Assignments 1 (CSE1002)

Bimal Parajuli(20BDS0405)

15th June 2021

Assignment 1-1

Problem

If Give an integer N . Write a program to obtain the sum of the first and last digits of this number.

Input

The first line contains an integer T, the total number of test cases. Then follow T lines, each line contains an integer N.

Output

For each test case, display the sum of first and last digits of N in a new line.

Constraints

- 1 ≤ T ≤ 1000
- 1 ≤ N ≤ 1000000

Input

```
3
1234
124894
242323
```

output

5

5

5

12:23

Your code has Passed Execution!

```c
#include <stdio.h>

int summ(int x)
{
    int first, last;
    last = x % 10;
    while (x > 0)
    {
        first = x%10;
        x=x/10;
    }

    return (first + last);
}


int main()
{
    int n;
    int arr[n];
    scanf("%d",&n);
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &arr[i]);
    }


    for (int i = 0; i < n; i++)
    {
        printf("%d\n",summ(arr[i]));
    }
    return 0;
}
```

12:23

Assignment 1

Pooja would like to withdraw **x** $US from an ATM. The cash machine will only accept the transaction if **x** is a multiple of 5, and Pooja's account balance has enough cash to perform the withdrawal transaction (including bank charges). For each successful withdrawal the bank charges 0.50 $US. Calculate Pooja's account balance after an attempted transaction.

Input

Positive integer 0 < **x** <= 2000 - the amount of cash which Pooja wishes to withdraw.

Nonnegative number 0<= Y <= 2000 with two digits of precision - Pooja's initial account balance.

Output

Output the account balance after the attempted transaction, given as a number with two digits of precision. If there is not enough money in the account to complete the transaction, output the current bank balance.

Example - Successful Transaction

```
Input:
30 120.00

Output:
89.50
```

Example - Incorrect Withdrawal Amount (not multiple of 5)

```
Input:
42 120.00

Output:
120.00
```

Example - Insufficient Funds

```
Input:
300 120.00

Output:
120.00
```

12:24

Font Size 18 | Language | Editor Theme: Select a Theme

Your code has passed Execution

```c
#include <stdio.h>

int main()
{
    int withdraw;
    float balance;
    scanf("%d %f", &withdraw, &balance);
    if (withdraw > 0 && withdraw <= 2000 && balance > 0 && balance <= 2000)
    {
        if (withdraw % 5 == 0 && (withdraw + 0.5) <= balance)
        {
            printf("%.2f", balance - withdraw - 0.5);
        }
        else
        {
            printf("%.2f", balance);
        }
    }
    return 0;
    // @the_calc_eject_48
}
```

