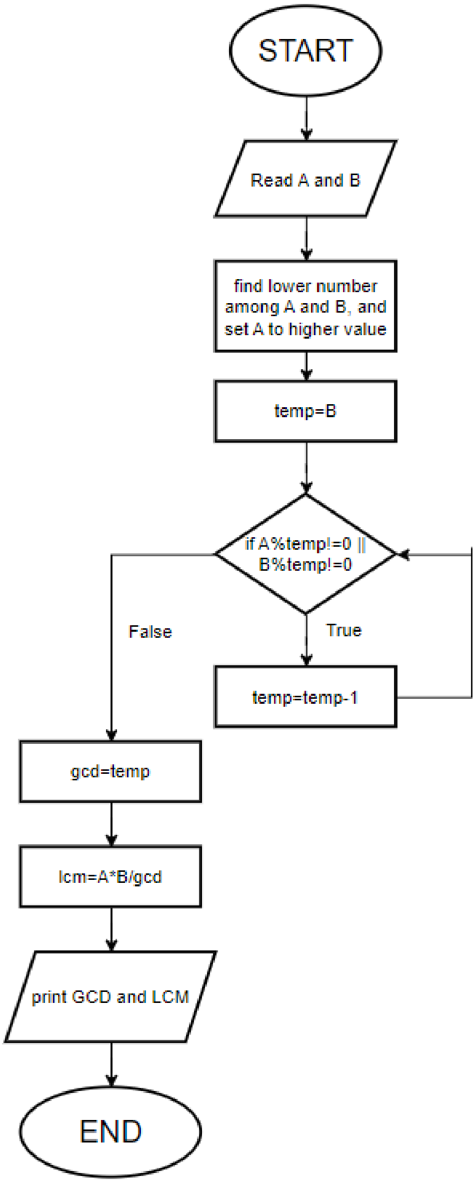
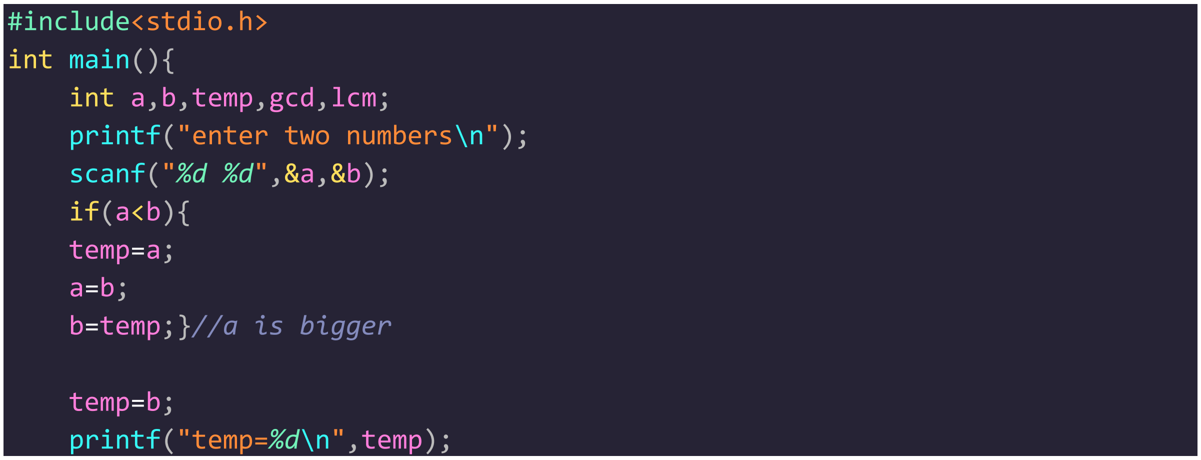
Name	Shubhan Singh
UID no.	2022300118
Experiment No.	2 (Control structures)

AIM:	To apply various control structures to solve given problems.
Program 1	
PROBLEM STATEMENT :	Take two numbers as input and calculate their LCM and GCD (HCF).
ALGORITHM:	Step 1: START Step 2: Read two numbers as input A and B. Step 3: Find the bigger of the two numbers Step 4: Set a temporary variable temp equal to the lower value. Step 5: Take modulus of both numbers with the temporary variable. Step 6: If any of the operations does not return zero, decrement temp by 1, and return to Step 5. Step 7: When both operations return 0, the value of temp is the GCD of the two numbers. Step 8: Calculate LCM using the formula LCM=A*B/GCD Step 9: Print GCD and LCM as output. Step 10: END.

