

While reading questions

1. Extract data (Data type – Storage type) -> memory allocation
2. Diff data types int / float / Bool / String / List-Array

TCS NQT Coding Question 2023 – September Day 1 – Slot 1

Problem Statement –

Joseph is learning digital logic subject which will be for his next semester. He usually tries to solve unit assignment problems before the lecture. Today he got one tricky question. The problem statement is “A positive integer has been given as an input. Convert decimal value to binary representation. Toggle all bits of it after the most significant bit including the most significant bit. Print the positive integer value after toggling all bits”.

Constraints-

$1 \leq N \leq 100$

Example 1:

Input :

10 -> Integer

Output :

5 -> result- Integer

Explanation:

Binary representation of 10 is 1010. After toggling the bits(1010), will get 0101 which represents “5”. Hence output will print “5”.

$2^7 \quad 2^6 \quad 2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$

128 64 32 16 8 4 2 1

0 0 0 0 1 0 1 0 = 10

0 0 0 0 1 1 1 1 = 15

0 0 0 0 0 1 0 1 = 5

1 0 1 0 = 10 => 0 1 0 1 => 5

1 1 1 1 = 15 => 0 0 0 0 => 0

1 1 1 = 7 => 0 0 0 => 0

1 0 0 1 = 9 => 0 1 1 0 => 6

128 64 32 16 8 4 2 1

1 1 0 1 0 1 => 53

Python Solution

```
import math
n=int(input())
k=(1<< int(math.log2(n))+1)-1
print(n^k)
```

C programming solution

```
#include<stdio.h>
#include<math.h>
int main()
```

```
{
    int n;
    scanf("%d", &n);
    int k = (1 << (int)(log2(n) + 1)) - 1;
    printf("%d", n ^ k);
    return 0;
}
```

Java Solution

```
import java.util.*;
class Main
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int z = (1 << (int)(Math.log(n) / Math.log(2)) + 1) -
1;
        System.out.println(n^z);
    }
}
```

TCS NQT Coding Question Day 1 Slot 2 – Question 2

Airport security officials have confiscated several item of the passengers at the security check point. All the items have been dumped into a huge box (array). Each item possesses a certain amount of risk[0,1,2]. Here, the risk severity of the items represent an array[] of N number of integer values. The task here is to sort the items based on their levels of risk in the array. The risk values range from 0 to 2.

Example :

Input :

7 -> Value of N

[1,0,2,0,1,0,2]-> Element of arr[0] to arr[N-1], while input each element is separated by new line.

Output :

0 0 0 1 1 2 2 -> Element after sorting based on risk severity

Example 2:

input : 10 -> Value of N

[2,1,0,2,1,0,0,1,2,0] -> Element of arr[0] to arr[N-1], while input each element is separated by a new line.

Output :

0 0 0 0 1 1 1 2 2 2 ->Elements after sorting based on risk severity.

Explanation:

In the above example, the input is an array of size N consisting of only 0's, 1's and 2s. The output is a sorted array from 0 to 2 based on S severity.

0 1 2 3 4 5 6 7 8 9

[2,1,0,2,1,0,0,1,2,0]

zeroCount = 4 onesCount = 3 twosCount = 3

You can also use predefined Python functions for solutions (Using predefined function may leads to time limit exceeded error for few test cases)

```
n = int(input())
arr = []
for i in range(n):
    arr.append(int(input()))
for i in sorted(arr):
    print(i, end=" ")
```

