The Bug

Question:

1: 2 of Java

2: 3 of C

JAVA-Armstrong Number

Write a program to check whether a given number is Armstrong or not.

You have to give a input and it will check whether it is Armstrong or not.

Input: 153

Output: Yes, It is Armstrong number.

Input: 123

Output: No, It is not Armstrong number.

Example:

Input: 153

No. of Digit= 3

Sum= $1^3 + 5^3 + 3^3$

= 153

Input: 9

No. of Digit= 1

Sum= 9^1

= 9

Input: 1634

No. of Digit= 4

Sum= $1^4 + 6^4 + 3^4 + 4^4$

=1634

These all are Armstrong Number.

The Bug code:

```
  import java.util.*;
class armstrong_number1
{
    void main()
    {
        float sum=0;          //Sum used to store Armstrong number
        int count=0;          // Integer used to count number of digit.
        Scanner sc = new Scanner(System.in);
        int num= sc.nextint();   //Integer to store Input from user
        int c = num;            // Integer to copy num value.
        while(num>=0)            //While loop used for counting digit
        {
            num=num/10;
            count++;
        }
        num=c;
        while(num>0)             //While used to store value of Armstrong in sum.
        {
            int rem=num/10;
            sum=sum + math.pow(rem,count);
            num=num/10;
        }
        if(sum==c)          //If used to check that given number is Armstrong or not
        System.out.println("Yes, It is Armstrong Number");
        else
        System.out.println("No,It is not Armstrong Number");
    }
}
```

Correct Code:

```
import java.util.*;
class armstrong_number
{
    void main()
    {
        double sum=0;         //Double used to store Armstrong number
        int count=0;          // Integer used to count number of digit.
        Scanner sc = new Scanner(System.in);
        int num= sc.nextInt();   //Integer to store Input from user
        int c = num;             // Integer to copy num value.
        while(num>0)             //While loop used for counting digit.
        {
            num=num/10;
            count++;
        }
        num=c;
        while(num>0)    //While used to store value of Armstrong in sum
        {
            int rem=num%10;
            sum=sum + Math.pow(rem,count);
            num=num/10;
        }
        if(sum==c)           //If used to check that given number is Armstrong or not
        System.out.println("Yes, It is Armstrong Number");
        else
        System.out.println("No,It is not Armstrong Number");
    }
}
```

JAVA-Disarium Number

Write a program to check whether a given number is Disarium or not.

You have to give a input and it will check whether it is Disarium or not.

Input: 175

Output: Yes, It is Disarium number.

Input: 123

Output: No, It is not Disarium number.

Example:

Input: 175

No. of Digit= 3

Sum= $1^1 + 7^2 + 5^3$

= 175

Input: 9

No. of Digit= 1

Sum= 9^1

= 9

Input: 135

No. of Digit= 3

Sum= $1^1 + 3^2 + 5^3$

=135

These all are Disarium Number

The Bug code:

```
import java.util.*;

class disarium2

{

   void main()

   {

      int  sum=0;        //Double used to store Disarium Number

      int count=0;         // Integer used to count number of digit.

      Scanner Sc = new Scanner(System.in);

      int num= sc.nextint();   //Integer to store Input from user

      int c = num;            // Integer to copy num value.

      while(num<0)    //While loop used for counting number of digit

      {

         num=num/10;

         count++;

      }

      num=c;

      while(num>0)   //While used to store value of Disarium in sum.

      {

         int rem=num%10;

         sum=sum + math.pow(rem,c);

         num=num%10;

         count++;

      }

     if(sum==c)                  //If used to check that given number is Disarium or not

      System.out.println("Yes, It is Disarium Number");

      else

      System.out.println("No,It is not Disarium Number");

   }

}
```

Correct Code:

```
import java.util.*;

class disarium

{

   void main()

   {

       double sum=0;        //Double used to store Disarium Number

      int count=0;         // Integer used to count number of digit.

      Scanner sc = new Scanner(System.in);

      int num= sc.nextInt();   //Integer to store Input from user

      int c = num;            // Integer to copy num value.

      while(num>0)            //While loop used for counting number of digit

      {

         num=num/10;

         count++;

      }

      num=c;

      while(num>0)             //While used to store value of Disarium in sum.

      {

         int rem=num%10;

         sum=sum + Math.pow(rem,count);

         num=num/10;

         count--;

      }

      if(sum==c)                 //If used to check that given number is Disarium or not

      System.out.println("Yes, It is Disarium Number");

      else

      System.out.println("No,It is not Disarium Number");

   }

}
```

C-Rotate Array

Dhiraj and Akash are friend they both have started coding few weeks ago. Dhiraj made a code which rotate the whole array which user input after that he shows his code to Akash. Seeing the code Akash challenge Dhiraj to make a code to rotate array which can be rotated by from specific position. Dhiraj accept the challenge and start to make code but he finds sum Bugs in his code which he cannot resolve please help him to resolve the bugs of the code so he can win the challenge.