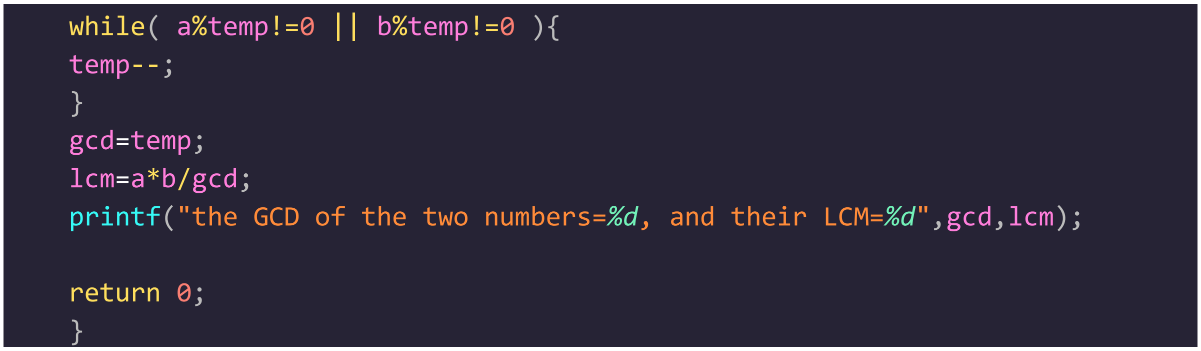
FLOWCHART:


START
Read A and B
find lower number among A and B, and set A to higher value
temp=B
if A%temp!=0 || B%temp!=0
False
True
temp=temp-1
gcd=temp
lcm=A*B/gcd
print GCD and LCM
END

PROGRAM:

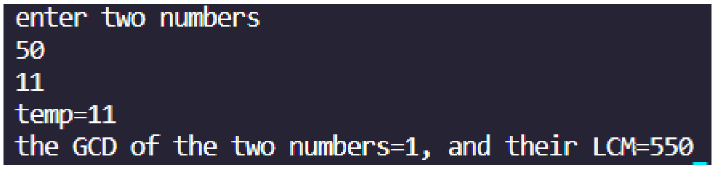


```
#include<stdio.h>
int main(){
    int a,b,temp,gcd,lcm;
    printf("enter two numbers\n");
    scanf("%d %d",&a,&b);
    if(a<b){
    temp=a;
    a=b;
    b=temp;}//a is bigger

    temp=b;
    printf("temp=%d\n",temp);
```



```
    while( a%temp!=0 || b%temp!=0 ){
    temp--;
    }
    gcd=temp;
    lcm=a*b/gcd;
    printf("the GCD of the two numbers=%d, and their LCM=%d",gcd,lcm);

    return 0;
    }
```

RESULT:



