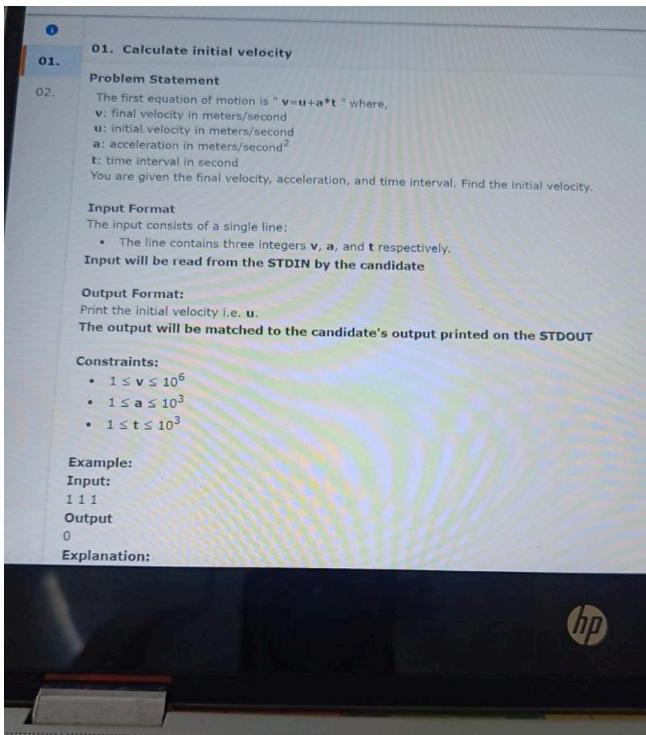


Q1.



```
#include <iostream>
using namespace std;
int main() {
    int v,a,t;
    cin>>v>>a>>t;
    int u=v-a*t;
    cout<<u;
    return 0;
}
```

Q2.

Solve to win

Your are preparing for a national-level coding competition and you want to make it to the last round. As part of your preparation , your cousin John has come up with a real time problem statement , so you could help him with the solution.

In this,

John is planning to start a business across **N** cities. As an initial investments consider all the investments are zero **{cities[N]}**. After a while John has set a goal of returning the maximum profit value from all the cities based on the number of operational goals **K** specified. Along with that ,John started partner with different vendors would also lend some investment amount for each of the operations.

You will have to help John find the maximum revenue gained across all the **N cities** based on the **K** operation(s) carried out.

Sample Input 1:

