# Zoho Round 1

Aptitude, puzzle related questions and C programming questions (Fill in the blanks)

1. Writing Section

- Algebra
- Number System
- Progression
- Averages
- Logarithms
- Time, Distance & Speed
- Height & Distance
- Proportion & Ratio
- Profit & Loss
- Mensuration & Geometry
- Data Sufficiency
- Probability
- Permutations & Combinations
- Simplification
- Simple & Compound Interest

2. Technical Assessment Section

- Coding Decoding
- Distances & Directions
- Ranking & Ordering
- Data Sufficiency
- Non-Verbal Reasoning
- Seating Arrangements.
- Puzzles
- Syllogisms
- Blood Relations
- Classification

- Logical Deductions
- Coded Inequalities
- Venn Diagrams

3. Computer Programming & Advanced Programming Test

Computer Programming:

- Strings
- Pointers
- Loops
- Matrix
- Nested & Complex Loops
- Dynamic Memory Allocation
- Control Flows

## Zoho Round 2

**Concepts to Learn:**

Basics of OOPs

Numbers

Strings

Data Structures

Related programming questions

Pattern printing programs

**Example programs:**

1) Pattern

1

2 4

3 5 7

```
import java.util.*;
import java.io.*;
class Main {
   public static void main(String[] args) {
      Scanner sc =new Scanner(System.in);
      int n2=0;
      int n = sc.nextInt();
      for(int i=1;i<=n;i++){
         n2=i;
         for(int j=0;j<i;j++){
            System.out.print(n2+" ");
            n2=n2+2;
         }
         System.out.println();
      }
   }
}
```

2) Sorting with factors

```
import java.util.*;
import java.io.*;
class Main {
   static int factor(int x){
      int c=0;
      for(int i=1;i<x;i++){
         if(x%i==0){
            c++;
```

```
        }
      }
      return c;
   }
   public static void main(String[] args) {
      Scanner sc =new Scanner(System.in);
      int arr[] = new int[100];
      int n =5, t, c=0;
      for(int i=0;i<n;i++){
         arr[i] = sc.nextInt();
      }
      for(int i=0;i<n;i++){
         for(int j=i;j<n;j++){
            if(factor(arr[i])>factor(arr[j])){
               t = arr[j];
               arr[j] = arr[i];
               arr[i] = t;
            }
         }
      }
      for(int i=0;i<n;i++){
         System.out.print(arr[i]+" ");
      }
   }
}
```

3) Vowels in a Sentence

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

```
import java.io.*;
class Main {
   public static void main(String[] args) {
      Scanner sc =new Scanner(System.in);
      String s;
      String res[] = new String[100];
      int c=0,r=0;
      s =sc.nextLine();
      for(int i=0;i<s.length();i++){
         if(s.charAt(i) == 'a'||s.charAt(i) == 'e'||s.charAt(i) == 'i'||s.charAt(i) == 'o'||s.charAt(i)
== 'u'||s.charAt(i) == 'A'||s.charAt(i) == 'E'||s.charAt(i) == 'I'||s.charAt(i) == 'O'||s.charAt(i)
== 'U'){
            c++;
            res[r] = s.charAt(i) + "";
            System.out.print(res[r]+" ");
            r++;
         }
      }
      System.out.println(" = "+c);
   }
}
```

4) Vowels of String with non repeating Vowels

```
import java.util.*;
import java.io.*;
class Main {
   public static void main(String[] args) {
      Scanner sc =new Scanner(System.in);
      String s;
```

```
        char res[] = new char[100];
        int count=0,c=0,r=0;
        s =sc.nextLine();
        for(int i=0;i<s.length();i++){
           if(s.charAt(i) == 'a'||s.charAt(i) == 'e'||s.charAt(i) == 'i'||s.charAt(i) == 'o'||s.charAt(i)
== 'u'||s.charAt(i) == 'A'||s.charAt(i) == 'E'||s.charAt(i) == 'I'||s.charAt(i) == 'O'||s.charAt(i)
== 'U'){
              c++;
              res[r] = s.charAt(i);
              System.out.print(res[r]+" ");
              r++;
           }
        }
        System.out.println(" = "+c);

        for(int i=0;i<res.length;i++){
           for(int j=i+1;j<res.length;j++){
              if(res[i] == res[j]){
                 count++;
              }
           }

           if(count == 0){
              System.out.print(res[i]+" ");
           }
           count = 0;
        }
     }
}
```

