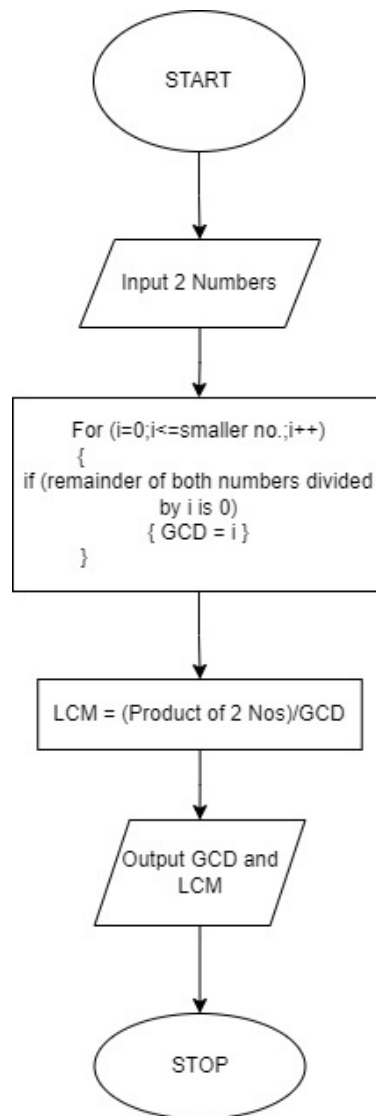


Name	Vineet Parmar
UID no.	2021300092
Experiment No.	2

AIM:	Apply various control structures to solve given problems.
Program 1	
PROBLEM STATEMENT :	Apply various control structures to solve given problems.
ALGORITHM:	<ol style="list-style-type: none"> 1. START 2. Input 2 Numbers 3. For (I=1;I<=Smaller No.;I++) <ul style="list-style-type: none"> { if(Remainder of both Numbers divided by I is 0) { GCD = I } } 4. LCM = (Product of 2 Nos.) / GCD 5. Print LCM and GCD 6. STOP

FLOWCHART:**PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int a,b;
    printf("Enter two numbers, the smaller one first: ");
    scanf("%d %d",&a,&b);
    int lcm,gcd;
    for(int i=1;i<=a;i++)
    {
        if(a%i == 0 && b%i == 0)
        {
```

	<pre> gcd = i; } } lcm=(a*b)/gcd; printf("The LCM and GCD of %d and %d are %d and %d respectively.",a,b,lcm,gcd); return 0 ; } </pre>
--	---

```

Enter two numbers, the smaller one first: 10
25
The LCM and GCD of 10 and 25 are 50 and 5 respectively.

...Program finished with exit code 0
Press ENTER to exit console.

