Top 25 Hackerrank Coding Questions and Answers

June 10, 2021

Hackerrank Coding Questions for Practice

Below you can find the Top 25 Hackerrank based coding questions with solutions for the Hackerrank Coding test. in this article we have collected the most asked and most important Hackerrank coding questions that you need to prepare to successfully crack Hackerrank coding round for companies like IBM, Goldman Sachs, Cisco, Mountblu, Cognizant, etc.

Here you can practice all the Top 25 free !!! coding questions that were asked in the latest placement drives held by Hackerrank
Hackerrank Coding questions are bit difficulty then the usual coding questions, as most of the product based companies hire through this platform.

Coding Questions

Sample Hackerrank Coding Questions

Details about Hackerrank as a Hiring platform

Topics	Details
Number of Questions asked by companies	2 – 5
Time Limit	2 – 4.5 hrs approx
Difficulty level	High
Package Offered	6 LPA – 12 LPA

Hackerrank Coding Questions are used by multiple organizations and MNC(s) for hiring coding proficient students, using

Hackerrank Platform. For instance Hackerrank regularly hold coding competitions sponsored by specific companies as a result to hire engineers. However these contests vary in duration, rules, or challenge type/topic, depending on what the sponsor is looking to test for. After the contest, the sponsoring companies contact top performers on the leader-board about job opportunities.

Shortcut keys (hotkeys) allowed are :

- **alt/option + R :** Run code
- **alt/option + Enter :** Submit code
- **alt/option + F :** Enable full screen
- **Esc :** Restore full screen

List of Hackerrank Practice Coding Questions

Hackerrank Coding Questions with Solutions

Question 1 – Maximum Passengers

Problem Statement -: A taxi can take multiple passengers to the railway station at the same time.On the way back to the starting point,the taxi driver may pick up additional passengers for his next trip to the airport.A map of passenger location has been created,represented as a square matrix.

The Matrix is filled with cells,and each cell will have an initial value as follows:

- A value greater than or equal to zero represents a path.
- A value equal to 1 represents a passenger.
- A value equal to -1 represents an obstruction.

The rules of motion of taxi are as follows:

- The Taxi driver starts at (0,0) and the railway station is at (n-1,n-1).Movement towards the railway station is right or down,through valid path cells.
- After reaching (n-1,n-1) the taxi driver travels back to (0,0) by travelling left or up through valid path cells.
- When passing through a path cell containing a passenger,the passenger is picked up.once the rider is picked up the cell becomes an empty path cell.
- If there is no valid path between (0,0) and (n-1,n-1),then no passenger can be picked.
- The goal is to collect as many passengers as possible so that the driver can maximize his earnings.

For example consider the following grid,

```
0  1
```

-1 0

Start at top left corner.Move right one collecting a passenger. Move down one to the destination.Cell (1,0) is blocked,So the return path is the reverse of the path to the airport.All Paths have been explored and one passenger is collected.

Returns:

Int : maximum number of passengers that can be collected.

Sample Input 0

4 -> size n = 4

4 -> size m = 4

0 0 0 1 -> mat

1 0 0 0

0 0 0 0

0 0 0 0

Output 0

2

Explanation 0

The driver can contain a maximum of 2 passengers by taking the following path (0,0) → (0,1) → (0,2) → (0,3) → (1,3) → (2,3) → (3,3) → (3,2) → (3,1) → (3,0) → (2,0) → (1,0) → (0,0)

Sample Input 1

STD IN Function

———— ————-

3 → size n=3

3 → size m=3

0 1 -1 → mat

1 0 -1

1 1 1

Sample Output 1

5

Explanation 1

The driver can contain a maximum of 5 passengers by taking the following path (0,0) → (0,1) → (1,1) → (2,1) → (2,2) → (2,1) → (2,0) → (1,0) → (0,0)

C++ Java

```
#include <bits/stdc++.h>
using namespace std;
int n, m;
int mat[105][105];
map<pair<int, pair<int, int>>, int> dp;
bool isValid(int i, int j)
{
    if (mat[i][j] == -1)
        return false;
    if (i < 0 || i >= n)
        return false;
    if (j < 0 || j >= m)
        return false;
    return true;
}
int solve(int i, int j, int x, int y)
{
    if (!isValid(i, j))
    {
        return INT_MIN;
    }
    if (!isValid(x, y))
    {
        return INT_MIN;
    }
    if (i == n - 1 && x == n - 1 && j == m - 1 && y == m - 1)
    {
        if (mat[i][j] == 1)
```

```
        {
            return 1;
        }
        else
        {
            return 0;
        }
    }
    if (dp.find({i, {j, x}}) != dp.end())
        return dp[{i, {j, x}}];

    int cur = 0;
    if (i == x && j == y)
    {
        if (mat[i][j] == 1)
            cur = 1;
    }
    else
    {
        if (mat[i][j] == 1)
            cur++;
        if (mat[x][y] == 1)
            cur++;
    }

    int op1 = solve(i + 1, j, x + 1, y);
    int op2 = solve(i, j + 1, x, y + 1);
    int op3 = solve(i + 1, j, x, y + 1);
    int op4 = solve(i, j + 1, x + 1, y);

    int ans = cur + max(op1, max(op2, max(op3, op4)));

    return dp[{i, {j, x}}] = ans;
}
int main()
{
    cin >> n >> m;

    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            cin >> mat[i][j];
    int ans = solve(0, 0, 0, 0);
    if (ans >= 0)
        cout << solve(0, 0, 0, 0) << endl;
    else
        cout << -1 << endl;

    return 0;
}
```