Use any sorting algorithms like Selection sort, Quick Sort, Merge Sort . . .

```
class HelloWorld {
    public static void sort(int arr[], int n){
        for(int i=0 ; i<n-1 ; i++){
            int minIdx= i;
            for( int j=i+1;j<n;j++){
                if(arr[j]<arr[minIdx])
                    minIdx=j;
            }
        }
    }
}
```

```

        }
        int temp;
        if(minIdx!=i){
            temp=arr[minIdx];
            arr[minIdx]=arr[i];
            arr[i]=temp;
        }
    }
}

public static void countSort(int arr[],int n){
    int z=0, o=0, t=0;
    for(int i=0;i<n ;i++){
        if(arr[i]==0){
            z++;
        }
        else if(arr[i]==1){
            o++;
        }
        else{
            t++;
        }
    }
    for(int i=0;i<z;i++){
        System.out.println("0");
    }
    for(int i=0;i<o;i++){
        System.out.println("1");
    }
    for(int i=0;i<t;i++){
        System.out.println("2");
    }
}

public static void main(String[] args) {
    int n=10;
    System.out.println("Enter no of items");
    int arr1[]={2,1,0,2,1,0,0,1,2,0};
    countSort(arr1,n);
    int arr2[]={2,1,0,2,1,0,0,1,2,0};
    sort(arr2,n);
    for(int i=0;i<n;i++){
        System.out.println(arr2[i]);
    }
}
}

```

TCS NQT Coding Question Day 2 Slot 1 – Question 1

Given an integer array Arr of size N the task is to find the count of elements whose value is greater than all of its prior elements.

Note : 1st element of the array should be considered in the count of the result.

For example,

Arr[]={7,4,8,2,9}

As 7 is the first element, it will consider in the result.

8 and 9 are also the elements that are greater than all of its previous elements.

Since total of 3 elements is present in the array that meets the condition.

Hence the output = 3.

Example 1:

Input

5 -> Value of N, represents size of Arr

7-> Value of Arr[0]

4 -> Value of Arr[1]

8-> Value of Arr[2]

2-> Value of Arr[3]

9-> Value of Arr[4]

Output :

3

Example 2:

5 -> Value of N, represents size of Arr

3 -> Value of Arr[0]

4 -> Value of Arr[1]

5 -> Value of Arr[2]

8 -> Value of Arr[3]

9 -> Value of Arr[4]

Output :

5

Constraints

1<=N<=20

1<=Arr[i]<=10000

0 1 2 3 4 -> 1 2 3 4 -> 2 3 4 -> 3 4 -> 4

```
For( i=0 ; i<n ;i++){ // i= 0 1 2 3 4
```

```
    For( j=i; j<n ;j++){
```

```
        Printf("%d", j);
```

```
    }
```

```
}
```

0 1 2 3 4 -> 0 1 2 3 4 -> 0 1 2 3 4 -> 0 1 2 3 4 -> 0 1 2 3 4

```
For( i=0 ; i<n ;i++){ // i= 0 1 2 3 4
```

```

        For( j=0; j<n ;j++){
            Printf("%d", j);
        }
    }
}

```

n=5

```

* * * * * -> r1 5e -> r0 5e
* * * * * -> r2 5e -> r1 5e
* * * * * -> r3 5e -> r2 5e
* * * * * -> r4 5e -> r3 5e
* * * * * -> r5 5e -> r4 5e

```

c1 c2 c3 c4 c5 r

```

11 12 13 14 15 1
21 22 23 24 252
31 32 33 34 353
41 42 43 44 454
51 52 53 54 555

```

```

j->  0 1 2 3 4 5      j<i
i      -----
1      *              0<1
2      * *            0<2 1<2
3      * * *          0<3 1<3 2<3
4      * * * *
5      * * * * *

```

```

j->  1 2 3 4 5      j<=i
i      -----
1      *              1<=1
2      * *            1<=2 2<=2
3      * * *          1<=3 2<=3 3<=3
4      * * * * 1<=4 2<=4 3<=4 4<=4
5      * * * * *      . . . . .