5) Expression

567*+

5*6+7 = 37

```
import java.util.*;
import java.io.*;
class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String s = sc.nextLine();
        int a[] = new int[100];
        int c=0,c2=0,t=0;
        for(int i=0;i<s.length();i++){
            if(Character.isDigit(s.charAt(i))){
                a[c] = Integer.parseInt(s.charAt(i) + "");
                c++;
                if(Character.isDigit(s.charAt(i+1))){}else{
                    c=0;
                    t=a[c];
                }
            }
            else{
                switch(s.charAt(i)){
                    case '+':
                        t += a[c+1];
                        c++;
                        break;
                    case '-':
```

```
                t -= a[c+1];
                c++;
                break;
            case '*':
                t *= a[c+1];
                c++;
                break;
            default:
                System.out.println(s.charAt(i));
          }
        }
      }
      System.out.println(t);
   }
}
```

6) Frequency of Alphabets

```
import java.util.*;
import java.io.*;
class Main {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      String s = sc.nextLine();
      int res[] = new int[26];
      int t;
      char c;
      for(int i=0;i<26;i++){
         res[i] = 0;
```

```
        }
        for(int i=0;i<s.length();i++){
           t = s.charAt(i) - 'A';
           res[t] += 1;
        }
        for(int i=0;i<26;i++){
           if(res[i]!=0){
              c = (char)(i+65);
              System.out.println(c+" = "+ res[i]);
           }
        }
    }
}
```

7) Stack expression

56+

5+6=11

```
import java.util.*;
import java.io.*;
public class Main
{
        public static void main(String[] args) {
            Scanner sc = new Scanner(System.in);
                Stack<String> numberStack = new Stack<String>();
                Stack<String> s2 = new Stack<String>();
                String str = sc.next();
                int a,b,c=0;
```

```
        String t;
        for(int i=0;i<str.length();i++){
            numberStack.push(str.charAt(i)+"");
            s2.push(str.charAt(i)+"");
        }
        for(int i=0;i<str.length();i++){

            if(Character.isDigit(numberStack.peek().charAt(0))){
              //System.out.println(numberStack.peek());
              s2.remove(s2.firstElement());
            }
            else{
               numberStack.pop();
            }
        }
        c = Integer.parseInt(numberStack.firstElement());
        numberStack.remove(numberStack.firstElement());
        int size = numberStack.size();
        for(int i=0;i<size;i++){
            b = Integer.parseInt(numberStack.firstElement());
            numberStack.remove(numberStack.firstElement());
            switch(s2.firstElement()){
    case "+":
       c += b;
       t = s2.firstElement();
       s2.remove(t);
       break;
    case "-":
       c -= b;
```

```
                t = s2.firstElement();
                s2.remove(t);
                break;
            case "*":
                c *= b;
                t = s2.firstElement();
                s2.remove(t);
                break;
            default:
                System.out.println("wrong");
          }
                }
                System.out.println(c);
        }
}
```

8) legal paranthesis

```
import java.util.*;
import java.io.*;
public class Main
{
        public static void main(String[] args) {
           Scanner sc = new Scanner(System.in);
                String s = sc.next(); // input string
                char a ='(',b=')';
                int t1=0,t2=0,c=0; //count of the balance
                for(int i=0;i<s.length();i++){
                   if(s.charAt(i) == a) t1++;
```

```
                else {
                    t2++;
                    if(t2>t1) c++; //c is check for Illegal closing Brackets
                }
            }
            if(c==0 && (t1-t2)==0){
                System.out.println("Legal Brackets");
            }
            else{
                System.out.println("Illegal Brackets");
            }
        }
}
```