Question 2 – Minimum streets lights

Problem Statement -: Street Lights are installed at every position along a 1-D road of length n. Locations[] (an array) represents the coverage limit of these lights. The ith light has a coverage limit of locations[i] that can range from the position max((i – locations[i]), 1) to min((i + locations[i]), n) (Closed intervals). Initially all the lights are switched off. Find the minimum number of fountains that must be switched on to cover the road.

Example

n = 3

locations[] = {0, 2, 13}then

For position 1: locations[1] = 0, max((1 – 0),

1) to mini (1+0), 3) gives range = 1 to 1

For position 2: locations[2] = 2, max((2-2)

For position 2: locations[2] = 2, max((2-2),

1) to min((2+2), 3) gives range = 1 to 3

For position 3: locations[3] = 1, max((3-1),

1) to min((3+1), 3) gives range = 2 to 3

For the entire length of this road to be covered, only the light at position 2 needs to be activated.

Returns:

int : the minimum number of street lights that must be activated

Constraints :

- 1<_n<_ 10^5
- O<_locations[i] <_ mini (n,100) (where 1 <_1<_10^5)

Sample Input For Custom Testing :

3 ->locations[] size n = 3

1 ->locations[] [1, 1, 1]

1 ->Sample Output

Sample Output :

1

C++ Python Java

```
#include <bits/stdc++.h>
#define ll long long
using namespace std;
bool compare(pair<int, int> A, pair<int, int> B)
{
    if (A.first = B.first)
        return A.second < B.second;
    return A.first < B.first;
}
int solve(int location[], int n)
{

    pair<int, int> range[n];
    for (int i = 0; i < n; i++)
    {
        int id = i + 1;
        range[i].first = max(1, id - location[i]);
        range[i].second = min(n, id + location[i]);
    }

    sort(range, range + n, compare);

    int i = 0;
    int ans = 0;
    while (i < n)
    {
        pair<int, int> p = range[i];
        ans++;
        while (i + 1 < n && range[i].first == range[i + 1].first)
        {
            p.second = max(p.second, range[i + 1].second);
            i++;
        }
        //cout<<p.second<<" "<<i<<endl;
        while (i < n && range[i].second <= p.second)
            i++;
        //cout<<p.second<<" "<<i<<endl;
    }
    return ans;
}
int main()
{
    int n;
    cin >> n;
    int location[n];
    for (int i = 0; i < n; i++)
        cin >> location[i];

    cout << solve(location, n) << endl;
    return 0;
}
```

Question 3 – Maximize Earnings

Problem Statement -: A company has a list of jobs to perform. Each job has a start time, end time and profit value. The manager has asked his employee Anirudh to pick jobs of his choice. Anirudh being greedy wants to select jobs for him in such a way that would maximize his earnings.

Given a list of jobs how many jobs and total earning are left for other employees once Anirudh

Picks jobs of his choice.

Note: Anirudh can perform only one job at a time.

Input format:

Input format:

Each Job has 3 pieces of info – Start Time,End Time and Profit

The first line contains the number of Jobs for the day. Say 'n'. So there will be '3n lines following as each job has 3 lines.

Each of the next '3n' lines contains jobs in the following format:

- start_time
- end-time
- Profit

start-time and end-time are in HHMM 24HRS format i.e. 9am is 0900 and 9PM is 2100

Constraints

- The number of jobs in the day is less than 10000 i.e. 0<_n<_10000
- Start-time is always less than end time.

Output format :-

Program should return an array of 2 integers where 1st one is number of jobs left and earnings of other employees.

Sample Input 1 :

3

0900

1030

100

1000

1200

500

53

1100

1200

300

Sample Output 1:

2

400

Sample Explanation 1

Anirudh chooses 1000-1200 jobs. His earnings is 500. The 1st and 3rd jobs i.e. 0900-1030 and 1100-1200 respectively overlap with the 2nd jobs. But profit earned from them will be 400 only. Hence Anirudh chooses 2nd one. Remaining 2 Jobs & 400 cash for other employees.

Sample Input 2:

5

0805

0830

100

0835

0900

100

0905

0930

100

0935

1000

100

1005

1030

100

Sample output 2:

0

0

Sample Explanation 2:

Anirudh can work on all appointments as there are none overlapping. Hence 0 appointments and 0 earnings for other employees.