Sample Input 1:

7

1 2 3 4 5 6 7

3

Sample Output 1:

4 5 6 7 1 2 3

Sample Input 2:

9

1 5 3 43 7 3 12 4 24

5

Sample Output 2:

3 13 4 24 1 5 3 43 7

Explanation:

Input 1:

7 = Number of value array can store (denoted in program by **n**)

1 2 3 4 5 6 7 = Value store in array (denote in program by **arr[]**)

3 =Position form where array has to be rotated (denote in program by **d**)

Input 2:

9 = Number of value array can store (denoted in program by **n**)

3 13 4 24 1 5 3 43 7 = Value store in array (denote in program by **arr[]**)

5 =Position form where array has to be rotated (denote in program by **d**)

Output 1:

```
4 5 6 7 1 2 3
```

As **d** is 3 array rotated by 3 position and (1 2 3) move back

Output 2:

```
3 13 4 24 1 5 3 43 7
```

As **d** is 3 array rotated by 5 position and (1 5 3 43 7) move back

The Bug Code:

```
//C program is Rotate rotate element by d element

#include <stdio.h>

//Function to left Rotate arr[] of size n by 1

void leftRotate(int arr, int n);

//Function to left rotate arr[] of size n by d

void Rotate(int arr,int d,int n);

//Function to left rotate arr[] of size n by d

void printArray(int arr,int n);


 int main() //Used for taking input by user

{

   int n;

   scanf("%d",&n);   //Input size of array

   int arr[n];

   for(int i=0;i<n;i++)       //Loop used for taking input of array

   {

      scanf("%d",arr[i]);

   }

   int d;

   scanf("%d",&d);        //d used to rotate array from the position

   Rotate(arr, n, d);      //Calling Rotate function

   PrintArray(arr, d);     //Calling PrintArray function to print rotated function

   return 0;

}

void Rotate(int arr[], int d, int n)
```

```
{
    for (int i = 0; i < d; i++)
      leftRotate(arr, n);  //Calling leftRotate to rotate array value by one position
}

void leftRotate(int arr[], int n)
{
    int temp = arr[0], i;
    for (i = 0; i < n; i++)   // loop used to rotate array
        arr[i] = arr[i + 1];
    arr[n] = temp;
}

void printArray(int arr[], int n)
{
    int i;
    for (i = 1; i < n; i++)   //loop used to print rotated array
        printf("%d ", arr[i]);
}
```

Correct Code:

```
//C program is Rotate rotate element by d element
#include <stdio.h>
//Function to left Rotate arr[] of size n by 1
void leftRotate(int arr, int n);
//Function to left rotate arr[] of size n by d
void Rotate(int arr,int d,int n);
//Function to left rotate arr[] of size n by d
void printArray(int arr,int n);

 int main() //Used for taking input by user
{
    int n;
    scanf("%d",&n);   //Input size of array
    int arr[n];
    for(int i=0;i<n;i++)       //Loop used for taking input of array
    {
        scanf("%d",arr[i]);
    }
    int d;
    scanf("%d",&d);        //d used to rotate array from the position
    Rotate(arr, n, d);      //Callling Rotate function
    PrintArray(arr, d);     //Calling PrintArray function to print rotated function
    return 0;
}
void Rotate(int arr[], int d, int n)
```

```
{
   for (int i = 0; i < d; i++)
     leftRotate(arr, n);   //Calling leftRotate to rotate array value by one position
}

void leftRotate(int arr[], int n)
{
   int temp = arr[0], i;
   for (i = 0; i < n; i++)   // loop used to rotate array
      arr[i] = arr[i + 1];
   arr[n] = temp;
}

void printArray(int arr[], int n)
{
   int i;
   for (i = 1; i < n; i++)   //loop used to print rotated array
      printf("%d ", arr[i]);
}
```

C-Lucky Number

Write a Program in c to check whether number is Lucky Number or Not.