9) Legal Parathesis using Stack

```
import java.util.*;
import java.io.*;
public class Main
{
        public static void main(String[] args) {
            Scanner sc = new Scanner(System.in);
            Stack<String> stringStack = new Stack<String>();
            int flag=0;
            String c;
                String s = sc.next(); // input string
                for(int i=0;i<s.length();i++){
                    c = s.charAt(i)+""; //brackets one by one
```

```
                    if(c.equals(")")){
                        if(stringStack.empty() == true){
                            flag=1;  // 1 determines its false
                            stringStack.push(c);
                        }
                        else{
                            stringStack.remove("(");
                        }
                    }
                    else{
                        stringStack.push(c);
                    }
                }
                if(stringStack.empty() == true){  // checking whether the brackets are equal or
not
                    flag=0;
                }else{
                    flag=1;  // 1 determines its false
                }
                if(flag==0) System.out.println("Legal Paranthesis");
                else System.out.println("Illegal Paranthesis");
        }
}
```

10) Frequency of Alphabets using HashMap

```
import java.util.*;
import java.io.*;
public class Main
{
```

```
        public static void main(String[] args) {
           Scanner sc = new Scanner(System.in);
           char c;
           HashMap<Character, Integer> freq = new HashMap<Character, Integer>(26);
           String s = sc.nextLine();  //input
           for(int i=0;i<s.length();i++){
              c = s.charAt(i);
              freq.putIfAbsent(c,0); //create key with value 0 (allocation)
           }
           for(int i=0;i<s.length();i++){
              c = s.charAt(i);
              freq.put(c,freq.get(c)+1); //increament of value when it it repeated
           }
          System.out.println(freq);
        }
}
```

11) Insert, Delete, Search fnc

```
import java.util.*;
import java.io.*;
public class Main
{
  static void Insert (int x, int n, int arr[])
  {
    int index = 0, c = 0;
    for (int i = 0; i < n; i++)
      {
          if (arr[i] > x && c == 0)
```

```
            {
              index = i;
              c++;
            }
    }
  n += 1;
  if (index == 0 && c==0)
    index = n - 1;
  for (int i = n - 1; i > index; i--)
    {
          arr[i] = arr[i - 1];
    }
  arr[index] = x;
  for (int i = 0; i < n; i++)
    {
          System.out.print (arr[i] + " ");
    }
}
static void Delete (int x, int n, int arr[])
{
  int c = 0, index = 0;
  for (int i = 0; i < n; i++)
    {
          if (arr[i] == x)
            {
              c++;
              index = i;
            }
    }
```

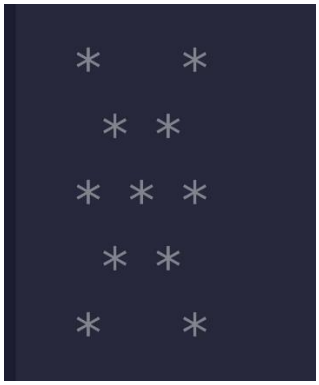
```
 if (c != 0)
  {
       n -= 1;
       for (int i = index; i < n; i++)
        {
          arr[i] = arr[i + 1];
        }
       for (int i = 0; i < n; i++)
        {
          System.out.print (arr[i] + " ");
        }
  }
 else
  {
       System.out.println ("element not found in the array");
       for (int i = 0; i < n; i++)
        {
          System.out.print (arr[i] + " ");
        }
  }
}
static void Search (int x, int n, int arr[])
{
 int c = 0;
 for (int i = 0; i < n; i++)
  {
       if (arr[i] == x)
        c++;
  }
```

```
  if (c == 0)
    System.out.println ("not found in the array");
  else
    System.out.println ("found in the array");
}
public static void main (String[]args)
{
  Scanner sc = new Scanner (System.in);
  int num = 0, n = 5;
  int arr[] = new int[10];
  for (int i = 0; i < n; i++)
    {
        arr[i] = sc.nextInt ();
    }
  String s = sc.next ();

  char func = s.charAt (0);
  for (int i = 1; i < s.length (); i++)
    {
        num = num * 10 + Integer.parseInt (s.charAt (i) + "");
    }
  switch (func)
    {
    case 'I':
        Insert (num, n, arr);
        break;
    case 'D':
        Delete (num, n, arr);
        break;
```

```
    case 'S':
        Search (num, n, arr);
        break;
    default:
        break;
    }
 }
}
```

**Pattern printing:**

```
Input : n = 5
Output :
    0
   101
  21012
 3210123
432101234

Input : n = 7
Output :
      0
     101
    21012
   3210123
  432101234
 54321012345
6543210123456
```



1. Print the word in below Format

Input : water

```
w       w       w
    a   a   a
w   a   t   e   r
    e   e   e
r       r       r
```