C++ Java

```
#include <bits/stdc++.h>
using namespace std;
class job
{
public:
    int st, ed, cost;
};
int getTime(string s)
{
    int hr = (s[0] - '0') * 10 + (s[1] - '0');
    int min = (s[2] - '0') * 10 + (s[3] - '0');

    return hr * 60 + min;
}
bool compare(job A, job B)
{
    return A.ed < B.ed;
}
int searchJob(job arr[], int st, int ed, int key)
{
    int ans = -1;
    while (st <= ed)
    {
        int mid = (st + ed) / 2;
        if (arr[mid].ed <= key)
        {
            ans = mid;
            st = mid + 1;
        }
        else
        {
            ed = mid - 1;
        }
    }
    return ans;
}
pair<int, int> solve(job arr[], int n)
{
    int dp[n] = {0};
    int numOfJobs[n] = {0};

    dp[0] = arr[0].cost;
    numOfJobs[0] = 1;

    for (int i = 1; i < n; i++)
    {
        int cur = arr[i].cost;
        int num = 1;
        int idx = searchJob(arr, 0, i - 1, arr[i].st);

        if (idx != cur)
        {
            cur += dp[idx];
            num += numOfJobs[idx];
        }
        if (cur > dp[i - 1])
```

```
        if (cur > dp[i - 1])
        {
            dp[i] = cur;
            numOfJobs[i] = num;
        }
        else
        {
            dp[i] = dp[i - 1];
            numOfJobs[i] = numOfJobs[i - 1];
        }
    }
    return {numOfJobs[n - 1], dp[n - 1]};
}
int main()
{
    int n;
    cin >> n;

    job arr[n];
    int cost;
    string st, ed;
    int total = 0;

    for (int i = 0; i < n; i++)
    {
        cin >> st >> ed >> cost;
        arr[i].st = getTime(st);
        arr[i].ed = getTime(ed);
        arr[i].cost = cost;
        total += cost;
    }
    sort(arr, arr + n, compare);

    pair<int, int> res = solve(arr, n);

    cout << n - res.first << endl;
    cout << total - res.second << endl;

    return 0;
}
```

Question 4 : Network Stream

Problem Statement – A stream of n data packets arrives at a server. This server can only process packets that are exactly 2^n units long for some non-negative integer value of n (0<=n).

All packets are repackaged in order to the 1 largest possible value of 2^n units. The remaining portion of the packet is added to the next arriving packet before it is repackaged. Find the size of the largest repackaged packet in the given stream.

Example :

- arriving Packets = [12, 25, 10, 7, 8]
- The first packet has 12 units. The maximum value of 2^n that can be made has 2^n = 2^3 = 8 units because the next size up is 2^n = 2^4 = 16 (16 is greater than 12).
- 12 – 8 = 4 units are added to the next packet. There are 4 + 25 = 29 units to repackage, 2^n = 2^4 = 16 is the new size leaving 9 units (29-16 = 9)
  Next packet is 9 + 10 = 29 unists & the maximum units(in 2^n) is 16 leaving 3 units.
- 3 + 7 = 10 , the max units is 8 Leaving 2 units, and so on.
- The maximum repackaged size is 16 units.

Returns:

- Long : the size of the largest packet that is streamed

Constraints :

- 1<=n<=10^5
- 1<=arriving Packets[i] size<=10^9

Sample case :

Sample input :

5 → number of packets=5

12 → size of packets=[13,25,12,2,8]

25

10

2

8

Sample output :

16

Python | Java

```
def largeRepackagedPacket(arr):
    twoP=[int(2**i) for i in range(31)]
    x=0
    ans=0
    for i in arr:
        i=i+x
        for j in range(31):
            if i<twoP[j]:
                break
        x=i-twoP[j-1]
        if ans<=twoP[j-1]:
            ans=twoP[j-1]
    return ans


Packets=[]
for i in range(int(input())):
    Packets.append(int(input()))
print(largeRepackagedPacket(Packets))
```

Question 5 – Astronomy Lecture

Problem Statement -: Anirudh is attending an astronomy lecture. His professor who is very strict asks students to write a program to print the trapezium pattern using stars and dots as shown below . Since Anirudh is not good in astronomy can you help him?

Sample Input:

N = 3

Output:

.

..

.....

..

.

C++ Python Java

```
#include <stdio.h>
int main()
{
    int i,j,n;
    scanf(“%d”,&n);
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            if(j<n-i-1)
                printf(“*”);
            else
               printf(“.”);
        }
        for(j=0;j<n-1;j++)
        {
            if(j<i)
                printf(“.”);
            else
                printf(“*”);
        }
        printf(“\n“);
    }

 for(i=2;i<=n;i++)
    {
     for(j=0;j<n;j++)

        {
            if(j<i-1)
```

```
            printf("*");
        else
            printf(".");

      }
    for(j=0;j<n-1;j++)
        {
            if(j<n-i)
               printf(".");
            else
                printf("*");
        }
        printf("\n");
    }
    return 0;
}
```

Question 6 – Disk Space Analysis

Problem Statement -: You are given an array, You have to choose a contiguous subarray of length 'k', and find the minimum of that segment, return the maximum of those minimums.