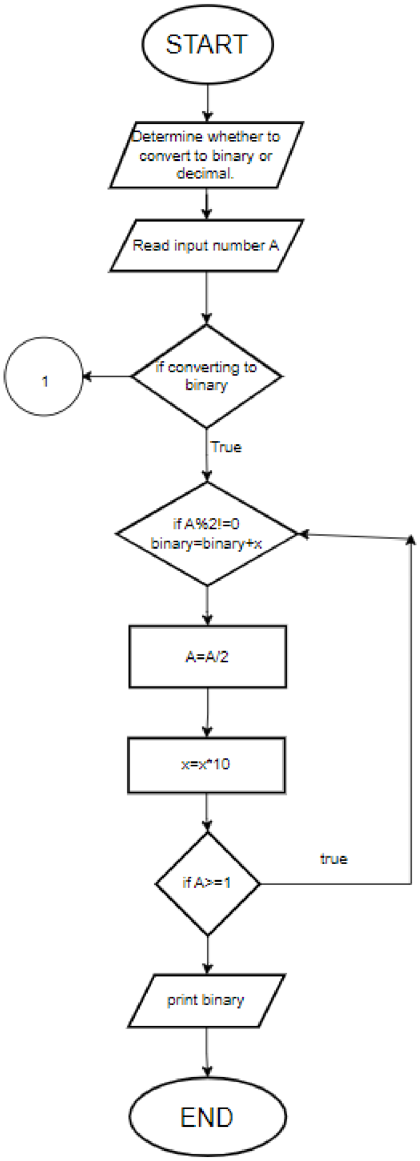
```
enter two numbers
50
11
temp=11
the GCD of the two numbers=1, and their LCM=550
```

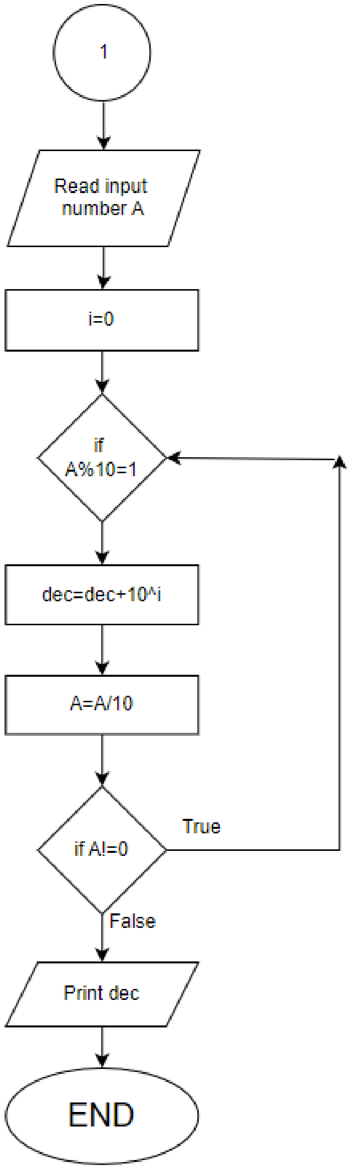
Program 2

PROBLEM STATEMENT : Write a program to convert a decimal number to binary or convert a binary number to decimal

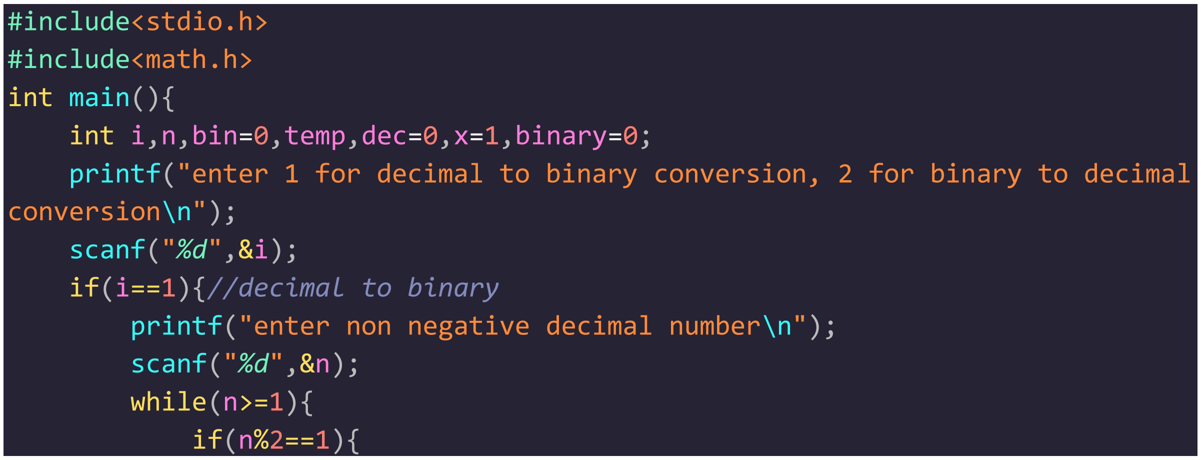
ALGORITHM:

```
Step 1: Start
Step 2: Determine whether to convert input to binary or decimal.
Step 3: Read input number.
If converting to binary,
Step 4a: If number is not divisible by 2, binary=binary+x , here, binary is a variable initialized to 0.
Step 5a: divide the number by 2
Step 6a: multiply x by 10
Step 7a: return to step 4a if number is greater than or equal to 1.
Step 8a: Print the value of binary.
If converting to decimal,
Step 4b: Read input binary number A.
Step 5b: declare a variable i initialized to 0.
Step 6b: if A%10 equals 1, dec=dec+10^i ,here, dec is a variable initialized to 0
Step 7b: divide number by 10.
Step 8b: If number is not equal to 0, return to step 6b.
Step 9b: Print value of dec.

Step 10: END
```