**3. Find the longest increasing sub-sequence in Array.**

```
Input : {1,5,3,7}
Output: {1,5,7} or {1,3,7}

Input : {10,22,9,33,21,50,41}
Output: {10,22,33,50}
```

**2. Program to check whether the given matrix is an upper triangular or lower triangular.**

The constraint was each element should be visited only once.

```
Input:
4 6 1 4
0 3 5 9
0 0 6 2
0 0 0 8

Output: Upper Triangular Matrix

Input:
1 0
1 1
Output: Lower Triangular Matrix
```

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

**Q4**. Given a 9×9 Sudoku, we have to evaluate it for its correctness. We have to check both the submatrix correctness and the whole sudoku correctness.

**Example Input/Output 1:**

Input:

CRY

Output:

**R

*RY

RYC

**Example Input/Output 2:**

Input:

PROGRAM

Output:

******G

*****GR

****GRA

***GRAM

**GRAMP

*GRAMPR

GRAMPRO

**Q3.** Save the string "WELCOMETOZOHOCORPORATION" in a two-dimensional array and search for a substring like "too" in the two-dimensional string both from left to right and from top to bottom. W E L C O M E T O Z O H O C O R P O R A T I O N And print the start and ending index as Start index: <1,2> End index: <3, 2>

```
Input : n = 5
Output :
    0
   101
  21012
 3210123
432101234

Input : n = 7
Output :
      0
     101
    21012
   3210123
  432101234
 54321012345
6543210123456
```

2. Write a program to sort the elements in odd positions in descending order and elements in ascending order

```
Eg 1: Input: 13,2 4,15,12,10,5
        Output: 13,2,12,10,5,15,4
Eg 2: Input: 1,2,3,4,5,6,7,8,9
        Output: 9,2,7,4,5,6,3,8,1
```

## Zoho Round 3

**Concepts to learn:**

Data structures in detail

Dynamic programming

Application development example:

Train ticket

- Booking
- Cancellation
- Chart Preparation
- Checking of Availability

**Example programs:**

1 Blood Relations(cousin)

```
import java.util.*;
import java.io.*;
public class Main
{
        public static void main(String[] args) {
           Scanner sc= new Scanner(System.in);
           int n=6,c=0,c2=0,c3=0;
           String dataString[][] = {{"Akash", "M", "Suresh", "Leela"},{"Meena", "F", "Suresh",
"Leela"},
           {"Priya", "F","Shyna", "Lalubai"},
           {"Suresh", "M", "Lalu", "Lali"},
           {"Shyna", "M", "Lalu", "Lali"},
           {"Kishore", "M", "Lalu", "Lali"}};
           int count[] = {0,0,0,0};
           String cousins[] = new String[5];
           int cousinscount=0;
           String s = sc.next();
      // for(int i=0;i<n;i++){
      //     dataString[i] = sc.nextLine().split(" ");
      // }
         for(int j=0;j<n;j++){
            if(s.equals(dataString[j][2])||s.equals(dataString[j][3])){
```

```
        // System.out.print(dataString[j][0]+" ");
        for(int k=0;k<n;k++){
           if(dataString[j][0].equals(dataString[k][2])){
              c2++;
              cousins[cousinscount] = dataString[k][0];
              cousinscount++;
           }
        }

        if(c2==0){
              // System.out.print(dataString[j][0]+"( has no Son/Daugther ) ");
              // System.out.println();
           }else{
              // System.out.print("s/o ");
              // System.out.print(dataString[j][0]+"( has Son/Daugther ) ");
              // System.out.println();
              c3++;
           }
        c2=0;
        c++;
     }
  }
  if(c==0){
     // System.out.print(s+" has no Son/Daugther"+".");
  }
  else if(c==1){
     // System.out.print("is the Son/Daugther of "+s+".");
  }
  else{
```

```
            // System.out.print("are the Son's/Daugther's of "+s+".");
          }
       for(int i=0;i<cousins.length;i++){
          if(cousins[i]!= null) System.out.print(cousins[i]+" ");
          for(int j=0;j<n;j++){
             if(cousins[i].equals(dataString[j][0])){
                if(cousins[j]!=null) System.out.print(dataString[j][1]);
                System.out.println();
             }
          }

       }
       // System.out.print("are cousins");
           }
}
```