Sample input 0

1 → Length of segment x =1

5 → size of space n = 5

1 → space = [1,2,3,1,2]

2

3

1

2

Sample output

3

Explanation

The subarrays of size x = 1 are [1],[2],[3],[1], and [2],Because each subarray only contains 1 element, each value is minimal with respect to the subarray it is in. The maximum of these values is 3. Therefore, the answer is 3

C++ Python Java

```
#include <bits/stdc++.h>
using namespace std;
vector<int> arr;
int prevmin=-1;
int flag=0;
int x,n,q;

int sorting(int start,int end)
{
    if(start+1==n) {start=0;end=end-n;}
    if(start==end) return arr[start];
    return min(arr[start],sorting(start+1,end));
}
```

```
}

int func(int start,int end)
{
    if(flag==0) {flag++;return prevmin=sorting(start,end);}
    if(arr[start-1]==prevmin) return prevmin;
    return prevmin=(arr[end]<=prevmin)?prevmin:sorting(start,end);

}

int main()
{
    cin>>x>>n;
    int ans=0;
    for(int i=0;i<n;i++)
    {cin>>q;arr.push_back(q);}
    for(int i=0;i<n;i++)
    {
       ans=max(ans,func(i,i+x-1));
    }
    cout<<ans;
}
```

Question 7 : Guess the word

Problem Statement – Kochouseph Chittilappilly went to Dhruv Zplanet , a gaming space, with his friends and played a game called "Guess the Word".

Rules of games are –

- Computer displays some strings on the screen and the player should pick one string / word if this word matches with the random word that the computer picks then the player is declared as Winner.
- Kochouseph Chittilappilly's friends played the game and no one won the game. This is Kochouseph Chittilappilly's turn to play and he decided to must win the game.
- What he observed from his friend's game is that the computer is picking up the string whose length is odd and also that should be maximum. Due to system failure computers sometimes cannot generate odd length words. In such cases you will lose the game anyways and it displays "better luck next time". He needs your help. Check below cases for better understand

Sample input :

5 → number of strings

Hello Good morning Welcome you

Sample output :

morning

Explanation:

- Hello → 5
- Good → 4
- Morning → 7
- Welcome → 7
- You → 3

First word that is picked by computer is morning

Sample input 2 :

3

Go to hell

Sample output 2:

Better luck next time

Explanation:

Here no word with odd length so computer confuses and gives better luck next time

Java | Python

```
import java.util.*;
public class OddLength
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        String arr[]=new String[n];

        for(int i=0;i<n;i++)
        arr[i]=sc.next();

        int len=0;
        ArrayList<String>  oddLength=new ArrayList<String>();
        for(int i=0;i<n;i++)
        {
            len=arr[i].length();
            if(len%2==1)
            oddLength.add(arr[i]);
        }
        if(oddLength.size()==0)
             System.out.println("Better luck next time");
        else
        {
            Iterator itr=oddLength.iterator();
            int max=-1;
            String res="";
            while(itr.hasNext())
            {
                String temp=(String)itr.next();
                if(temp.length()>max)
                {
```

```
                    res=temp;
                    max=temp.length();
                }
            }
            System.out.println(res);
        }
    }
}
```

Question 8 – Minimum Start value

Problem Statement -: Raman was playing a game, in starting he has x coins at some point of the game he has to pay some coins to get into the next level of the game, during each game he can collect some coins. If at anypoint of the game numbers of coins of Raman is less than one he will lose the game. Find the minimum value of x such that Raman wins.

C++ Python

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

int main() {
    int n; cin>>n;
    int a[n];
    for(int i=0;i<n;i++) cin>>a[i];
    int sum=0,ans=0;
    for(int i=0;i<n;i++)
    {
        sum+=a[i];
        if(sum<1)
        {
            sum=-sum;
            ans+=sum+1;
            sum=1;
        }
    }
    cout<<ans<<endl;
}
```

Question 9 : Complex Math

Problem Statement – The math assignment says you will be given numbers, mostly with imaginary additions, that means complex numbers, and you need to add them and tell the answer in your answer script. You told your friend John that you don't know the addition of complex numbers, so John will write a program, which you can write in order to get the results of addition.

John knows Object oriented programming enough to complete the task.

Input Format:

Three integers a b and c

Output format:

First print the complex number a+bi

Next line print a + bi + c as i2.

Next line i2+a+bi

Sample Input:

4 5 2

Sample Output:

4 + 5i

6 + 5i

10 + 10i

C++ Java