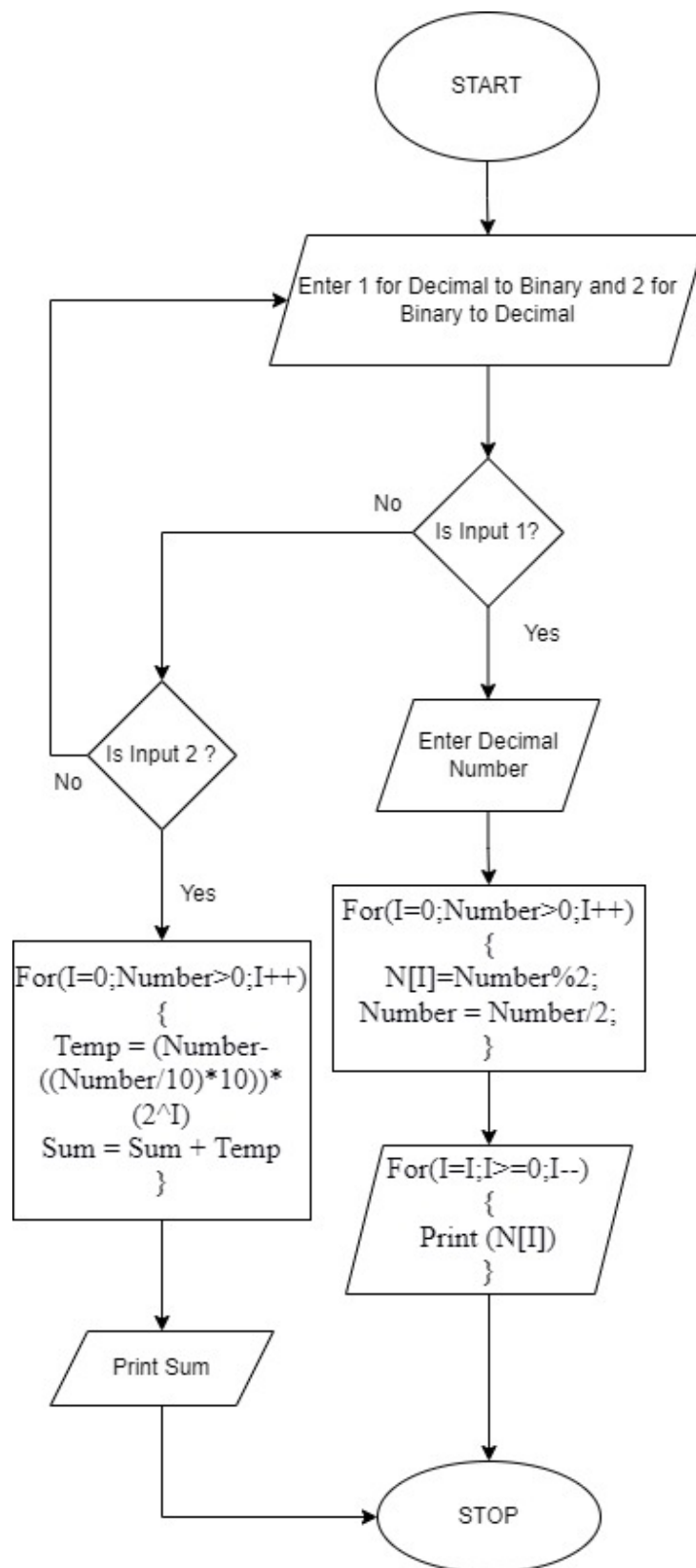
```

RESULT:

Program 2

PROBLEM STATEMENT :	Write a program to convert a decimal number to binary or convert a binary number to decimal
ALGORITHM:	<ol style="list-style-type: none"> 1. START 2. Input 1 for Decimal to Binary and 2 for Binary to Decimal 3. If Input =1 Input Decimal Number For(I=0;Number>0;I++) { N[I]=Number%2; Number = Number/2; } For(I=I;I>=0;I--) { Print (N[I]) } 4. Else If Input = 2 Input Binary Number Sum = 0 For(I=0;Number>0;I++) { Temp = (Number-((Number/10)*10))*(2^I) Sum = Sum + Temp } Print Sum 5. STOP

FLOWCHART:



PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
int main()
{
    int flag=0;
    int input;
    do
    {
        printf("If you want to convert Decimal to Binary, type 1 and if you
want to convert Binary to Decimal, then type 2\n");
        scanf("%d",&input);
        if(input==1)
        {
            flag=1;
            int dec,i,n,a[100];
            printf("\nEnter the Decimal Number: ");
            scanf("%d",&dec);
            n = dec;
            for(i=0;dec>0;i++)
            {
                a[i]=dec%2;
                dec=dec/2;
            }
            printf("\nBinary of %d is ",n);
            for(i=i-1;i>=0;i--)
            {
                printf("%d",a[i]);
            }
        }
        else if(input==2)
        {
            flag=1;
            int bin,n,temp,sum=0;
            printf("\nEnter the Binary Number: ");
            scanf("%d",&bin);
            n=bin;
            for(int i=0;n>0;i++)
            {
                temp=(n-((n/10)*10))*pow(2,i);
```

	<pre> n=n/10; sum = sum + temp; } printf("\nDecimal of %d is %d",bin,sum); } else { flag=0; printf("\nWrong Choice. Enter Again\n"); } }while(flag==0); return 0 ; } </pre>
--	---

RESULT:

```

If you want to convert Decimal to Binary, type 1 and if you want to convert Binary to Decimal, then type 2
1
Enter the Decimal Number: 299
Binary of 299 is 100101011
...Program finished with exit code 0
Press ENTER to exit console.