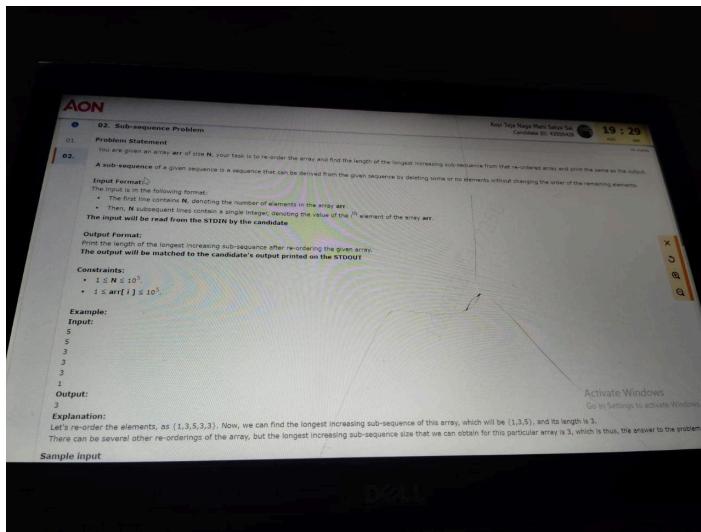
```
10 4
4 7 8
1 8 19
9 10 5
1 10 7
```

Sample Output:

```
34
```

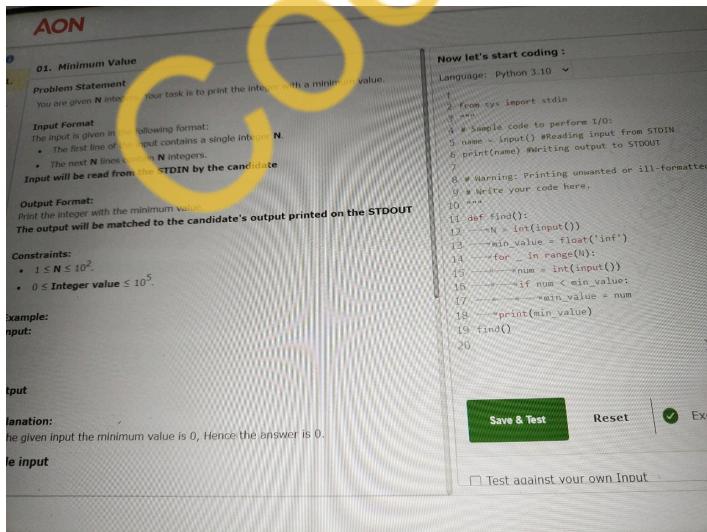
```
#include <iostream>
#include <vector>
using namespace std;
int main() {
    int N,K;
    cin>>N>>K;
    vector<long long> diff(N+2,0);
    for(int i=0;i<K;i++){
        int l,r,val;
        cin>>l>>r>>val;
        diff[l]+=val;
        diff[r+1]-=val;
    }
    long long ans=0,cur=0;
    for(int i=1;i<=N;i++){
        cur+=diff[i];
        if(cur>ans) ans=cur;
    }
    cout<<ans;
    return 0;
}
```

Q3.



```
#include <iostream>
#include <vector>
#include <set>
using namespace std;
int main() {
    int N;
    cin>>N;
    vector<int> arr(N);
    for(int i=0;i<N;i++) cin>>arr[i];
    set<int> s(arr.begin(),arr.end());
    cout<<s.size();
    return 0;
}
```

Q4.



```
#include <iostream>
using namespace std;
int main() {
    int N;
    cin>>N;
    int x,minVal=100000;
    for(int i=0;i<N;i++){
        cin>>x;
        if(x<minVal) minVal=x;
    }
    cout<<minVal;
    return 0;
}
```

Q5.

02. What's Missing

Problem Statement

You are given 3 arrays of size N , $N-1$, and $N-2$ namely A , B , and C respectively. You have to print two integers X and Y in separate lines such that

- X is an element of A but not an element of B and
- Y is an element of B but not an element of C .

Note:

- The array B contains all the elements of A except for exactly one element X .
- The array C contains all the elements of B except for exactly one element Y .

Input Format:

The input is given in the following format:

- The first line contains an integer N .
- The second line contains N space-separated integers denoting the elements of A .
- The third line contains $N-1$ space-separated integers denoting the elements of B .
- The fourth line contains $N-2$ space-separated integers denoting the elements of C .

The input will be read from the STDIN by the candidate.

Output Format:

The output is in the following format:

- Print two line-separated integers X and Y respectively.

The output will be matched to the candidate's output printed on the STDOUT.

Constraints:

$$1 \leq N \leq 500$$

Explanation:

22 is an element of array A but not an element of array B . 21 is an element of array B but not an element of array C .

Example:

Input:

```
5
2 23 12 29 21
23 12 29 2
2 12
```

Output:

```
21
2
```

Sample Input

```
3
1 2 3
1 2
```

Sample Output

```
3
2
```

Instructions:

- Program should take input from standard input and print output to standard output.
- Your code is judged by an automated system, do not write any additional welcome/greeting message.
- Your code is judged by an automated system, do not write any additional welcome/greeting message.
- "Space and Tab" key strokes for paste test cases, move mouse cursor will not be used to judge your code.
- Do not use any global variables.