Save | Pause Test

Submit Code

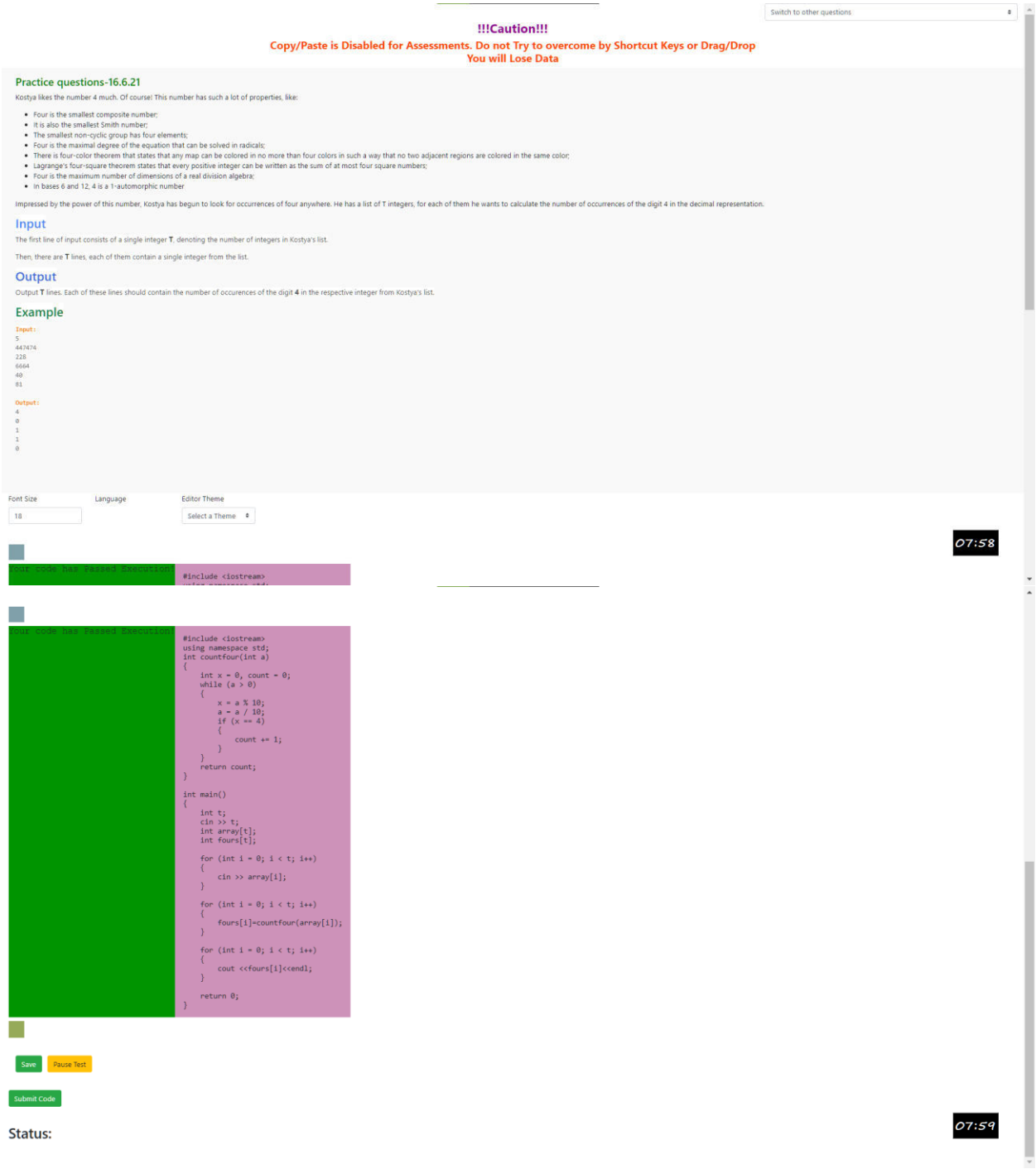
12:25

Status:

BIMAL PARAJULI (20BDS0405)

Practice problem 16th June, 2021

Problem 1: Counting 4s in a number:



Switch to other questions

!!!Caution!!!

Copy/Paste is Disabled for Assessments. Do not Try to overcome by Shortcut Keys or Drag/Drop
You will Lose Data

Practice questions-16.6.21

Kostya likes the number 4 much. Of course! This number has such a lot of properties, like:

- Four is the smallest composite number;
- It is also the smallest Smith number;
- The smallest non-cyclic group has four elements;
- Four is the maximal degree of the equation that can be solved in radicals;
- There is four-color theorem that states that any map can be colored in no more than four colors in such a way that no two adjacent regions are colored in the same color;
- Lagrange's four-square theorem states that every positive integer can be written as the sum of at most four square numbers;
- Four is the maximum number of dimensions of a real division algebra;
- In bases 6 and 12, 4 is a 1-automorphic number.

Impressed by the power of this number, Kostya has begun to look for occurrences of four anywhere. He has a list of T integers, for each of them he wants to calculate the number of occurrences of the digit 4 in the decimal representation.

Input

The first line of input consists of a single integer **T**, denoting the number of integers in Kostya's list.

Then, there are **T** lines, each of them contain a single integer from the list.

Output

Output **T** lines. Each of these lines should contain the number of occurences of the digit **4** in the respective integer from Kostya's list.