```
#include <bits/stdc++.h>
using namespace std;

class Complex //Writing the variables and the functions together, but cant be changed globally -> Encapsulation
{
    private:
    int r,i; //Data Abstraction
    public:
    Complex(){r=i=0;}
    Complex(int f,int k)
    {
        r=f;i=k;
    }
    void show()
    {
        cout<<r<<" + "<<i<<"i"<<endl;
    }
    Complex operator+(Complex obj)
    {
        Complex temp;
        temp.r=r+obj.r;
        temp.i=i+obj.i;
        return temp;
    }
    Complex operator+(int a)
```

```
    {
        Complex temp;
        temp.r=r+a;
        return temp;
    }
};

int main()
{
    int a,b,c;
    cin>>a>>b>>c;
    Complex i1(a,b);
    i1.show();
    Complex i2;
    i2=i1+c; //Operator Overloading -> Polymorphism
    i2.show();
    (i1+i2).show();

}
```

Question 10 : Minimum Occurrence

Problem Statement – Given a sting , return the character that appears the minimum number of times in the string. The string will contain only ascii characters, from the ranges (“a”-”z”,”A”-”Z”,0-9), and case matters . If there is a tie in the minimum number of times a character appears in the string return the character that appears first in the string.

Input Format:

Single line with no space denoting the input string.

OutputFormat:

Single character denoting the least frequent character.

Constraints:

Length of string <=10^6

Sample Input:

cdadcda

Sample Output:

c

Explanation:

C and A both are with minimum frequency. So c is the answer because it comes first with less index.

C++ Python Java

```
include <bits/stdc++.h>
using namespace std;

map<char,int> ma;

int main()
{
    string s;int m=INT_MAX;
    cin>>s;
    for(int i=s.length()-1;i>=0;i--)
    {
        ma[s[i]]++;
        m=min(m,ma[s[i]]);
    }
    for(int i=0;i<s.length();i++)
```

```
    if(ma[s[i]]==m) {cout<<s[i];break;}
}
```

Question 11 : Devil Groups

Problem Statement –

There are some groups of devils and they splitted into people to kill them. Devils make People to them left as their group and at last the group with maximum length will be killed. Two types of devils are there namely "@" and "$"
People is represented as a string "P"

Input Format:

First line with the string for input

Output Format:

Number of groups that can be formed.

Constraints:

2<=Length of string<=10^9

Input string

PPPPPP@PPP@PP$PP

Output

7

Explanation

4 groups can be formed

- PPPPPP@
- PPP@
- PP$
- PP

Most people in the group lie in group 1 with 7 members.

C++ Python Java

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    string s;
    cin>>s;
    int a=0,mx=0;
    for(int i=0;i<s.length();i++)
    {
        a++;
        if(s[i]=='@'||s[i]=='$')
            {mx=max(a,mx);a=0;}
    }
    cout<<mx;
}
```

Question 12 : Vampire Battle

Problem Statement – Stephan is a vampire. And he is fighting with his brother Damon. Vampires get energy from human bloods, so they need to feed on human blood, killing the human beings. Stephan is also less inhuman, so he will like to take less life in his hand. Now all the people's blood has some power, which increases the powers of the Vampire. Stephan just needs to be more powerful than Damon, killing the least human possible. Tell the total power Steohan will have after drinking the bloods before the battle.

- Note that: Damon is a beast, so no human being will be left after Damon drinks everyone's blood. But Stephan always comes early in the town.

Input Format:

First line with the number of people in the town, n.

Second line with a string with n characters, denoting the one digit power in every blood.

Output Format:

Total minimum power Stephan will gather before the battle.

Constraints:

- n<=10^4

Sample input :

6

093212

Sample output :

9

Explanation:

Stephan riches the town, drinks the blood with power 9. Now Damon cannot reach 9 by drinking all the other bloods.

C++ Python

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int n,sum=0,sum1=0; cin>>n;
    string s;
    cin>>s;
    sort(s.begin(),s.end(),greater<char>());
    for(auto i:s) sum+=(i-'0');
    for(auto i:s)
    {if(sum1>sum) break; sum1+=(i-'0');sum-=i-'0';}
    cout<<sum1;
}
```

Question 13 : Copycat in exam

Problem Statement – Rahul copies in the exam from his adjacent students. But he doesn't want to be caught, so he changes words keeping the letter constant. That means he interchanges the positions of letters in words. You are the examiner and you have to find if he has copied a certain word from the one adjacent student who is giving the same exam, and give Rahul the

markings he deserves.

Note that: Uppercase and lowercase are the same.

Input Format:

- First line with the adjacent student's word
- Second line with Rahul's word

Output Format:

- 0 if not copied
- 1 if copied

Constraints:

- 1<=Length of string<=10^6

Sample Input:

CAR

Acr

Sample Output:

1

C++ Python Java

```
#include<bits/stdc++.h>
using namespace std;

int main()
{
    string s2,s1;
    cin>>s1>>s2;
    transform(s1.begin(),s1.end(),s1.begin(),::tolower);
    transform(s2.begin(),s2.end(),s2.begin(),::tolower);
    sort(s1.begin(),s1.end());
    sort(s2.begin(),s2.end());
    cout<< (s2==s1);

