COMPUTING APTITUDE

Student Name: Ayushi Aggarwal **UID:** 21MCA2806

Branch: MCA **Section/Group:**5A

Semester: II Sem **Date of Performance:**19 Feb’2022

Subject Name: Computing Aptitude **Subject Code:** 654

1. Aim/Overview of the practical:

Experiment 1.1.1

You are provided an array A of size N that contains non-negative integers. Your task is to determine whether the number that is formed by selecting the last digit of all the N numbers is divisible by 10.

Note: View the sample explanation section for more clarification.

Input format

- First line: A single integer N denoting the size of array A
- Second line: N space-separated integers.

Output format

If the number is divisible by 10, then print Yes. Otherwise, print No.

Constraints

1≤N≤1050≤A[i]≤105

Experiment 1.1.2

A bracket sequence is a string that contains only characters '(' and ')'. A correct bracket sequence is a bracket sequence that can be transformed into a correct arithmetic expression by inserting characters '1' and '+' between the original characters of the sequence. For example, bracket sequences '()()' and '(())' are correct. The resulting expressions of these sequences are: '(1)+(1)' and '((1+1)+1)'. However, '(', ')(', and '(' are incorrect bracket sequences.

You are given a bracket sequence s(s1s2……sn) , where denotes the type of 's bracket (open or close). It is not mandatory that s is necessarily correct. Your task is to determine the number of 's such that si si+1… sn s1 s2…si-1 is a correct bracket sequence.

Input format

K The single line contains sequence s.

Output format

Print the number of shifts denoting the correct bracket sequence.

Constraints

|s|<=5*105

Experiment 1.1.3

You are given 2 arrays A and B, each of the size N. Each element of these arrays is either a positive integer or -1. The total number of -1s that can appear over these 2 arrays are >=1 and <=2.

Now, you need to find the number of ways in which we can replace each -1 with a non-negative integer, such that the sum of both of these arrays is equal.

Input format

First line: An integer N

Second line: N space-separated integers, where the of these denotes A[i]

Third line: N space-separated integers, where the ith of these denotes B[i]

Output format

If there exists a finite number X, then print it. If the answer is not a finite integer, then print 'Infinite'.

Constraints

1<= N <= 105

-1<=A[i], b[i] <= 109

The -1’s may spread out among both arrays, and their quantity is between 1 and 2 (both inclusive)

2. Task to be done:

Experiment:1.1.1:

```
#include<stdio.h>
int main()
{
    int n,a[100],sum=0;
    printf("Enter Size of an Array:");
    scanf("%d",&n);
    printf("enter elements of an array:");
    for(int i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    for(int i=0;i<n;i++)
    {
        int rem=a[i]%10;
        sum=sum*10+rem;
    }
    printf("sum:%d\n",sum);
    if(sum%10==0)
    {
        printf("%d Divisible by 10",sum);
    }
    else
    {
         printf("%d Not Divisible by 10",sum);
    }
    return 0;
}
```

Experiment:1.1.2:

```
#include<stdio.h>
#include <string.h>
#include<stdbool.h>
```

```
char stack[100001];
int top = -1;
void push(char x)
{
    top++;
   /* if(top == n)
    {
        //printf("Stack overflows");
        return;
    }*/
    stack[top] = x;
}
void pop()
{
    if(top == -1)
    {
        //printf("No element to pop..");
        return;
    }
    top--;
}
bool isEmpty()
{
    if(top == -1)
        return true;
    else
        return false;
}
bool isBalanced(char s[100001], int n, int i)
{
    for(int j=i;j<i+n;j++)
    {
        if(s[j%n]=='(')
            push('(');
        else
        if(s[j%n]==')' && isEmpty()== false)
            pop();  // for ) bracket
        else
        {
            return false;
        }
    }
    if(isEmpty() == true) return true;
    return false;
```

```
}
void main()
{
    char s[100001];
    gets(s);
    int n = strlen(s);
    int ans=0;
    for(int i=0;i<n;i++)
    {
      int c=0;
      if(s[i]=='(')// run only when first bracket = '('
     {
        if(isBalanced(s,n,i) == true) // if correct seq then only increasing ans
        {
           //printf("Hello ");
           ans++;
        }
      }
   }
         printf("Ans : %d",ans);
}
```

