

HackerRank Coding Problems 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

HackerRank Coding Problems with Solutions



-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

HackerRank Coding Problems with Solutions



Explanation 0

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

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

HackerRank Coding Problems with Solutions



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

C++ Code

```
#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}}];
```

HackerRank Coding Problems with Solutions



```
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 – 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

HackerRank Coding Problems with Solutions



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:

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

HackerRank Coding Problems with Solutions



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

HackerRank Coding Problems with Solutions



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

HackerRank Coding Problems with Solutions



100

0835

0900

100

0905

0930

100

0935

1000

100

1005

1030

100

Sample output 2:

0

0

HackerRank Coding Problems with Solutions



Sample Explanation 2:

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

C++

```
#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;
}
```

HackerRank Coding Problems with Solutions



```
}
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])
        {
            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++)
    {
```

HackerRank Coding Problems with Solutions



```
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 3 – 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 i th light has a coverage limit of locations[i] that can range from the position $\max((i - \text{locations}[i]), 1)$ to $\min((i + \text{locations}[i]), n)$ (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 $\min(1+0), 3)$ gives range = 1 to 1

HackerRank Coding Problems with Solutions



For position 2: $\text{locations}[2] = 2$, $\max((2-2),$

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

For position 3: $\text{locations}[3] = 1$, $\max((3-1),$

$1)$ 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 :

- $1 \leq n \leq 10^5$
- $0 \leq \text{locations}[i] \leq \min(n, 100)$ (where $1 \leq i \leq 10^5$)

Sample Input For Custom Testing :

3 ->locations[] size n = 3

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

1 ->Sample Output

Sample Output :

HackerRank Coding Problems with Solutions



1

C++

```
#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;
}
```

HackerRank Coding Problems with Solutions



```
}  
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 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 ($0 \leq n$).

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 $9 + 10 = 29$ units & 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.

HackerRank Coding Problems with Solutions



Returns:

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

Constraints :

- $1 \leq n \leq 10^5$
- $1 \leq \text{arriving Packets}[i] \text{ size} \leq 10^9$

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

```
def largeRepackagedPacket(arr):
```


HackerRank Coding Problems with Solutions



```
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())):
```

HackerRank Coding Problems with Solutions



```
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:

```
**.**
```

```
*...*
```

```
.....
```

HackerRank Coding Problems with Solutions



...

.

C++

```
#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++)
```

HackerRank Coding Problems with Solutions



```
{
    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

HackerRank Coding Problems with Solutions



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++

```
#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);
}
```

HackerRank Coding Problems with Solutions



```
}  
  
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

HackerRank Coding Problems with Solutions



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 :

HackerRank Coding Problems with Solutions



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

```
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++)
```


HackerRank Coding Problems with Solutions



```
{
    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 any point 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.

HackerRank Coding Problems with Solutions



C++

```
#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.

HackerRank Coding Problems with Solutions



Input Format:

Three integers a b and c

Output format:

First print the complex number $a+bi$

Next line print $a + bi + c$ as i^2 .

Next line i^2+a+bi

Sample Input:

4 5 2

Sample Output:

$4 + 5i$

$6 + 5i$

HackerRank Coding Problems with Solutions



10 + 10i

C++

```
#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;
    }
}
```

HackerRank Coding Problems with Solutions



```
};

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 string, 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.

Output Format:

Single character denoting the least frequent character.

Constraints:

HackerRank Coding Problems with Solutions



Length of string $\leq 10^6$

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++

```
#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;}
```

HackerRank Coding Problems with Solutions



}

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:

$2 \leq \text{Length of string} \leq 10^9$

HackerRank Coding Problems with Solutions



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++

```
#include <bits/stdc++.h>
```

```
using namespace std;
```


HackerRank Coding Problems with Solutions



```
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;}

    }
```

HackerRank Coding Problems with Solutions



```
}  
  
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.

HackerRank Coding Problems with Solutions



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:

- $n \leq 10^4$

Sample input :

6

093212

Sample output :

9

Explanation:

HackerRank Coding Problems with Solutions



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

<https://www.freshersnow.com/placement-papers-download/>

C++

```
#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)
```

HackerRank Coding Problems with Solutions



```
{if(sum1>sum) break; sum1+=(i-'0');sum-=i-'0';}  
  
cout<<sum1;  
  
}
```