}
```

Question 14 : Mr. Robot's Password

Problem Statement – Mr. Robot is making a website, in which there is a tab to create a password. As other websites, there are rules so that the password gets complex and none can predict the password for another. So he gave some rules like:

- At least one numeric digit
- At Least one Small/Lowercase Letter
- At Least one Capital/Uppercase Letter
- Must not have space
- Must not have slash (/)
- At least 6 characters

If someone inputs an invalid password, the code prints: "Invalid password, try again".

Otherwise, it prints: "password valid".

Input Format:

Input Format:

A line with a given string as a password

Output Format:

- If someone inputs an invalid password, the code prints: “Invalid password, try again”.
- Otherwise, it prints: “password valid”, without the quotation marks.

Constraints:

- Number of character in the given string <=10^9

Sample input 1:

abjnlL09

Sample output 1:

password valid

Sample input 2:

jjnaskpk

Sample output 2:

Invalid password, try again

C++ Python

```
#include<bits/stdc++.h>
using namespace std;

int CheckPassword(char str[],int n)
{
   if(n<4) return 0;
   int a=0,cap=0,nu=0,low=0;
   while(a<n)
   {
       if(str[a]==' ' || str[a]=='/') return 0;
       if(str[a]>=65&&str[a]<=90) {cap++;}
       if(str[a]-32>=65&&str[a]-32<=90) {low++;}

       else if(str[a]-'0'>=0 && str[a]-'0'<=9) nu++;
       a++;
   }
   return cap>0 && nu>0 && low>0 ;
}

int main()
{
    string s;
    getline(cin,s);
    int len=s.size();
    char *c=&s[0];
    if(CheckPassword(c,len))
    cout<<"password valid";
    else cout<<"Invalid password, try again";
}
```

Question 15 : Weird Terminal

Problem Statement – Here is a weird problem in Susan's terminal. He can not write more than two words each line, if she writes more than two, it takes only 2 words and the rest are not taken. So she needs to use enter and put the rest in a new line. For a given paragraph, how many lines are needed to be written in Susan's terminal?

Input Format:

- A string as the text to input in the terminal

Output Format:

- Number of lines written.

Constraints:

- Number of words <=10^7

Sample Input:

How long do you have to sit dear ?

Sample Output:

4

Explanation:

The writing will be:

- How long
- Do you
- Have to
- Sit dear ?

C++ Python

```
#include<bits/stdc++.h>
using namespace std;

int main()
{
    string s;
    getline(cin,s);
    int ans=0;
    istringstream ss(s);
    while(ss)
    {
        string word;
        int flag=0;
        ss>>word;
        if(word=="") break;
        for(int i=0;i<word.length();i++)
        if(!isalpha(word[i]))
        {
            if(i!=0){
            if(i==word.length()-1){
            if(word[i]==','||word[i]=='.'||word[i]==';'||word[i]==':'||word[i]=='?'||word[i]=='!') continue;}
            else if(word[i]=='.'&&word[i+1]!='.') {flag++;break;}
            else if(word[i]=='-'&&isalpha(word[i+1])) continue;
            else {flag++;break;}
            }
            else
            {
                flag++;break;
            }

        }

        if(flag==0) {ans++;
        //cout<<word<<" "<<word.length()<<endl;
        }
    }
    cout<<floor((ans-1)/2) +1;
}
```

Question 16 : Set Bit calculator

Problem Statement – Angela plays with the different bits. She was given a bunch of numbers and she needs to find how many set bits are there in total. Help Angela to impress her, write a code to do so.

Input Format:

- First line with n, an integer
- Next n lines denoting n integers angela is given.

Output Format:

- Total number of set bits

Constraints:

- Number of elements or number <=10^7
- numbers<=10000

Sample Input:

4

1

3

2

1

Sample Output:

5

C++

```
#include<bits/stdc++.h>
using namespace std;

int main()
{
    int n,m,s=0;cin>>n;
    while(n--)
    {
        cin>>m;
        s+=__builtin_popcount(m);
    }
    cout<<s;
}
```

Question 17 : Duplicates

Problem Statement – The principal has a problem with repetitions. Everytime someone sends the same email twice he becomes angry and starts yelling. His personal assistant filters the mails so that all the unique mails are sent only once, and if there is someone sending the same mail again and again, he deletes them. Write a program which will see the list of roll numbers of the student and find how many emails are to be deleted.

Sample Input:

6

1

3

3

4

3

3

Sample Output:

3

C++ Python Java

```
#include<bits/stdc++.h>
using namespace std;
map<int,int> m;
int main()
{
    int n,a,ans=0;
    cin>>n;
    while(n--)
    {
        cin>>a;
        if(m[a]==0)
        {
            m[a]++;
```

```
            ans++;
        }
    }
    cout<<ans;