```

```

j->  0 1 2 3 4 5      j<i
i      -----
5      * * * * *      0<5 1<5 2<5 3<5 4<5
4      * * * * 0<4 1<4 2<4 3<4
3      * * *          0<3 1<3 2<3

```

```

2      * *      0<2 1<2
1      *      0<1

```

```

j->  0 1 2 3 4      j<i
i      -----
1      *      0<1
2      * *      0<2
3      * * *
4      * * * *
5      * * * * *

```

```

-----
*      41 8 4-[ ] 1-[* ]
* * 32 6 3-[ ] 2-[* ]
* * * 23 4 2-[ ] 3-[* ]
* * * * 14 2 1-[ ] 4-[* ]
* * * * * 05 0 0-[ ] 5-[* ]

```

```

i      -----      n-i
1      *      5-1=4
2      * * 5-2=3
3      * * * 5-3=2
4      * * * * 5-4=1
5      * * * * * 5-5=0

```

```

* 6[ ] 1[* ]
* * 5[ ] 2[* ]
* * * 4[ ] 3[* ]
* * * * 3[ ] 4[* ]
* * * * * 2[ ] 5[* ]
* * * * * * 1[ ] 6[* ]
* * * * * * * 0[ ] 7[* ]

```

5

12345
1234
123
12
1

12345
23451
34512
45123
51234

5

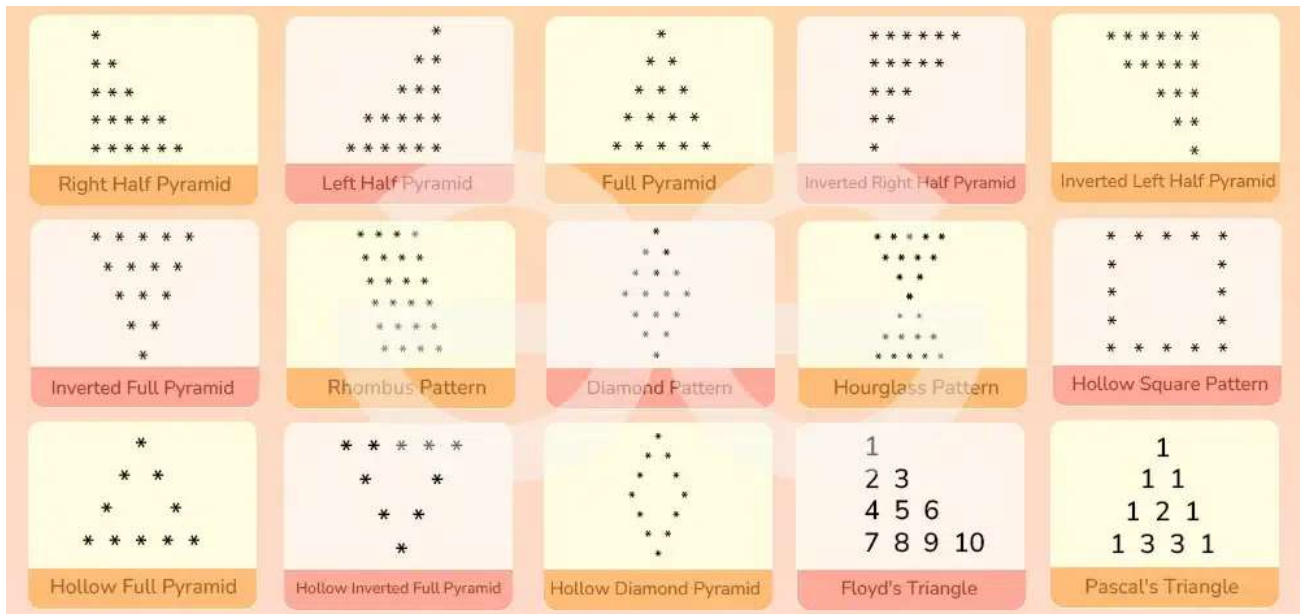
A B C D E

5

A B C D E
B C D E A
C D E A B
D E A B C
E A B C D

```
int n=5,r,c;  
    for(r=1;r<=n;r++){  
        for(c=r ; c<=n ; c++){  
            System.out.print(c);  
        }  
        for(c=1 ; c<r ; c++){  
            System.out.print(c);  
        }  
        System.out.println(); // printf(
```


Code this patterns



Input : 5

Output:

```

* * * * * * * * * *
* * * *   * * * *
* * *     * * *
* *       * *
*         *
*         *
* *      * *
* * *    * * *
* * * *  * * * *
* * * * * * * * *

```

Input : n = 4

Output :

```

1
5 2
8 6 3
10 9 7 4

```

A

AB

A B

A B

A B

A B

ABCDEFG

Input 7

TCS NQT Coding Question 2023 – September Day 1 – Slot 1

Problem Statement –

A chocolate factory is packing chocolates into the packets. The chocolate packets here represent an array of N number of integer values. The task is to find the empty packets(0) of chocolate and push it to the end of the conveyor belt(array).