```

```

If you want to convert Decimal to Binary, type 1 and if you want to convert Binary to Decimal, then type 2
3
Wrong Choice. Enter Again
If you want to convert Decimal to Binary, type 1 and if you want to convert Binary to Decimal, then type 2
2
Enter the Binary Number: 1111
Decimal of 1111 is 15
...Program finished with exit code 0
Press ENTER to exit console.

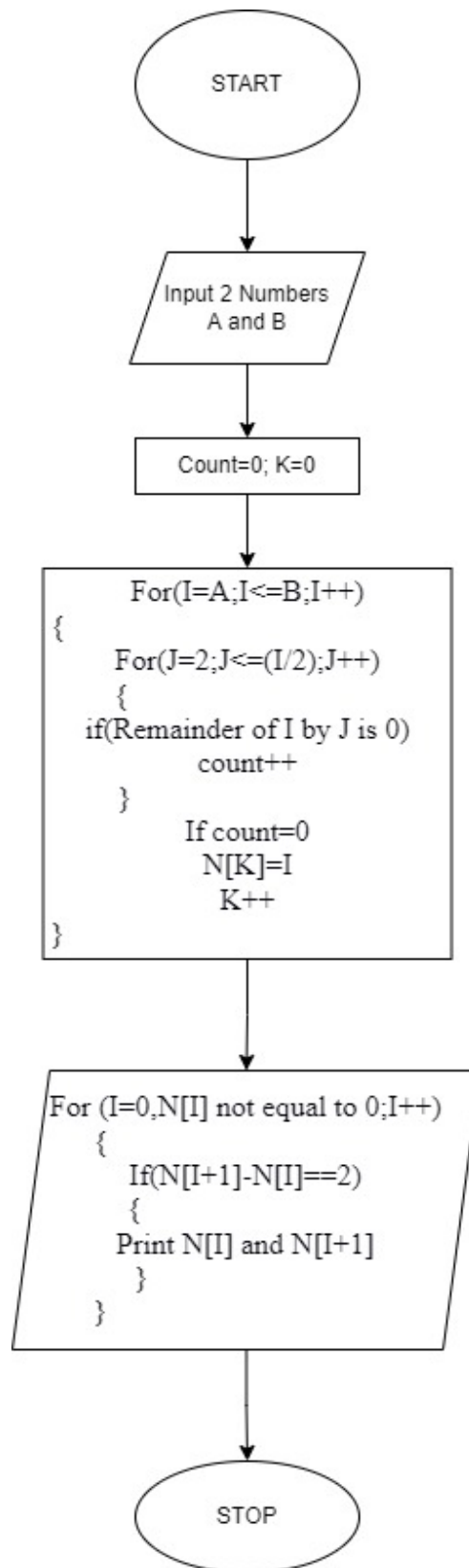
```

Program 3

PROBLEM STATEMENT:	Twin primes are consecutive odd numbers, both of which are prime numbers. Write a program which inputs two positive integers A and B and outputs all twin primes in range A to B.
ALGORITHM:	<ol style="list-style-type: none"> 1. START 2. Input 2 Numbers A and B 3. Count = 0, K=0 4. For(I=A;I<=B;I++) <pre> { For(J=2;J<=(I/2);J++) { </pre>

	<pre> if(Remainder of I by J is 0) count++ } If count=0 N[K]=I K++ } 5. For (I=0,N[I] not equal to 0;I++) { If(N[I+1]-N[I]==2) { Print N[I] and N[I+1] } } 6. STOP</pre>
--	---

FLOWCHART:



PROGRAM:

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int a,b;
    printf("Enter two positive numbers: ");
    scanf("%d %d",&a,&b);
    int n[100],k=0,count=0,i,j;
    for(i=a;i<=b;i++)//calling all numbers between A to B
    {
        count=0;
        for(j=2;j<i;j++)//for loop for checking if a number is prime or not
        {
            if(i%j==0)
            {
                count++;
            }
        }
        if(count==0)//storing all prime numbers inside an array
        {
            n[k]=i;
            k++;
        }
    }
    for(int i=0;i<=k;i++)
    {
        if(n[i+1]-n[i]==2)
        {
            printf("\n%d %d\n",n[i],n[i+1]);
        }
    }
    return 0;
}
```