}
```

Question 18 : Device Name System

Problem Statement – Rocky is a software engineer and he is creating his own operating system called “myFirst os”. myFirst os is a GUI (Graphical user interface) based operating system where everything is stored in files and folders. He is facing issues on creating unique folder names for the operating system . Help rocky to create the unique folder name for it’s os.If folder name already exists in the system and integer number is added at the name to make it unique. The integer added starts with 1 and is incremented by 1 for each new request of an existing folder name. Given a list of folder names , process all requests and return an array of corresponding folder names.

Example :

- n=5
- foldername= [‘home’ , ‘myfirst’ ,’downloads’, ‘myfirst’, ‘myfirst’]
- foldername[0]= ‘home’ is unique.
- foldername[1]= ‘myfirst’ is unique.
- foldername [2]=’downloads’ is unique.
- foldername[3]=’myfirst’ already exists in our system. So Add1 at the end of the folder name i.e foldername[3]=”myfirst1"
- foldername[4]=’myfirst’ also already exists in our system.So add 2 at the end of the folder name i.e. foldername[4]=”myfirst2".

folderNameSystem function has the following parameters

- string foldername[n]: an array of folder name string in the order requested

Returns:

- String[n]: an array of strings usernames in the order assigned

Constraints :

- 1<=n<=10^4
- 1<=length of foldername[i]<20
- foldername[i] contains only lowercase english letter in the range ascii[a-z]

Input Format:

- The first line contains an integer n , denoting the size of the array usernames
- Each line i of the n subsequent lines (where i<=0<=n) contains a string usernames[i] representing a username request in the order received.

Sample case :

4

home

download

first

first

Sample Output :

home

download

first

first1

Explanation :

- foldername[0] = 'home' is unique
- foldername[1]='download' is unique
- foldername[2]= 'first' is unique
- foldername[3]='first' is already existing . so add 1 to it and it become first1

C++ Python Java

```
#include<bits/stdc++.h>
using namespace std;

int main()
{
    int n;cin>>n;vector<string> v(n);
    for(int i=0;i<n;i++)
    cin>>v[i];
    map<string,int> m;
    for(auto i:v)
    {
        if(m[i]) cout<<i<<m[i]<<endl;
        else cout<<i<<endl;
        m[i]++;
    }
}
```

Question 19 : Formatting large Products

Problem Statement – Rohan is weak in mathematics.He is giving mathematics Olympiad , but he got stuck in one of the question. Help rohan to solve the question.In Question there are two positive integer A and B. You have to find the product of all integer between A and B which is represented in the form C=D*10^E , where C is the product of numbers , D and E are non-negative integers and the last digit of D is non-zero.

Function Description

Complete the function formatProducts in the editor below, formatProduct must return a string that represents C in the above described form.

Function has the following parameters

- A: an integer
- B: an integer

Constraints :

- A will between 1 and 1,000,000 . Inclusive.
- B will be between A and 1,000,000. Inclusive.

Sample Input :

1

5

Sample Output :

12 * 10^1

Explanation :

1*2*3*4*5=120 = 12 * 10^1

Sample Input :

3

10

Sample Output :

18144 * 10^2

Explanation :

3*4*....*10=1814400 =18144 * 10^2

Python | Java

```
def formatProducts(L,H):
    a=1
    for i in range(L,H+1):
        a=a*i
    c=0
    while(True):
        if a%10!=0:
            break
        else:
            c+=1
        a=a//10
    return (str(a)+" * 10^"+str(c))

L=int(input())
H=int(input())
print(formatProducts(L,H))
```

Question 20 : Maximum Toys

Problem Statement – In a toy shop there are a number of toys presented with several various – priced toys in a specific order. You have a limited budget and would like to select the greatest number of consecutive toys that fit within the budget. Given prices of the toys and your budget, what is the maximum number of toys that can be purchased for your child?

Example :

- prices=[1,4,5,3,2,1,6]
- money=6

All subarrays that sum to less than or equal to 6 .

- length 1: [1] [4] [5] [3] [2] [1] [6]
- length 2: [1,4] [3,2] [2,1]
- length 3: [3,2,1]

The longest of these or the maximum number of toys that can be purchased is 3.

Function description

Complete the function

- getMaxToys in the editor below
- getMaxToys has the following parameters:
  int prices[n] : the prices of the various toys.
- int money: the amount of money you can spend on toys

Returns :

Int the maximum number of toys you can purchase

Constraints :

- 1<=n<=10^5
- 1<=price[i]<=100
- 1<=money<=10^6

Sample case

Sample input :

7

1

4

5

3

2

1

6

6

Python Java