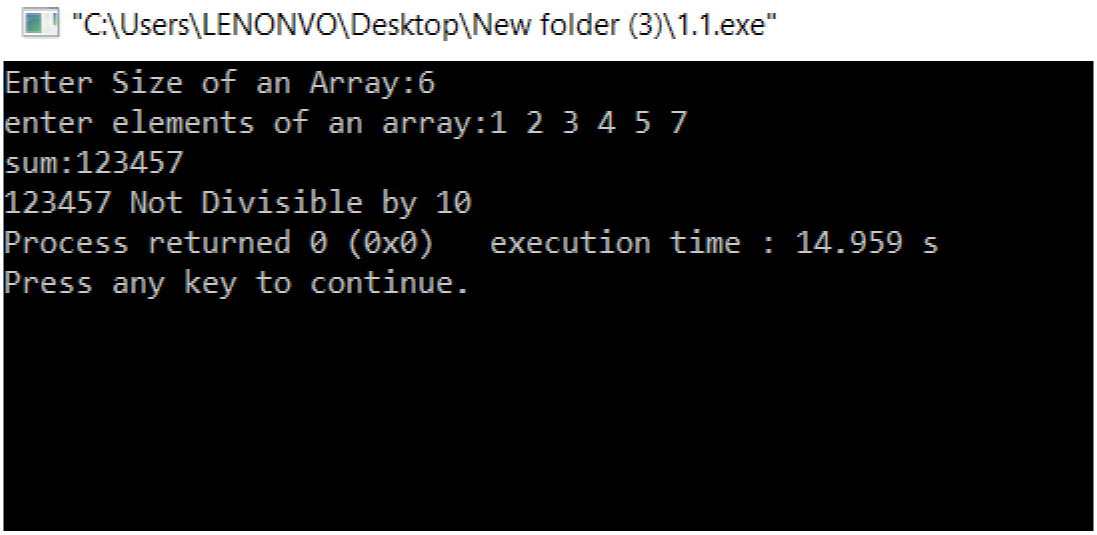
Experiment:1.1.3:

```
#include <stdio.h>
int mod(int x)
{
   if(x<0) return -1*x;
   return x;
}
int main()
{
         int n;
         printf("Enter the size of an array:");
         scanf("%d", &n);
         int a[n],b[n];
         int freq1=0, freq2=0;
         int sum1=0, sum2=0;
         printf("enter the element of 1st array:");
         for(int i=0;i<n;i++)
   {
      scanf("%d",&a[i]);
   }
   printf("enter the element of 2nd array:");
   for(int i=0;i<n;i++)
   {
        scanf("%d",&b[i]);
    }
    for(int i=0;i<n;i++)
    {
        if(a[i] == -1)
            freq1++;
        else
            sum1+=a[i];

        if(b[i] == -1)
            freq2++;
        else
            sum2+=b[i];
    }
    if(freq1>0 && freq2>0)
    {
        printf("Infinite");
    }
    else
    if( (freq1==1 && freq2==0 && sum1<sum2) || (freq1==0 && freq2==1 && sum1>sum2))
    {
        printf("1");
    }
    else
    if((freq1>=1 && freq2==0 && sum1>sum2) || (freq1==0 && freq2>=1 && sum1<sum2))
    {
        printf("0");
    }
    else
    {
        printf("%d",mod(sum2-sum1) +1);
    }
    return 0;
}
```

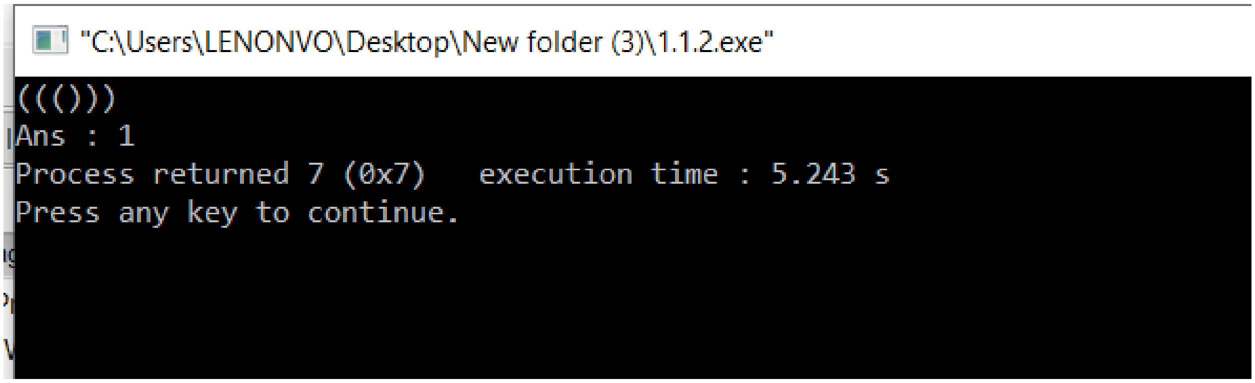
3. Result/Output/Writing Summary:

Experiment:1.1.1:



```
"C:\Users\LENONVO\Desktop\New folder (3)\1.1.exe"
Enter Size of an Array:6
enter elements of an array:1 2 3 4 5 7
sum:123457
123457 Not Divisible by 10
Process returned 0 (0x0)   execution time : 14.959 s
Press any key to continue.
```

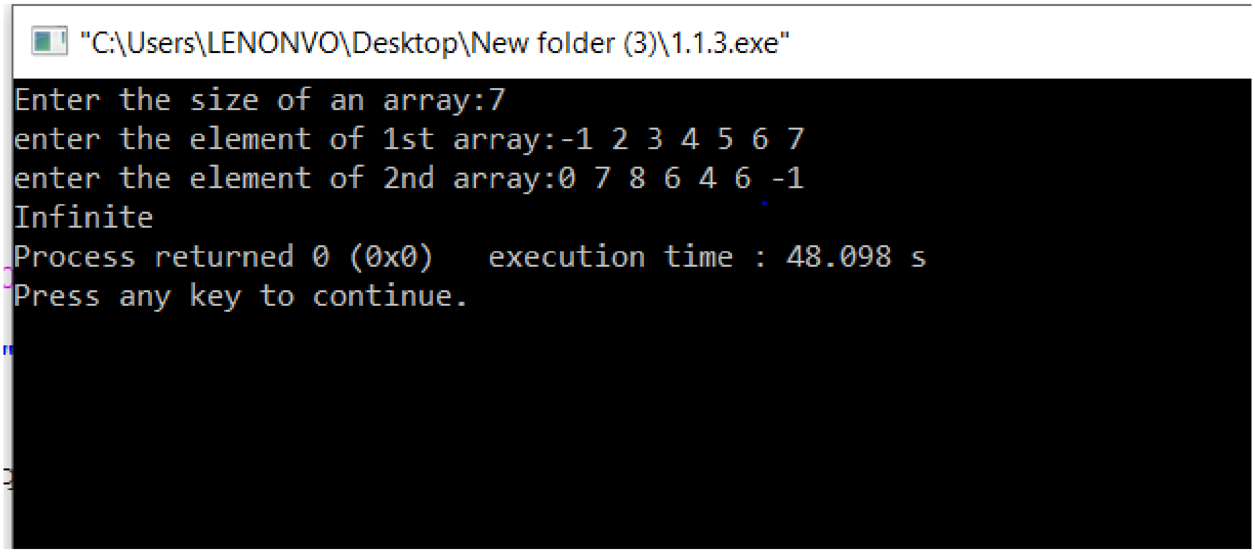
Experiment:1.1.2:



```
"C:\Users\LENONVO\Desktop\New folder (3)\1.1.2.exe"
((()))
Ans : 1
Process returned 7 (0x7)   execution time : 5.243 s
Press any key to continue.
```

Experiment:1.1.3:

1.



"C:\Users\LENONVO\Desktop\New folder (3)\1.1.3.exe"

```
Enter the size of an array:7
enter the element of 1st array:-1 2 3 4 5 6 7
enter the element of 2nd array:0 7 8 6 4 6 -1
Infinite
Process returned 0 (0x0)   execution time : 48.098 s
Press any key to continue.
```

Learning outcomes (What I have learnt):

1. **Learnt about array and conditional statement.**

2. **Learnt about stack**

3. **Learnt about string.**

Evaluation Grid:

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Worksheet		10
2.	Demonstration/Performance /Pre Lab Quiz		5
3.	Post Lab Quiz		5

1.