```

Enter two positive numbers:
3
99
3 5
5 7
11 13
17 19
29 31
41 43
59 61
71 73

...Program finished with exit code 0
Press ENTER to exit console.

```

RESULT:

Program 4

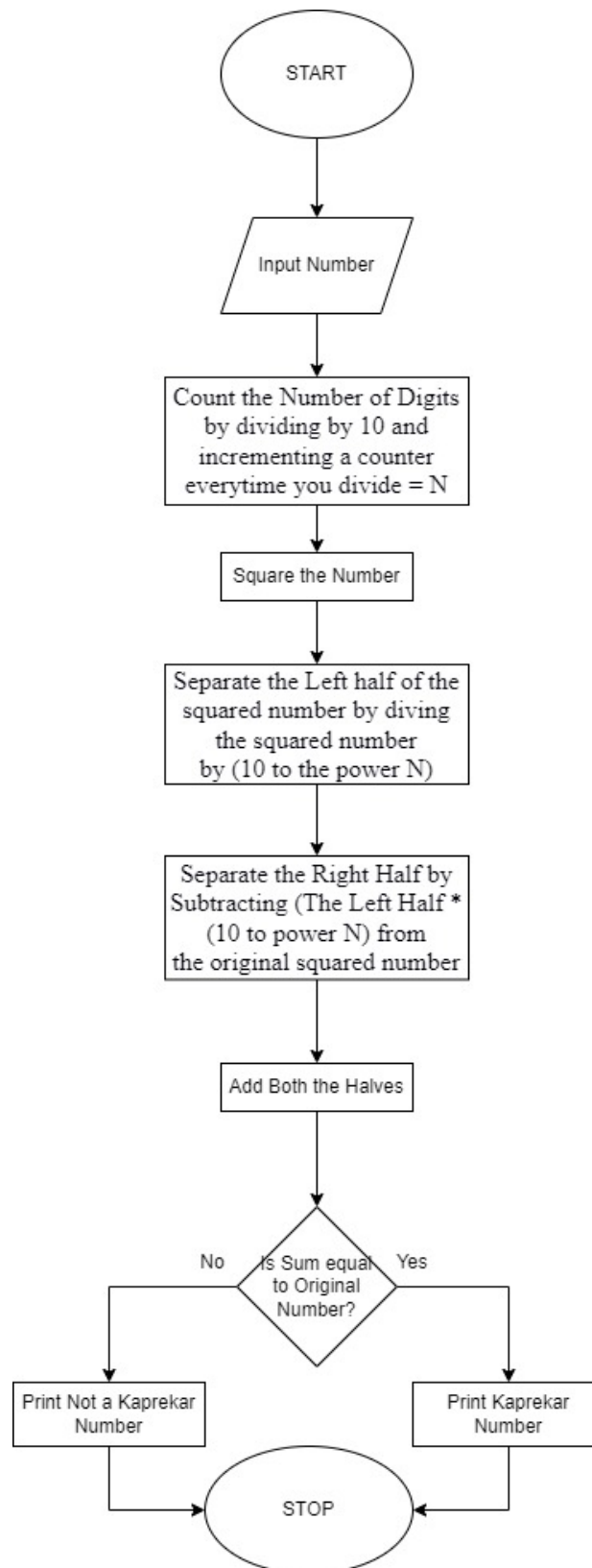
PROBLEM STATEMENT:

Write a program to find out whether a number is kaprekar or not. Consider an n-digit number k. Square it and add the right n digits to the left n or n-1 digits. If the resultant sum is k, then k is called a Kaprekar number. For example, 9 is a Kaprekar number.

ALGORITHM:

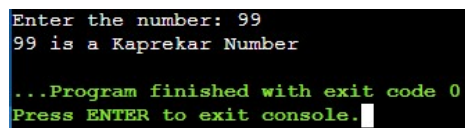
1. START
2. Input the Number
3. Count the Number of Digits by dividing by 10 and incrementing a counter everytime you divide = N
4. Square the Number
5. Separate the Left half of the squared number by dividing the squared number by (10 to the power N)
6. Separate the Right Half by Subtracting (The Left Half * (10 to power N) from the original squared number
7. Add both the Halves
8. If Sum= Original Number
Print Kaprekar Number
Else
Print Not a Kaprekar Number
9. STOP

FLOWCHART:

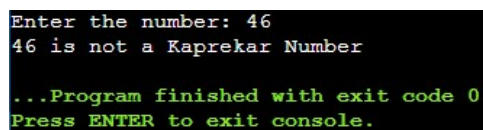


PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
int main()
{
    int a,n=0,temp;
    printf("Enter the number: ");
    scanf("%d",&a);
    temp=a;
    for(int i=0;temp>0;i++)
    {
        temp=temp/10;
        n++;
    }
    int b = a*a;
    int b1,b2,p;
    p=pow(10,n);
    b1=b/p;
    b2=b%p;
    if((b1+b2)==a)
    {
        printf("%d is a Kaprekar Number",a);
    }
    else
    {
        printf("%d is not a Kaprekar Number",a);
    }
    return 0;
}
```

RESULT:

```
Enter the number: 99
99 is a Kaprekar Number
...Program finished with exit code 0
Press ENTER to exit console.
```



```
Enter the number: 46
46 is not a Kaprekar Number
...Program finished with exit code 0
Press ENTER to exit console.
```