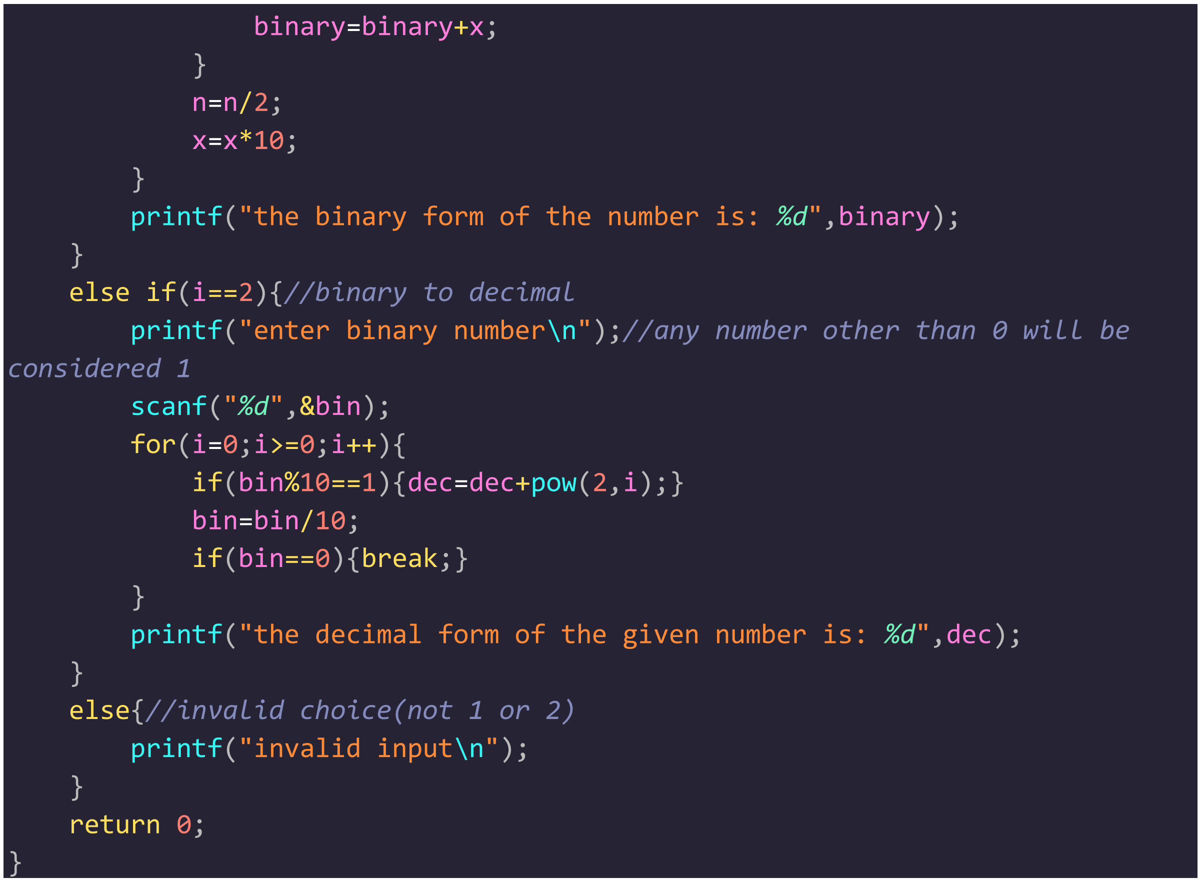
FLOWCHART:	START Determine whether to convert to binary or decimal. Read input number A if converting to binary 1 True if A%2!=0 binary=binary+x A=A/2 x=x*10 if A>=1 true print binary END