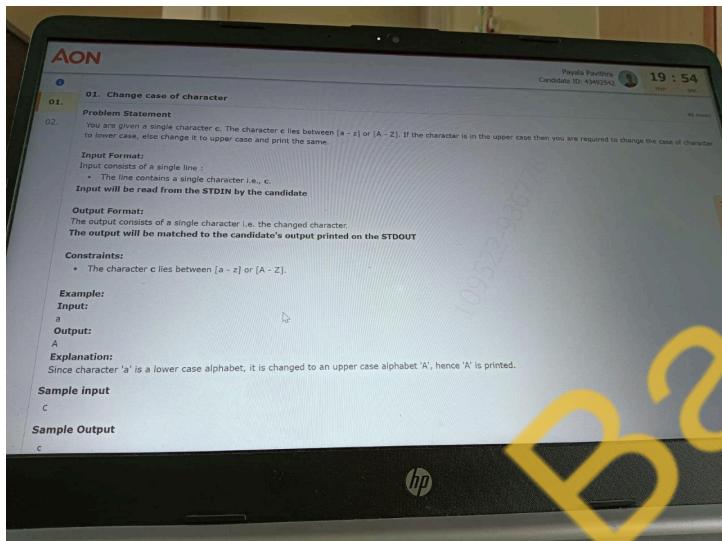
```
#include <iostream>
#include <vector>
using namespace std;
int main() {
    int N;
    cin>>N;
    vector<long long> A(N),B(N-1),C(N-2);
    for(int i=0;i<N;i++)
        cin>>A[i];
    for(int i=0;i<N-1;i++)
        cin>>B[i];
    for(int i=0;i<N-2;i++)
        cin>>C[i];
    long long X,Y;
    for(int i=0;i<N;i++)
        if(A[i] > B[0])
            X=A[i];
    for(int i=0;i<N-1;i++)
        if(B[i] > C[0])
            Y=B[i];
    cout<<X<<endl<<Y;
}
```

```

long long sumA=0,sumB=0,sumC=0;
for(int i=0;i<N;i++){cin>>A[i];sumA+=A[i];}
for(int i=0;i<N-1;i++){cin>>B[i];sumB+=B[i];}
for(int i=0;i<N-2;i++){cin>>C[i];sumC+=C[i];}
cout<<sumA-sumB<<endl;
cout<<sumB-sumC;
return 0;
}

```

Q6.

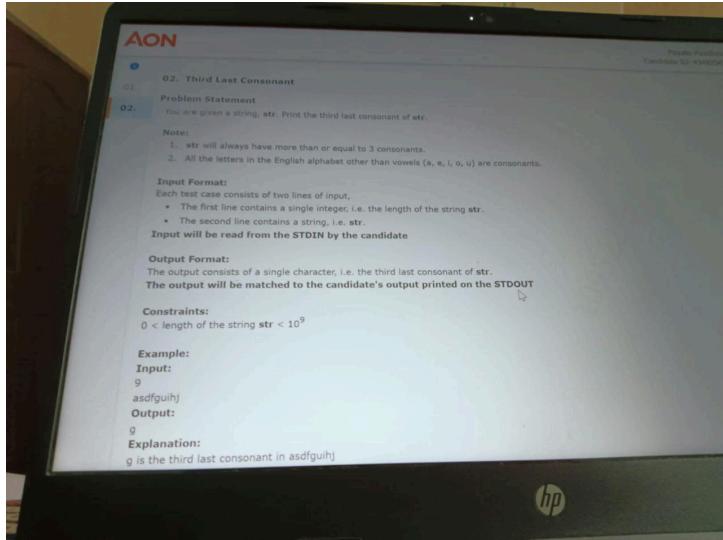


```

#include <iostream>
using namespace std;
int main() {
    char c;
    cin>>c;
    if(c>='a' && c<='z') cout<<char(c-'a'+'A');
    else cout<<char(c-'A'+'a');
    return 0;
}

```

Q7.



```
#include <iostream>
#include <string>
using namespace std;
bool isVowel(char c){
    return c=='a'||c=='e'||c=='i'||c=='o'||c=='u'||c=='A'||c=='E'||c=='I'||c=='O'||c=='U';
}
int main(){
    long long n;
    cin>>n;
    string str;
    cin>>str;
    int count=0;
    for(int i=str.size()-1;i>=0;i--){
        if(!isVowel(str[i])){
            count++;
            if(count==3){
                cout<<str[i];
                break;
            }
        }
    }
    return 0;
}
```

Q8.

