tiếp của dãy số sao cho số đứng sau không nhỏ hơn số đứng trước nó. Mỗi chuỗi cần kéo dài hết mức có thể. Ví dụ, dãy số 10 10 12 4 có hai chuỗi con không giảm là 10 10 12 (xuất phát từ chỉ số 0) và 4 (xuất phát từ chỉ số 3).

Viết chương trình nhập vào một số nguyên dương N và một chuỗi gồm N số nguyên N ≤10000000 và in ra chỉ số của phần tử đầu của các chuỗi con không giảm của dãy số đó theo thứ tự xuất hiện của các chuỗi đó.

Đầu vào:

- Dòng đầu tiên chứa số nguyên dương N.
- Dòng thứ hai chứa N số nguyên cách nhau bởi dấu cách.

Đầu ra:

- Chỉ số đầu của các chuỗi con không giảm theo thứ tự xuất hiện của các chuỗi con đó trong dãy số đã cho. Chú ý: dãy N số bắt đầu từ chỉ số 0.

For example:

Input	Result
4 10 10 12 4	0 3

Bài 2:

Cho một struct Node biểu diễn một nodenode của 1 danh sách liên kết đơn như sau:

```
struct Node {
   int value;
   Node *next;
};
```

Biết **head** là con trỏ trỏ tới một danh sách liên kết có tối thiểu 2 phần tử, trong đó có tối đa một node **đáy** có giá trị nhỏ hơn một nửa giá trị của nút nhỏ hơn trong các nút bên cạnh, (node đứng đầu hoặc cuối dãy có thể là đáy nếu có nó giá trị nhỏ hơn một nửa giá trị của nút bên cạnh).

KHÔNG sử dụng mảng phụ, viết hàm Node* deleteBottom(Node* head); xoá node đáy khỏi danh sách liên kết (nếu có). Hàm trả về con trỏ trỏ tới vị trí đầu tiên của danh sách liên kết sau khi đã xoá.

For example:

Input	Result
5 3 2 0 4 4	3 2 4 4

Bài 3:

Cho một lưới vuông gồm các ô có giá trị 0 hoặc 1. Hai ô số 0 kề cạnh được coi là liên thông nhau. Hai ô số 0 được coi là liên thông nhau nếu chúng cùng liên thông với một ô thứ ba. Một vùng liên thông 0 là vùng các ô số 0 liên thông nhau. Trong test ví dụ dưới đây, lưới vuông có ba vùng liên thông với diện tích 1, 2 và 3. Bạn cần viết chương trình đọc vào một lưới vuông và tính diện tích vùng liên thông lớn nhất. Nếu không có vùng liên thông 0 nào thì đáp số là 0.

Đầu vào:

- Dòng đầu tiên chứa số nguyên dương N. N≤100
- NN dòng sau, mỗi dòng chứa N số nguyên 1 hoặc 0, cách nhau bởi dấu cách.

Đầu ra:

- Diện tích của vùng liên thông 0 lớn nhất tính theo số ô.

For example:

Input	Result
4 0 1 1 1 0 1 0 0 1 1 0 1 0 1 1 1	3

Bài 4:

Cho danh sách gồm N chữ cái thường. In ra các bộ gồm đúng K chữ cái khác nhau theo thứ tự giảm dần, nếu lớn hơn 100 bộ thì in ra 100 bộ đầu tiên.

Đầu vào:

Dòng đầu chứa 2 số nguyên: N, K

Dòng thứ 2 chứa N chữ cái.

Đầu ra:

Các bộ gồm K chữ cái theo thứ tự giảm dần.

Ví dụ:

Input	Output
7 3 a b d c e f g h	h g f h g e …

Đề 2

Bài 1.

Cho một dãy số, một chuỗi con chẵn là chuỗi các phần tử liên tiếp toàn số chẵn của dãy số, một chuỗi con lẻ là chuỗi các phần tử liên tiếp toàn số lẻ của dãy số. Mỗi chuỗi cần kéo dài hết mức có thể. Ví dụ, dãy số 10 12 10 1 có một chuỗi con chẵn là 10 12 10 (độ dài 3), và một chuỗi con lẻ 1 (độ dài 1).

Viết chương trình nhập vào một số nguyên dương N và một chuỗi gồm N số nguyên N≤10000000 và in ra độ dài các chuỗi con chẵn hoặc lẻ của dãy số đó theo thứ tự xuất hiện của các chuỗi đó.

Đầu vào:

- Dòng đầu tiên chứa số nguyên dương N.
- Dòng thứ hai chứa N số nguyên cách nhau bởi dấu cách.

Đầu ra:

- Độ dài của các chuỗi con chẵn hoặc lẻ theo thứ tự xuất hiện của các chuỗi con đó trong dãy số đã cho.

For example:

Input	**Result**
4 10 12 10 1	3 1

Bài 2: Cho một struct Node biểu diễn một nodenode của 1 danh sách liên kết đơn như sau:

```
struct Node {
   int value;
   Node *next;
```

};

Biết **head** là con trỏ trỏ tới một danh sách liên kết có tối thiểu 2 phần tử, trong đó có tối đa một node **đỉnh** có giá trị lớn hơn hai lần tổng các nút bên cạnh, (node đứng đầu hoặc cuối dãy thì chỉ có một nút bên cạnh nhưng vẫn có thể là đỉnh).