PROGRAM:

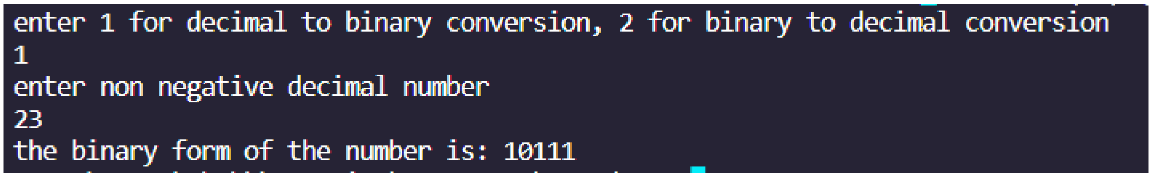


```
#include<stdio.h>
#include<math.h>
int main(){
    int i,n,bin=0,temp,dec=0,x=1,binary=0;
    printf("enter 1 for decimal to binary conversion, 2 for binary to decimal
conversion\n");
    scanf("%d",&i);
    if(i==1){//decimal to binary
        printf("enter non negative decimal number\n");
        scanf("%d",&n);
        while(n>=1){
            if(n%2==1){
```

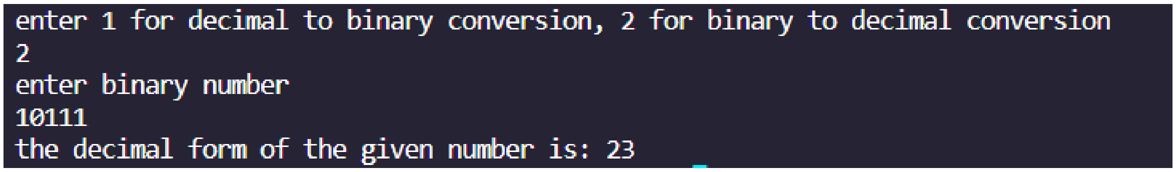


```
                binary=binary+x;
            }
            n=n/2;
            x=x*10;
        }
        printf("the binary form of the number is: %d",binary);
    }
    else if(i==2){//binary to decimal
        printf("enter binary number\n");//any number other than 0 will be
considered 1
        scanf("%d",&bin);
        for(i=0;i>=0;i++){
            if(bin%10==1){dec=dec+pow(2,i);}
            bin=bin/10;
            if(bin==0){break;}
        }
        printf("the decimal form of the given number is: %d",dec);
    }
    else{//invalid choice(not 1 or 2)
        printf("invalid input\n");
    }
    return 0;
}
```

RESULT:



```
enter 1 for decimal to binary conversion, 2 for binary to decimal conversion
1
enter non negative decimal number
23
the binary form of the number is: 10111
```



```
enter 1 for decimal to binary conversion, 2 for binary to decimal conversion
2
enter binary number
10111
the decimal form of the given number is: 23
```

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:

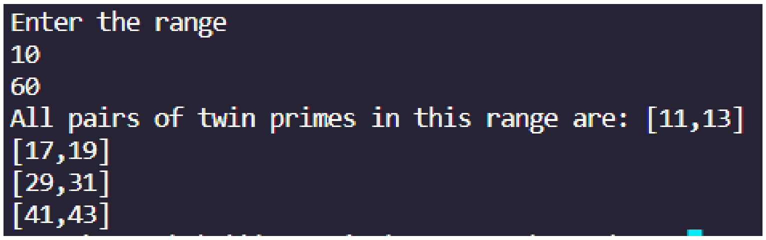
```
Step 1: START
Step 2: Read input for range A to B.
Step 3: If A<=2, A=3
Step 4: For i=A,
```