02.
02.

Problem Statement
You are given an integer n find and print the sum of all of its divisors starting from 1.

Note:

- Sum lies within the integer range.

Input Format:
The input consists of a single line of input:

- The line contains a single integer, i.e. n .

Input will be read from the STDIN by the candidate

Output Format:
The output will be a single integer, i.e. the sum of all its divisors starting from 1.

The output will be matched to the candidate's output printed on the STDOUT

Constraints:
 $0 < n < 10^9$

Example:
Input:
6
Output:
12

Explanation:
Divisors of 6 are {1, 2, 3, 6}
Sum = $1 + 2 + 3 + 6 = 12$, hence the output is 12.

Sample Input
12

Sample Output
28

Instructions :
Program should take input from standard input and print output to standard output.

```
#include <iostream>
#include <cmath>
using namespace std;
int main(){
    long long n;
    cin>>n;
    long long sum=0;
    for(long long i=1;i<=sqrt(n);i++){
        if(n%i==0){
            sum+=i;
            if(i!=n/i) sum+=n/i;
        }
    }
    cout<<sum;
    return 0;
}
```

Q9.

Problem Statement
You are given two integers **A** and **B**. **A** represents a coordinate on the X-axis(0, A) and **B** represents a coordinate on the Y-axis (B,0).
There are two co-ordinate points of a right-angled triangle, the third point being the origin(0,0). You will be given **N** such triangles in the input. Find out and print the length of the hypotenuse of all the triangles.

Note:

- The formula of the length of a hypotenuse = $\sqrt{a^2 + b^2}$ where **a** and **b** represent the length of the other two sides of the triangle.
- If the length of the hypotenuse is decimal, round it to the next greater integer.

Input Format:
The input consist is given in the following format:

- The first line contains an integer **N** denoting the number of triangles.
- The next **N** lines contain two space-separated integers representing **A** and **B** respectively.

The input will be read from the STDIN by the candidate

Output Format:
The output consists of **N** lines:

- Each line representing the length of the hypotenuse of the i^{th} triangle.

The output will be matched to the candidate's output printed on the STDOUT

Constraints:

- $1 \leq A, B \leq 10^9$

Example:
Input:
2
20 21
8 15
Output:
29

```
#include <iostream>
#include <cmath>
using namespace std;
int main(){
    int N;
    cin>>N;
    while(N--){
        long long A,B;
        cin>>A>>B;
        double h=sqrt((double)A*A+(double)B*B);
        cout<<(long long)ceil(h)<<endl;
    }
    return 0;
}
```

Q10.

Problem Statement
A string is called special if it can be written as the concatenation of two strings 's1' and 's2' such that 's1' contains all characters equal to 'v', and 's2' contains all characters equal to 'w'.
You are provided with **T** test cases and for each test case, you are provided with string **s**. Your task is to find and print the number of special strings.

Note:

- Each string contains only lowercase characters.

Input Format:
The input is given in the following format:

- The first line contains **T**.
- Each of the **T** subsequent lines contains string **s**.

The input will be read from the STDIN by the candidate

Output Format:
Print the number of special strings.
The output will be matched to the candidate's output printed on the STDOUT

Constraints:

- $1 \leq T \leq 100$.
- $1 \leq \text{string length} \leq 5000$.