KHÔNG sử dụng mảng phụ, viết hàm Node* deleteTop(Node* head); xoá node đỉnh khỏi danh sách liên kết (nếu có). Hàm trả về con trỏ trỏ tới vị trí đầu tiên của danh sách liên kết sau khi đã xoá.

For example:

Input	Result
5 3 2 13 4 4	3 2 4 4

Bài 3:

Cho một string đã được mã hóa cho các ký tự lặp bằng cách thêm số lần lặp vào trước ký tự, viết hàm hàmstring deCode(string str) chuyển đổi string này về dạng gốc của nó.

VD: "el4ev2al" => "eleeeevaal"

Lưu ý: string dạng gốc không có số.

For example:

Test	Input	Result
string s; cin >> s; cout << deCode(s);	4a6b	aaaabbbbbb
string s; cin >> s; cout << deCode(s);	afriend3.	afriend...

Bài 4

Cho danh sách gồm N chữ cái thường. In ra các bộ gồm đúng K chữ cái khác nhau theo thứ tự tăng dần, nếu lớn hơn 100 bộ thì in ra 100 bộ đầu tiên.

Đầu vào:

Dòng đầu chứa 2 số nguyên: N, K

Dòng thứ 2 chứa N chữ cái.

Đầu ra:

Các bộ gồm K chữ cái theo thứ tự tăng dần.

Ví dụ:

Input	Output
8 8 a b d c e f g h	a b c d e f g h a b c d e f h g a b c d e g f h a b c d e g h f

Đề 3

Bài 1:

Đếm các số nguyên tố trong khoảng từ số L đến số R.

Đầu vào:

1 dòng duy nhất chứa 2 số nguyên L, R.

Đầu ra:

In ra màn hình 1 số nguyên duy nhất là số số nguyên tố trong khoảng L đến R

Ví dụ:

Input	Output
5 10	2 Giải thích: có 2 số là số nguyên tố: 5, 7
10 20	4 Giải thích: có 4 số là số nguyên tố: 11 13 17 19

Bài 2:

Cho một struct Node biểu diễn một $node$ của 1 danh sách liên kết đơn như dưới đây, với head là con trỏ tới nút đầu tiên, tail là con trỏ tới nút cuối cùng, cả hai đều nhận giá trị null nếu danh sách rỗng:

```
struct Node {
    Node* next;
    int data;
};
struct LinkedList {
```

```
    Node *head, *tail;
    LinkedList* createEvenList();
};
```

Viết một hàm thành viên **LinkedList* createEvenList()** với nhiệm vụ tạo d một danh sách mới với nội dung lần lượt là các nút có thứ tự chẵn (thứ hai, thứ tư, thứ sáu, …) của danh sách gốc **Hàm sau đó trả về địa chỉ của danh sách mới**, trong khi không được thay đổi dữ liệu của danh sách gốc.

```
LinkedList* LinkedList::createEvenList() {.....} ;
```

Ví dụ:

Input	**Result**
5 1 2 3 4 5	2 4 1 2 3 4 5
10 320 260 357 453 375 110 289 143 253 385	260 453 110 143 385 320 260 357 453 375 110 289 143 253 385

Bài 3:

Cho một lưới kích thước [m,n] và Q điểm giá trị trên ô lưới đấy. K hình chữ nhật có ô trên trái là [x_k, y_k] và kích thước là [w_k, h_k] được phủ lên lưới đã cho. Các ô trên lưới được đánh các số trong khoảng [0, 100]. Tìm trong K hình chữ nhật, hình chữ nhật có tổng các ô nó bao phủ có giá trị lớn nhất.

Đầu vào:

Dòng đầu chứa 4 số nguyên M, N, Q, K

Q dòng tiếp theo, mỗi dòng chứa 3 số nguyên x_q, y_q , v_q tương ứng lần lượt là vị trí và giá trị.

K dòng tiếp theo, mỗi dòng chứa 4 số nguyên x_k, y_k , w_k, h_k tương ứng lần lượt là ô trên trái và kích thước của hình chữ nhật.

Đầu ra:

1 số duy nhất là giá trị lớn nhất mà hình chữ nhật trong K hình chữ nhật bao phủ

Ví dụ:

Input	Output
20 20 3 2	10

0 0 5 2 3 1 3 6 10 0 0 4 5 3 4 1 3	Giải thích: Hình chữ nhật thứ 2 có tổng lớn nhất

Bài 4:

Cho một số nguyên dương N. Hãy tìm số cách phân tích N thành tổng của các số nguyên dương đôi một khác nhau. In ra số cách phân tích này.

Đầu vào: Một số nguyên dương N (1≤N≤20).

Đầu ra: In ra số cách phân tích N thành tổng của các số nguyên dương đôi một khác nhau.

Giải thích ví dụ:

6 có thể phân tích được thành: 6, 5+1, 4+2, 3+2+1

For example:

Input	Result
6	4