<table>
<tr>
<td></td>
<td>Step 5: for j=2,
Step 6: if i%j equals 0, increment flag variable count and go to step 9. (count is initialised at 2)
Step 7: increment j.
Step 8: if j<i/2, return to step 6.
Step 9: for x=2,
Step 10: if (i+2)%x equals 0, increment flag variable count and go to step 13. (count is initialised at 2)
Step 11: increment x.
Step 12: if x<i/2+1 , return to step 10.
Step 13: if count equals 2, print that i and i+2 are a pair of twin primes.
Step 14:increment i
Step 15: if i<=b/2, return to step 5.
Step 16: END</td>
</tr>
<tr>
<td><b>FLOWCHART:</b></td>
<td>Didn’t know how to draw this flowchart.</td>
</tr>
<tr>
<td><b>PROGRAM:</b></td>
<td>

```c
#include<stdio.h>

int main(){
    int a,b,j,x,i;
    printf("Enter the range\n");
    scanf("%d %d",&a,&b);
    if(a<0 && b<=0){
        printf("invalid input");
        return 0;
    }
    printf("All pairs of twin primes in this range are: ");
    if(a<=2){
        a=3;
    }
    if(a>2){
    for(i=a;i<=b-2;i++){
        int count=2;
        for(j=2;j<i/2;j++){
            if(i%j==0){count++;break;}
            }
        for(x=2;x<(i/2+1);x++){
            if((i+2)%x==0){count++;break;}
        }
        if(count==2){printf("[%d,%d]\n",i,i+2);};
    }
    }
return 0;
}
```

</td>
</tr>
</table>

RESULT:



```
Enter the range
10
60
All pairs of twin primes in this range are: [11,13]
[17,19]
[29,31]
[41,43]
```

Program 4

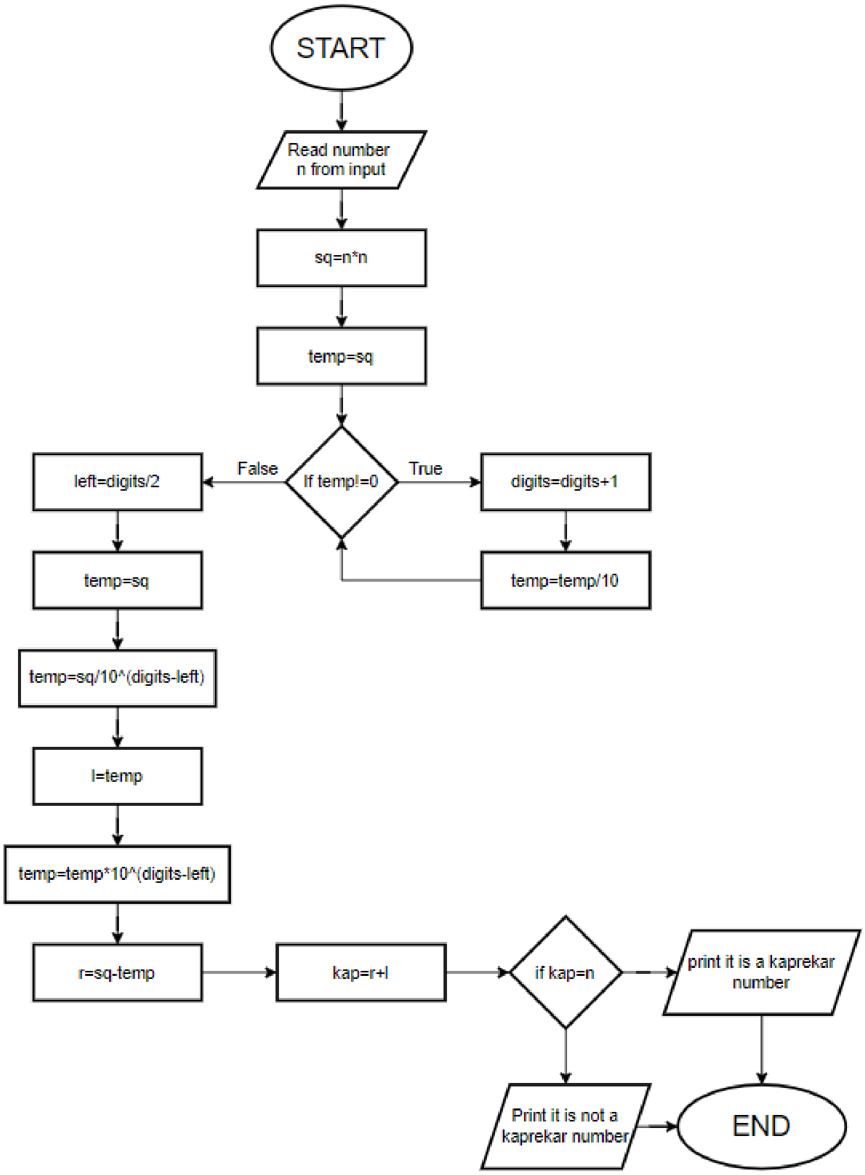
PROBLEM STATEMENT:

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.

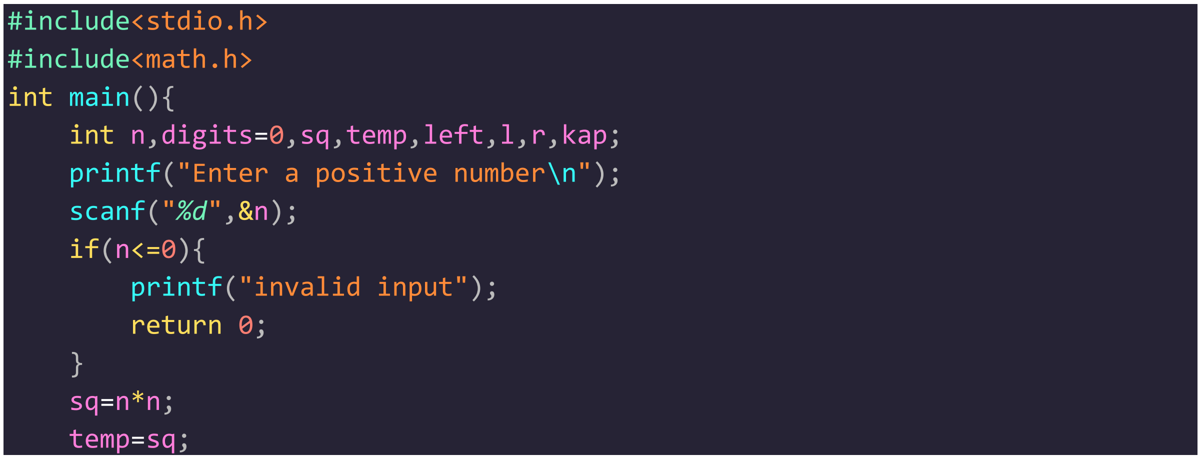
ALGORITHM:

Step 1: START
Step 2: read number from input
Step 3: square the number and store it in sq.
Step 4: make a temporary variable temp=sq
Step 5: if temp != 0, increment flag variable digits, else jump to step 7.(digits is initialized at 0)
Step 6: temp=temp/10, return to step 5
Step 7: left=digits/2
Step 8: temp=sq
Step 9: temp=sq/10^(digits-left)
Step 10: l=temp.
Step 11: temp=temp*10^(digits-left)
Step 12: r=sq-temp
Step 13: kap=r+l
Step 14: if kap equals the original input number, print that it is a kaprekar number
Step 15: else print it is not a kaprekar number.
Step 16: END