2

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

class Items
{
  String name;
  String item[] = new String[10];
  int qn[] = new int[10];
  int rate[] = new int[10];
  int tot = 0;
```

```
  int n;
  static int condition = 0;
  static int search = 1;
  static int id = 0;
  int subtot[] = new int[10];
  public void itemlist (int i, String it[], int q[], int r[])
  {
    item[i] = it[i];
    qn[i] = q[i];
    rate[i] = r[i];
  }
}

class Invoice extends Items
{
  void input (String nm, String it[], int q[], int r[], int list, int c)
  {
    name = nm;
    condition = c;
    for (int i = 0; i < list; i++)
      {
            itemlist (i, it, q, r);
            n = i + 1;
            subtot[i] = qn[i] * rate[i];
            tot += subtot[i];
      }
  }

  void display ()
```

```
 {
  id++;
  if (condition != id)
   {
        System.out.println ("Invoice");
        System.out.println ("*******");
        System.out.println (id + " " + name);
        for (int i = 0; i < n; i++)
         {
           System.out.println ((i + 1) + ". " + item[i] + " " + qn[i] + " " +
                                    rate[i] + " " + subtot[i]);
         }
        System.out.println ("Total : " + tot);
        System.out.println ();
   }
  else
   {
        System.out.println (id + " has been Deleted");
   }
 }
}

public class Main
{
  public static void main (String[]args)
  {
   Scanner sc = new Scanner (System.in);
   Invoice i1 = new Invoice ();
   Invoice i2 = new Invoice ();
```

```
Invoice i3 = new Invoice ();
int num = 0, lcount = 2;
int condition = 0;
int search = 1;
System.out.println("Enter I for Insert");
System.out.println("Enter D for Delete");
System.out.println("Enter S for Search");
//item 1
String name1 = "Faheem";
String item1[] = { "Tea", "Coffe" };
int qn1[] = { 2, 2 };
int rate1[] = { 10, 20 };
//item 2
String name2 = "Bala";
String item2[] = { "Juice", "Coffe" };
int qn2[] = { 3, 1 };
int rate2[] = { 35, 20 };
String sinput = sc.next ();
char func = sinput.charAt (0);
switch (func)
 {
 case 'I':
       String item3[] = new String[2];
       int q3[] = new int[2];
       int rate3[] = new int[2];
       System.out.println ("Name :");
       String name3 = sc.next ();
       System.out.println ("Items :");
       for (int i = 0; i < lcount; i++)
```

```
        {
          item3[i] = sc.next ();
        }
      System.out.println ("Quantity :");
      for (int i = 0; i < lcount; i++)
        {
          q3[i] = sc.nextInt ();
        }
      System.out.println ("Rate :");
      for (int i = 0; i < lcount; i++)
        {
          rate3[i] = sc.nextInt ();
        }
      i3.input (name3, item3, q3, rate3, item3.length, condition);
      i1.display ();
      i2.display ();
      i3.display ();
      break;

case 'D':
      System.out.println ("Enter the Invoice to be Deleted :");
      condition = sc.nextInt ();
      i1.input (name1, item1, qn1, rate1, item1.length, condition);
      i2.input (name2, item2, qn2, rate2, item2.length, condition);
      i1.display ();
      i2.display ();
      break;

case 'S':
```

```
System.out.println ("Enter the Invoice to be Searched :");
search = sc.nextInt ();
if (search == 0 || search > 2)
 {
  System.out.println ("The Searched Invoice didn't exist");
 }
else
 {
  switch (search)
   {
   case 1:
        System.out.println ("The Searched Invoice :");
        System.out.println ("**********************");
        i1.input (name1, item1, qn1, rate1, item1.length, condition);
        i1.display ();
        break;

   case 2:
        System.out.println ("The Searched Invoice :");
        System.out.println ("**********************");
        i2.input (name2, item2, qn2, rate2, item2.length, condition);
        i2.display ();
        break;

   default:
        System.out.println ("The Searched Invoice didn't exist");
   }
 }
break;
```

```
    default:
        break;
    }
  }
}
```

## Zoho Round 4

Technical interview

- OOPS concepts
- Multithreading
- Database concepts (SQL, CRUD operations)