User has to give a input and check whether it is lucky number or not.

Method to find Lucky Number.

Let take a number as input

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Know Remove all the Number at 2nd Position

1 **2** 3 **4** 5 **6** 7 **8** 9 **10** 11 **12** 13 **14** 15 **16** 17 **18** 19 **20**

Left Number

1 3 5 7 9 11 13 15 17 19

Know Remove all the Number at 3rd Position

1 3 **5** 7 9 **11** 13 **15** 17 19

Left Number

1 3 7 9 13 17 19

Know Remove all the Number at 4th Position

1 3 7 **9** 13 17 19

Left Number

1 3 7 13 17 19

Know Remove all the Number at 5th Position

1 3 7 13 **17** 19

Left Number

1 3 7 13 19

Know Remove all the Number at 5th Position

1 3 7 13 19

It is not possible because it does not have 6th Position

So, Lucky Number

1   3   7   13   19

Sample Input 1: 7

Sample Output 1: Yes, It's Lucky Number

Sample Input 2: 14

Sample Output 2: No, It's not Lucky Number

Sample Input 3: 19

Sample Output 3: Yes, It's Lucky Number

Explanation:

Sample Input 1: 7

Sample Input 1: 14

Sample Input 1: 19

Input any positive number.

Sample Output 1: Yes, It's Lucky Number

Sample Output 2: No, It's not Lucky Number

Sample Output 3: Yes, It's Lucky Number

First find lucky number using above method if it lies in list then it is lucky

Number otherwise it is lucky number.

The Bug Code:

```
//The Program is to Check given number is Lucky number or Not.
#include <stdio.h>

int main()
{
   int n;        //use to Take input number
   int c=0;l=3;   //use to Take input number
   scanf("%d",&n);   // Taking input
   int num[n];     //initialising variable Space
   int num1[n];    //initialising variable Space
   int m=n;
   for(int i=0;i<n;i++)    //loop to take input in array num[]
   {
      num[i]=i;
   }
while(m>l)    //while used to check position should not go above number of
                        variable
   {
      for(int i=0;i<m;i++)   //for loop used to delete l  position variable
      {
         if((i-1)%l!=0)
         {
            num1[c]=num[i];
            c++;
```

```
          }
       }
        m=c;
        c=0;
        for(int i=0;i<m;i++)   //for loop used initialize num[] by new value using num1[]
        {
         num[i]=num1[i];
        }
        l++;
}
for(int i=0;i<m;i++)   //for loop checking if desired number is variable or not
{
   if(num[i]==n)
      c=1;
      else
         c=0;
}
if(c==1)               //Checking given number is lucky number or not
printf("Yes, It's Lucky Number");
else
printf("NO, It's not Lucky Number");
}
```

C- Sum prime Combi

Dhiraj is student who is very good in maths so his maths teacher gives him assignment to express the number in all possible combination of sum of Two Prime Number. Dhiraj is very smart to decrease his load of work he wrote a program that can find all such all combination.

Example:

Input: 34

Output:

```
34 = 3 + 31
34 = 5 + 29
34 = 11 + 23
34 = 17 + 17
```

Input: 42

Output:

```
42 = 5 + 37
42 = 11 + 31
42 = 13 + 29
42 = 19 + 23
```

Input: 41

Output:

```
No Possible Combination
```

Explanation:

Input:

Input any Natural Number.

Output:

In prime combination available print combination otherwise print a message

“No Possible Combination”.

The Bug Code:

```
#include <stdio.h>
// Function used to check Prime number
int Prime(int n);
int main()
{
  int num, count = 0;
  scanf("%d",&num);   //taking input number
  for(int i=1;i<num/2;i++)    //for loop used to check all sum combination
  {
 if(Prime(i) == 1)      //condition to check whether first number of combination
                      is prime or not by calling function
    {
 if(Prime(num)==0)       // condition to check whether second number of
                             combination is prime or not by calling function
       {
printf("%d = %d + %d\n",num,i,num-1); // printing all possible combination if
                                         possible
         count =1;
       }
  }
}
       if(count==0)
printf("No Possible Such Combination"); // printing message if no such possible
                                            combination
return 0;
}

int Prime(int n)

{

int CPrime;

for(int i=1;i<=n;i++)    // for loop to check prime number

{

  if(n%i==0)             // checking condition of prime

    CPrime++;

}

 return 1;        // returning value of Cprime

}
```