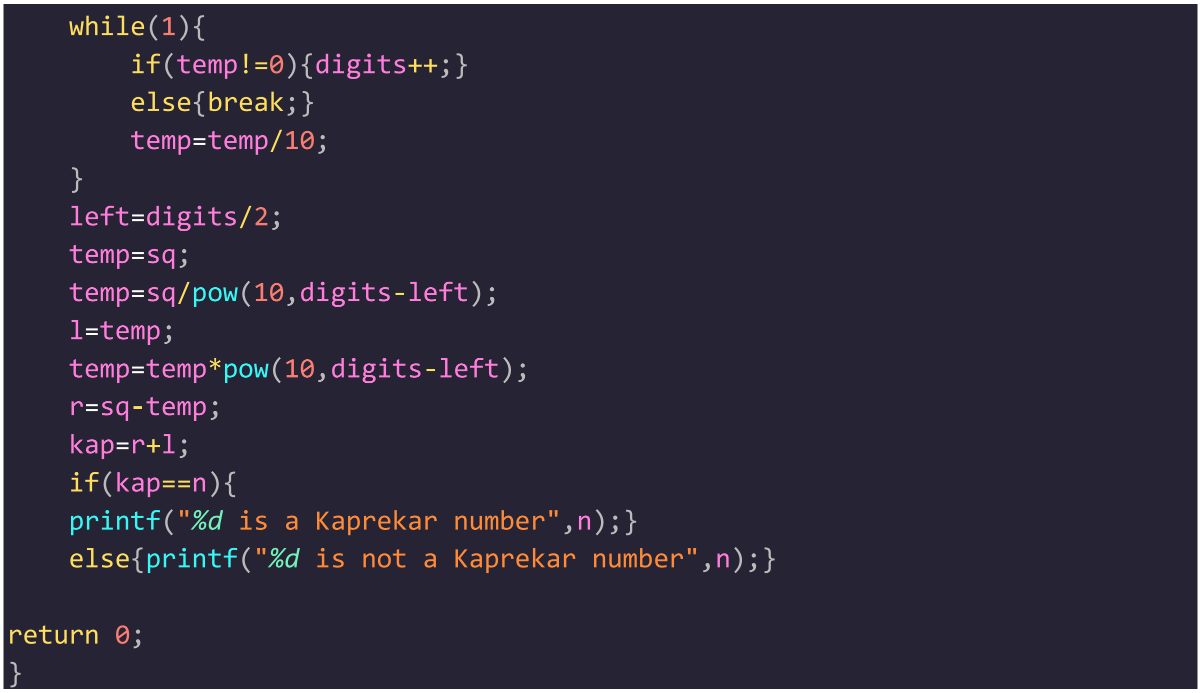
FLOWCHART:



PROGRAM:

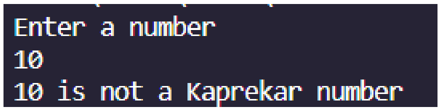


```
#include<stdio.h>
#include<math.h>
int main(){
    int n,digits=0,sq,temp,left,l,r,kap;
    printf("Enter a positive number\n");
    scanf("%d",&n);
    if(n<=0){
        printf("invalid input");
        return 0;
    }
    sq=n*n;
    temp=sq;
```

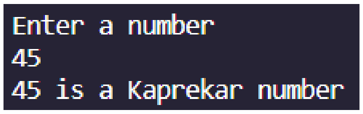


```
    while(1){
        if(temp!=0){digits++;}
        else{break;}
        temp=temp/10;
    }
    left=digits/2;
    temp=sq;
    temp=sq/pow(10,digits-left);
    l=temp;
    temp=temp*pow(10,digits-left);
    r=sq-temp;
    kap=r+l;
    if(kap==n){
    printf("%d is a Kaprekar number",n);}
    else{printf("%d is not a Kaprekar number",n);}

return 0;
}
```

RESULT:



```
Enter a number
10
10 is not a Kaprekar number
```



```
Enter a number
45
45 is a Kaprekar number
```