Example:
Input:
4
ab
xyz
aab
axb

```

4
ab
xyz
aab
axb
Output:
2
Explanation:
For the third test case, we can have s1= "aa" and s2= "b". Since the given strings are the concatenation of s1 and s2, where s1 contains all characters as 'a', and s2 contains all characters as 'b', and both s1 and s2 are non-empty, hence, this is a special string.
A similar explanation can be framed for the first test case as well, and those are the only two special strings in the given list of strings, and hence, the answer is 2.
Sample Input
2
abab
aabbb
Sample Output
1
Instructions :
• Program should take input from standard input and print output to standard output.
• Your code is judged by an automated system, do not write any additional welcome/greeting messages.
• "Save and Test" only checks for basic test cases; more rigorous cases will be used to judge your code while scoring.
• Additional score will be given for writing optimized code both in terms of memory and execution time.

Now let's start coding :
Language: C (GCC 11.3) ▾

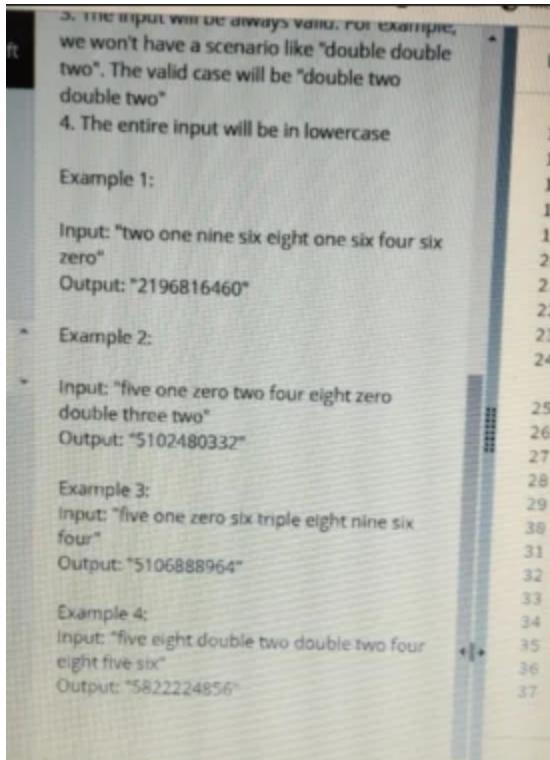
```

```

#include <iostream>
#include <string>
using namespace std;
bool isSpecial(const string &s){
    int n=s.size();
    for(int i=1;i<n;i++){
        string s1=s.substr(0,i);
        string s2=s.substr(i);
        bool ok1=true,ok2=true;
        for(char c:s1) if(c!='a') {ok1=false;break;}
        for(char c:s2) if(c!='b') {ok2=false;break;}
        if(ok1 && ok2) return true;
    }
    return false;
}
int main(){
    int T;
    cin>>T;
    int count=0;
    while(T--){
        string s;
        cin>>s;
        if(isSpecial(s)) count++;
    }
    cout<<count;
    return 0;
}

```

Q11.



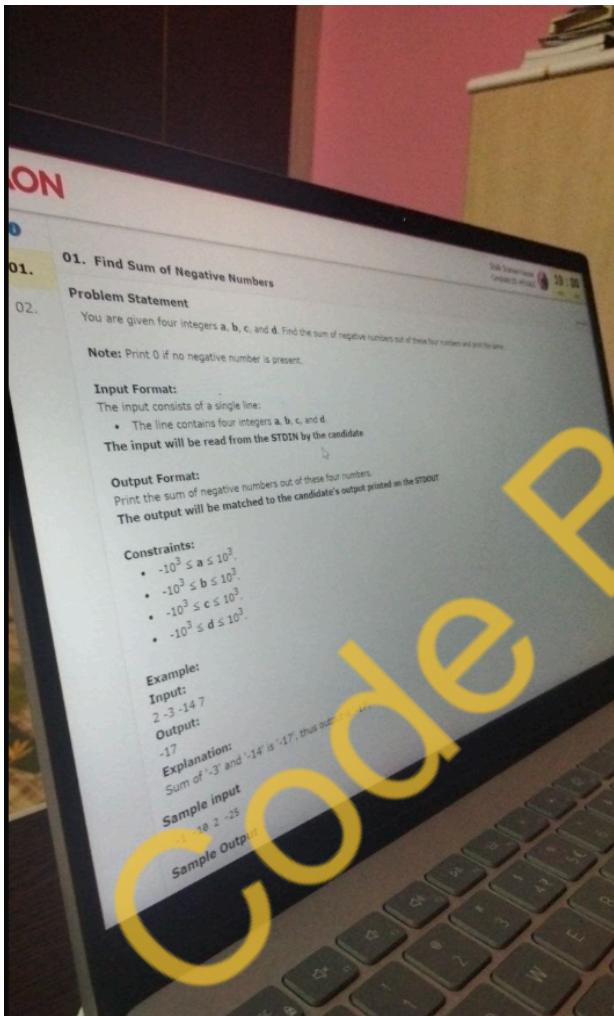
```
#include <iostream>
#include <sstream>
#include <string>
#include <unordered_map>
using namespace std;
int main(){
    string line;
    getline(cin,line);
    unordered_map<string,char> mp={
        {"zero",'0'}, {"one",'1'}, {"two",'2'}, {"three",'3'}, {"four",'4'},
        {"five",'5'}, {"six",'6'}, {"seven",'7'}, {"eight",'8'}, {"nine",'9'}
    };
    stringstream ss(line);
    string word,res;
    while(ss>>word){
        if(word=="double"){
            string nxt; ss>>nxt;
            res.push_back(mp[nxt]);
            res.push_back(mp[nxt]);
        }else if(word=="triple"){
            string nxt; ss>>nxt;
            res.push_back(mp[nxt]);
            res.push_back(mp[nxt]);
            res.push_back(mp[nxt]);
        }
    }
}
```

```

    }else{
        res.push_back(mp[word]);
    }
}
cout<<res;
return 0;
}