Example 1 :

N=8 and arr = [4,5,0,1,9,0,5,0].

There are 3 empty packets in the given set. These 3 empty packets represented as 0 should be pushed towards the end of the array

Input :

8 – Value of N

[4,5,0,1,9,0,5,0] – Element of arr[0] to arr[N-1], While input each element is separated by newline

Output:

4 5 1 9 5 0 0 0

Example 2:

Input:

6 — Value of N.

[6,0,1,8,0,2] – Element of arr[0] to arr[N-1], While input each element is separated by newline

Output:

6 1 8 2 0 0

TCS NQT Coding Question Day 1 Slot 2 – Question 1

Jack is always excited about Sunday. It is favourite day, when he gets to play all day. And goes to cycling with his friends.

So every time when the month starts he counts the number of Sundays he will get to enjoy.

Considering the month can start with any day, be it Sunday, Monday.... Or so on.

Count the number of Sunday Jack will get within n number of days.

Example 1:

Input

mon-> input String denoting the start of the month.

13 -> input integer denoting the number of days from the start of the month.

Mon	0	1	8
Tue		2	9
Wed		3	10
Thu		4	11
Fri		5	12
Sat		6	13
Sun	0	7	14

www.hackerrank.com/learn-algorithms

Output :

2 -> number of days within 13 days.

Explanation:

The month starts with mon(Monday). So the upcoming Sunday will arrive in next 6 days. And then next Sunday in next 7 days and so on.

Now total number of days are 13. It means 6 days to first Sunday and then remaining 7 days will end up in another Sunday. Total 2 Sundays may fall within 13 days.

```
import java.io.*;
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String ips = sc.next();
        int n = sc.nextInt();
        int i,num=1,rem;
        String arr[] = {"mon", "tue", "wed", "thu", "fri", "sat", "sun"};
```

```

        for(i=0;i<7;i++){
            if(arr[i].equals(ips)){
                break;
            }
        }
        rem=6-i;
        n=n-rem;
        if(n>0){
            num=num+n/7;
        }
        System.out.println(num);
    }
}

```

```

#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
    int n,i,j=0,*a,ip;
    scanf("%d",&n);
    //int a[n]; //{ , , , , , , , }
    a=(int*)malloc(sizeof(int)*n);
    for(i=0;i<n;i++){ // 0 1 2 3 4 5 6 7
        scanf("%d",&ip);
        if(ip!=0){
            a[j++]=ip;
        }
    }
    for(i=0;i<n;i++){ // 0 1 2 3 4 5 6 7
        printf("%d ",a[i]);
    }
    return 0;
}

```

N=3

9 => 1001

1 => 0001

1000 => 8

9 => 1001

0001 => 1

1	01
2	10
<hr/>	
	11
3	11
<hr/>	
	00
2	10
<hr/>	
	10
3	11
<hr/>	
	01
1	01
<hr/>	
	00
3	10
<hr/>	
	10 => 3

```
import java.io.*;
import java.util.*;

public class Solution {
    public static int oddOccurance(int n, int arr[]) {
        int xor = 0;
        for(int i : arr) {
            xor ^= i;
        }
        //System.out.println(xor);
        return xor;
    }
    public static void main(String[] args) {
        int arr[] = {33,22,45,22,33,22,45,22,45};
        System.out.println(oddOccurance(9,arr));
    }
}
```

TCS NQT Coding Question Day 2 Slot 1 – Question 2

A supermarket maintains a pricing format for all its products. A value N is printed on each product. When the scanner reads the value N on the item, the product of all the digits in the value N is the price of the item. The task here is to design the software such that given the code of any item N the product (multiplication) of all the digits of value should be computed(price).