Example

```
Input:
5
447474
228
6664
40
81

Output:
4
0
1
1
0
```

Font Size: 18 | Language | Editor Theme: Select a Theme

07:58

Your code has Passed Execution

```
#include <iostream>
using namespace std;
int countfour(int a)
{
    int x = 0, count = 0;
    while (a > 0)
    {
        x = a % 10;
        a = a / 10;
        if (x == 4)
        {
            count += 1;
        }
    }
    return count;
}

int main()
{
    int t;
    cin >> t;
    int array[t];
    int fours[t];

    for (int i = 0; i < t; i++)
    {
        cin >> array[i];
    }

    for (int i = 0; i < t; i++)
    {
        fours[i]=countfour(array[i]);
    }

    for (int i = 0; i < t; i++)
    {
        cout <<fours[i]<<endl;
    }

    return 0;
}
```

Save | Pause Test

Submit Code

Status:

07:59

Problem 2: Merge Sort Problem:

Problem 3: Queue operations:

Practice question2-16.6.21

Write a C++ Menu Driven Program for Queue Operations using Arrays. Here's a Simple Program for Queue Operations using Arrays in C++ Programming Language.

OUTPUT : :

```
// **********QUEUE**********

1.Insertion
2.Deletion
3.Display

Enter your choice:1

Enter the number to be inserted: 1

Number pushed in the queue:1
1.Insertion
2.Deletion
3.Display

Enter your choice:1

Enter the number to be inserted: 2

Number pushed in the queue:2
1.Insertion
2.Deletion
3.Display

Enter your choice:1

Enter the number to be inserted: 3

Number pushed in the queue:3
1.Insertion
2.Deletion
3.Display

Enter your choice:3

Contents of queue is:1  2       3
1.Insertion
2.Deletion
3.Display

Enter your choice:2

Deleted element is:1
1.Insertion
2.Deletion
3.Display

Enter your choice:3

Contents of queue is:2  3
1.Insertion
2.Deletion
3.Display

Enter your choice:4

Enter your choice:4

Invalid choice!!

Exit code: 0
```

08:05

Font Size: 18 | Language | Editor Theme: Select a Theme

Your code has passed execution

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

#define SIZE 100
int arr[100];
int front = -1;
int rear = -1;

bool isempty() //returns true if value at both front and rear pointers are -1.
{
    if (front == -1 && rear == -1)
    {
        return true;
    }
    else
    {
        return false;
    }
}

void enqueue(int value)  //adds an element to the rear of the existing queue
{
    if (rear == SIZE - 1) //check if the queue is full
    {
        cout << "Queue is full\n";
    }
    else
    {
        if (front == -1) //inserting the first element
        {
            front = 0;
        }
        rear++;
        arr[rear] = value; //inserting a value at rear
        cout<<"Number pushed in the queue: "<<value;
    }
}

void dequeue()
{
    if (isempty())
    {
        cout<<"The Queue is empty\n";
    }
    else if (front==rear)
```

08:05

```
    }
    else if (front==rear)
    {
        front=rear=-1;
    }
    else
    {
        cout<<"Deleted element is : "<<arr[front];
        front++;
    }
}

void showfront()
{
    if(isempty())
    {
        cout<<"Queue is empty\n";
    }
    else
    {
        cout<<"front element:"<<arr[front];
    }
}

void displayqueue()
{
    if (isempty())
    {
        cout<<"Empty Queue";
    }
    else
    {
        for(int i=front; i<=rear; i++)
        {
            cout<<arr[i]<<" ";
        }
    }

}

int main()
{
    int value, operation;
    bool check=true;
    cout<<"Enter your choice: ";
    cin>>operation;
    while(check)
    {
        if(operation==1)
        {
            cin>>value;
            enqueue(value);
        }
        else if (operation==2)
        {
            dequeue();
        }
        else if (operation==3)
        {
            cout<<"The contents of the queue is: ";
            displayqueue();
        }
        else
        {
            cout<<"Invalid Choice";
            check=false;
        }

    }
    return 0;
}
```

Save Pause Test

Submit Code

Status:

08:07

08:08