```

Q12.



```

#include <iostream>
using namespace std;
int main(){
    int a,b,c,d;
    cin>>a>>b>>c>>d;
    int sum=0;
    if(a<0) sum+=a;
    if(b<0) sum+=b;
}

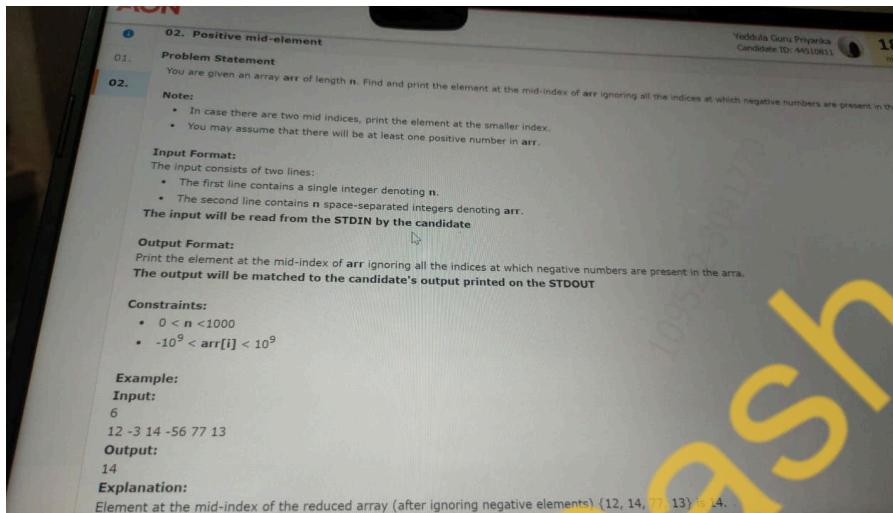
```

```

if(c<0) sum+=c;
if(d<0) sum+=d;
cout<<sum;
return 0;
}

```

Q13.