Example 1:

Input :

5244 -> Value of N

Output :

160 -> Price

Explanation:

From the input above

Product of the digits 5,2,4,4

$$5*2*4*4= 160$$

Hence, output is 160.

TCS Coding Question Day 1 Slot 2 – Question 2

A parking lot in a mall has $R \times C$ number of parking spaces. Each parking space will either be empty(0) or full(1). The status (0/1) of a parking space is represented as the element of the matrix. The task is to find index of the row(R) in the parking lot that has the most of the parking spaces full(1).

Note :

$R \times C$ - Size of the matrix

Elements of the matrix M should be only 0 or 1.

Example 1:

Input :

3 -> Value of R(row)

3 -> value of C(column)

[0 1 0 1 1 0 1 1 1] -> Elements of the array $M[R][C]$ where each element is separated by new line.

Output :

3 -> Row 3 has maximum number of 1's

Example 2:

input :

4 -> Value of R(row)

3 -> Value of C(column)

[0 1 0 1 1 0 1 0 1 1 1 1] -> Elements of the array $M[R][C]$

Output :

4 -> Row 4 has maximum number of 1's

0 1 0

1 1 0

1 1 1

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;
```

```

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int r,c,max=0,rs=0,ind=0;
        r = sc.nextInt();
        c = sc.nextInt();
        int arr[][]=new int[r][c];

        for(int i=0; i<r; i++){
            for(int j=0; j<c; j++){
                arr[i][j]=sc.nextInt();
            }
        }
        for(int i=0; i<r; i++){
            for(int j=0; j<c; j++){
                rs+=arr[i][j];
            }
            if(max<rs){
                max=rs;
                ind=i;
            }
        }
        System.out.println(ind+1);
    }
}

```

Max=3 maxR = 3

R=3

C=0 1 2 < 3

rowSum = 3

0 1 0 1 1 0 1 0 1 1 1 1

```

import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int r,c,max=0,rs=0,ind=0,ipn=0;
        r = sc.nextInt();
        c = sc.nextInt();

        for(int i=0; i<r; i++){
            rs=0;
            for(int j=0; j<c; j++){
                rs+=sc.nextInt();
            }
        }
    }
}

```



```
        }
        if (max < rs) {
            max = rs;
            ind = i;
        }
    }
    System.out.println(ind+1);
}
}
```

TCS Coding Question Day 2 Slot 1 – Question 1

A party has been organised on cruise. The party is organised for a limited time(T). The number of guests entering (E[i]) and leaving (L[i]) the party at every hour is represented as elements of the array. The task is to find the maximum number of guests present on the cruise at any given instance within T hours.

Example 1:

Input :

- 5 -> Value of T
- [7,0,5,1,3] -> E[], Element of E[0] to E[N-1], where input each element is separated by new line
- [1,2,1,3,4] -> L[], Element of L[0] to L[N-1], while input each element is separate by new line.

Output :

8 -> Maximum number of guests on cruise at an instance.

Explanation:

1st hour:

Entry : 7 Exit: 1

No. of guests on ship : 6

2nd hour :

Entry : 0 Exit : 2

No. of guests on ship : $6-2=4$

Hour 3:

Entry: 5 Exit: 1

No. of guests on ship : $4+5-1=8$

Hour 4:

Entry : 1 Exit : 3

No. of guests on ship : $8+1-3=6$

Hour 5:

Entry : 3 Exit: 4

No. of guests on ship: $6+3-4=5$

Hence, the maximum number of guests within 5 hours is 8.

Example 2:

Input:

4 -> Value of T

[3,5,2,0] -> E[], Element of E[0] to E[N-1], where input each element is separated by new line.

[0,2,4,4] -> L[], Element of L[0] to L[N-1], while input each element in separated by new line

Output:

6

Cruise at an instance

Explanation:

Hour 1:

Entry: 3 Exit: 0

No. of guests on ship: 3

Hour 2:

Entry : 5 Exit : 2

No. of guest on ship: $3+5-2=6$

Hour 3:

Entry : 2 Exit: 4

No. of guests on ship: $6+2-4=4$

Hour 4:

Entry: 0 Exit : 4

No. of guests on ship : $4+0-4=0$

Hence, the maximum number of guests within 5 hours is 6.

The input format for testing

The candidate has to write the code to accept 3 input.

First input- Accept value for number of T(Positive integer number)

Second input- Accept T number of values, where each value is separated by a new line.

Third input- Accept T number of values, where each value is separated by a new line.

The output format for testing

The output should be a positive integer number or a message as given in the problem statement(Check the output in Example 1 and Example 2)

Constraints:

- $1 \leq T \leq 25$
- $0 \leq E[i] \leq 500$
- $0 \leq L[i] \leq 500$