```
def getMaxToys(n,arr,money):
    L,H=0,n-1
    while(L<=H):
        if(sum(arr[L:H+1])<=money):
            break
        else:
            if arr[L]>arr[H]:
                L+=1
            else:
                H-=1
    return (H-L+1)

n=int(input())
arr=[]
for i in range(n):
    arr.append(int(input()))
money=int(input())
print(getMaxToys(n,arr,money))
```

Question 21 : Maximum Attendance

Problem Statement – A teacher wants to look at students' attendance data. Given that there is a class , and the teacher has the record of the students present on n days of the month, find the maximum number of consecutive days on which all students were present in the class.

Example

- m=4
- n=7

data=[PPPP, PPPP ,PPPP ,PPAP ,AAPP ,PAPA ,AAAA]

There are 4 students and 7 days attendance data . There are only three days, at the beginning where all students are present. Student 3 is absent on the fourth day , and students 1 and 2 are absent on the fifth day , and students 2 and 4 are absent on the sixth day and all are absent on the last day.

The maximum number of consecutive days on which all the students were present in the class is 3 days long.

Function Description :

Complete the maxConsecutive function in the editor below. The function must return an integer denoting the maximum number of consecutive days where all the students are present in the class.

- int m : the number of students in the class.
- string data[n] : the value of each element data[i] is a
- string where data[i] denotes ith student is present on the ith day.

Constraints :

- 1<=m<=10
- 1<=n<=31
- Each data[i][j]={'P','A'}

Input Format :

- 3
- 4
- PPP
- PPA
- AAP

Sample Output:

1

Explanation :

There is only one day in which all the students are present.

Java

```
import java.util.*;
class MaximumAttendance
{
  public static void main(String[] args)
{
   Scanner sc=new Scanner(System.in);
  int m=sc.nextInt();
  int n=sc.nextInt();
 String arr[]=new String[n];

 for(int i=0;i<n;i++)
    arr[i]=sc.next();

String present="";
int j=0;
while(j<m)
{
   present=present +"P";
   j++;
}
 int max=0;
 int count=0;
for(int i=0;i<n;i++)
 {
    if(arr[i].equals(present))
      count++;
   else
{
  max=Math.max(max,count);
    count=0;
}
 }

System.out.println(max);
}
}
```

Question 22 : Solve equations

Solve the given equations: You will be given an array, and T number of equations. Solve that equation and update the array for every equation you solve

Input Example:

2 3 4 5 1 → input array

3 → number of equations

x*x

x+x

3*x+x

Output:

32 72 128 200 8

Explanation :

- For first case array becomes arr=[4 9 16 25 1]
- For second case array becomes arr=[8 18 32 50 2]
- For third case array becomes arr=[32 72 128 200 8]

Output will be : 32 72 128 200 8

Python

```
n=int(input())
arr=list(map(int,input().split()))
t=int(input())
Lam=[]
for i in range(t):
    Lam.append(input())
for k in range(t):
    for i in range(n):
        x=arr[i]
        arr[i]=eval(Lam[k])
print(*arr)
```

Question 23 : Solve equations

Solve the given equations: There are consecutive lighthouses present in the x axis of a plane.You are given n, which represents the the number of light position and x coordinate array which represent the position of the lighthouses.You have to find maximum lighthouses which have absolute difference less than or equal to 1 between adjacent numbers.

Python Java

```
Solution (in python)
n=int(input())
arr=[]
for i in range(n):
    arr.append(int(input()))
arr.sort()
c=0
count=[]
for i in range(n-1):
    if(abs(arr[i]-arr[i+1])==0 or abs(arr[i]-arr[i+1])==1):
        c+=1
    else:
        count.append(c+1)
        c=0
count.append(c+1)
print(max(count))
```

Question 24: Match

Solve the given equations: The number of matches won by two teams in matches in leagues is given in the form of two lists. For each league score of team B. Compute the total number of matches of team A where team A has won less than or equal to the number of wins scored by team B in that match.

Python Java

Python Java

```
ar=list(map(int,input().split()))
br=list(map(int,input().split()))
ar.sort()
ans=[]
import bisect
for i in br:
    ans.append(bisect.bisect(ar,i))

print(ans)
```

Question 25: jumble the words

Solve the given equations:

Confuse your friends by jumbling the two words given to you. To don't get yourself into confusion follow a pattern to jumble the letters.
Pattern to be followed is , pick a character from the first word and pick another character from the second word.
Continue this process

Take two strings as input , create a new string by picking a letter from string1 and then from string2, repeat this until both strings are finished and maintain the subsequence. If one of the strings is exhausted before the other, append the remaining letters from the other string all at once.

Python Java

```
def newPassword(a,b):
    ans=""
    if(len(a)>len(b)):
        m=len(b)
        x=a[m:]
    if(len(a)<=len(b)):
        m=len(a)
        x=b[m:]
    for i in range(m):
        ans+=a[i]
        ans+=b[i]

    return ans+x

a=input()
b=input()
print(newPassword(a,b))
```

Login/Signup to comment

India.

We help students to prepare for placements with the best study material, online classes, Sectional Statistics for better focus and Success stories & tips by Toppers on PrepInsta.

© 2019 Prep Insta

Disclaimer
Terms and Conditions

Mettl Dashboard
DevSquare Dashboard

Instagram
Linkedin
Telegram