Program 5

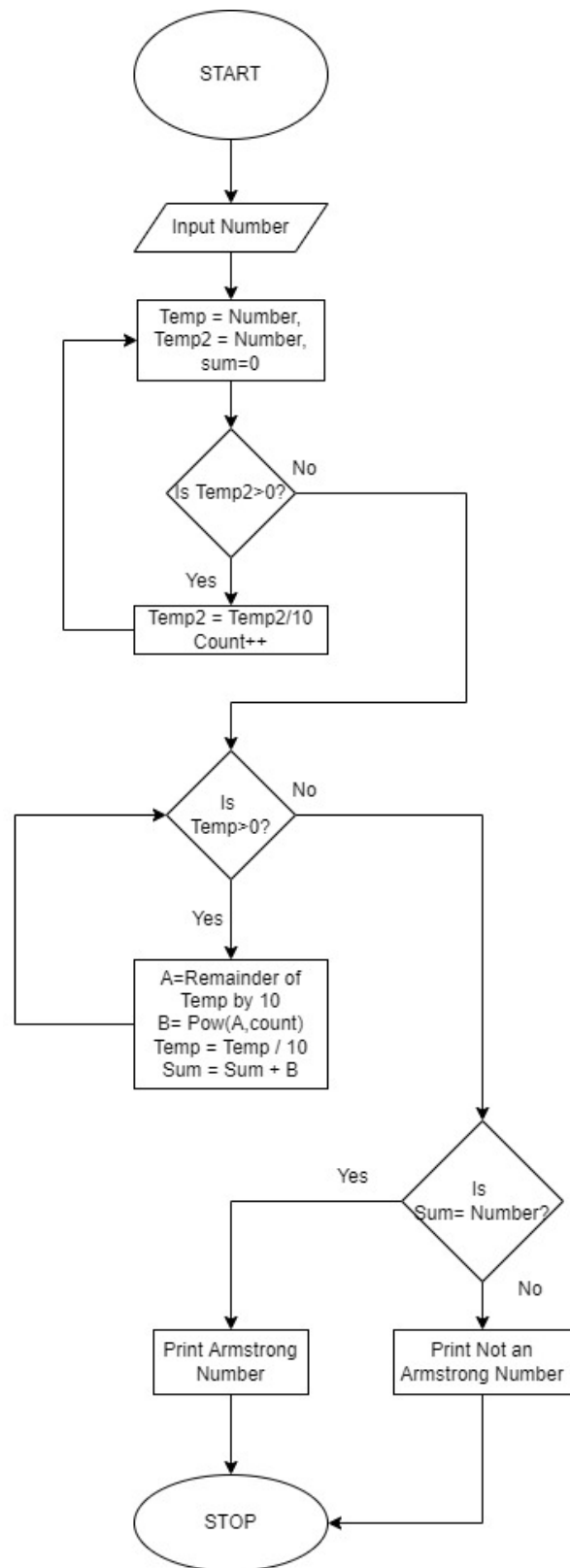
**PROBLEM
STATEMENT:**

Write a program to check whether a given number is Armstrong number or not. For Example 371 is $3^3+7^3+1^3=371$.

ALGORITHM:

1. START
2. Input Number
3. Get the number of digits by dividing the number by 10 and incrementing a counter everytime it is divided.
4. Get Each Digit by finding the Remainder of Number by 10
5. Raise the Digit to the number of digits and add to temporary variable sum whose value is 0
6. Divide by 10 to remove the earlier digit
7. Repeat 3,4,5 till All digits are removed
8. If Sum = Original Number
 Print Armstrong Number
 Else
 Print Not an Armstrong Number
9. STOP

FLOWCHART:



PROGRAM:	<pre> #include<stdio.h> #include<conio.h> #include<math.h> int main() { int n,a,b,c,sum=0,temp,count; printf("Enter the Number: "); scanf("%d",&n); temp=n; int temp2 = n; while(temp2>0) { temp2 /= 10; count++; } for(int i=0;temp>0;i++) { a=temp%10; b=pow(a,count); sum=sum+b; temp /= 10; } if(sum==n) { printf("%d is an Armstrong Number",n); } else { printf("%d is not an Armstrong Number",n); } return 0; } </pre>
RESULT:	<div> <div> Enter the Number: 371 371 is an Armstrong Number ...Program finished with exit code 0 Press ENTER to exit console. </div> <div> Enter the Number: 22 22 is not an Armstrong Number ...Program finished with exit code 0 Press ENTER to exit console. </div> </div>
CONCLUSION:	<p>In this experiment, we learned about the various types of Control Structures like Loops and Operators as well as their uses. We</p>

	learned in detail about For, While and Do-While Loops as well as all the Logical, Relational and Arithmetic Operators.
--	--