```
import java.io.*;
```

```

import java.util.*;
import java.text.*;
import java.math.*;

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n,sum=0,max=0,hr=0;
        n = sc.nextInt();
        int e[]= new int[n];
        int l[]= new int[n];

        for(int i=0; i<n; i++){
            e[i]=sc.nextInt();
        }
        for(int i=0; i<n; i++){
            l[i]=sc.nextInt();
        }
        for(int i=0; i<n; i++){
            sum+=e[i]-l[i];
            if(sum>max){
                max=sum;
                hr=i+1;
            }
            //max = Math.max(sum,max);
        }
        System.out.println("Max gusets present is "+max+ " at "+hr+"
hour");
    }
}

```

TCS Coding Question Day 2 Slot 1 – Question 2

At a fun fair, a street vendor is selling different colours of balloons. He sells N number of different colours of balloons (B[]). The task is to find the colour (odd) of the balloon which is present odd number of times in the bunch of balloons.

Note: If there is more than one colour which is odd in number, then the first colour in the array which is present odd number of times is displayed. The colours of the balloons can all be either upper case or lower case in the array. If all the inputs are even in number, display the message “All are even”.

Example 1:

- 7 -> Value of N
- [r,g,b,b,g,y,y] -> B[] Elements B[0] to B[N-1], where each input element is separated by new line.

Output :

- r -> [r,g,b,b,g,y,y] -> “r” colour balloon is present odd number of times in the bunch.

Explanation:

From the input array above:

- r: 1 balloon
 - g: 2 balloons
 - b: 2 balloons
 - y : 2 balloons
- Hence , r is only the balloon which is odd in number.

Example 2:

Input:

- 10 -> Value of N
- [a,b,b,b,c,c,c,a,f,c] -> B[], elements B[0] to B[N-1] where input each element is separated by new line.

Output :

b-> ‘b’ colour balloon is present odd number of times in the bunch.

Explanation:

From the input array above:

- a: 2 balloons
- b: 3 balloons
- c: 4 balloons
- f: 1 balloons

Here, both 'b' and 'f' have odd number of balloons. But 'b' colour balloon occurs first.
Hence, b is the output.

Input Format for testing

The candidate has to write the code to accept: 2 input

- First input: Accept value for number of N(Positive integer number).
- Second Input : Accept N number of character values (B[]), where each value is separated by a new line.

Output format for testing

The output should be a single literal (Check the output in example 1 and example 2)

Constraints:

- $3 \leq N \leq 50$
- $B[i] = \{ \{a-z\} \text{ or } \{A-Z\} \}$

Problem can be solved using LinkedHashMap sample code to understand

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) throws IOException{

        InputStreamReader isr = new InputStreamReader(System.in);
        BufferedReader br = new BufferedReader(isr);

        LinkedHashMap<String,Integer> hm = new
LinkedHashMap<String,Integer>();
        hm.put(br.readLine(),1);
        hm.put(br.readLine(),1);
        hm.put("b",1);

        System.out.println(hm);
        System.out.println(hm.containsKey("a"));
        System.out.println(hm.containsKey("c"));

    }
}
```