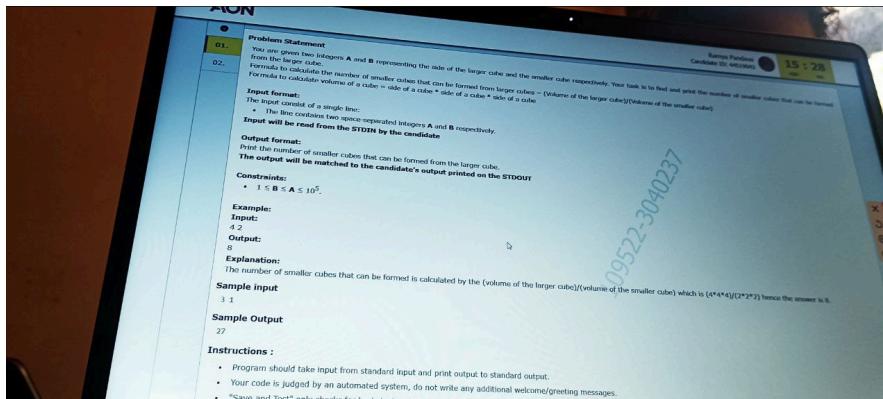


```

#include <iostream>
#include <vector>
using namespace std;
int main(){
    int n;
    cin>>n;
    vector<int> arr(n),pos;
    for(int i=0;i<n;i++){
        cin>>arr[i];
        if(arr[i]>=0) pos.push_back(arr[i]);
    }
    int mid=pos.size()/2;
    if(pos.size()%2==0) cout<<pos[mid-1];
    else cout<<pos[mid];
    return 0;
}

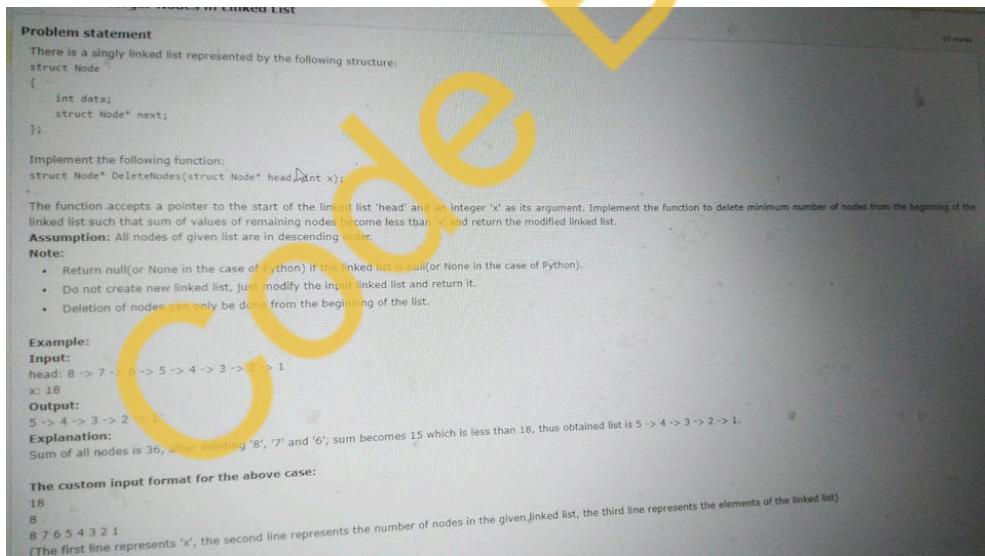
```

Q14.



```
#include <iostream>
using namespace std;
int main(){
    long long A,B;
    cin>>A>>B;
    long long big=A*A*A;
    long long small=B*B*B;
    cout<<big/small;
    return 0;
}
```

Q15.



```
#include <iostream>
using namespace std;
```

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

```
};

Node* createList(int n) {
    Node* head = nullptr;
    Node* tail = nullptr;
    for(int i=0;i<n;i++){
        int val; cin>>val;
        Node* node = new Node();
        node->data = val;
        node->next = nullptr;
        if(!head) head = tail = node;
        else { tail->next = node; tail = node; }
    }
    return head;
}

Node* DeleteNodes(Node* head, int x) {
    if(!head) return nullptr;
    long long sum = 0;
    Node* temp = head;
    while(temp){ sum += temp->data; temp = temp->next; }
    while(head && sum >= x){
        sum -= head->data;
        head = head->next;
    }
    return head;
}

int main(){
    int x,n;
    cin>>x>>n;
    Node* head = createList(n);
    head = DeleteNodes(head,x);
    if(!head) return 0;
    while(head){
        cout<<head->data;
        if(head->next) cout<<"->";
        head = head->next;
    }
    return 0;
}
```