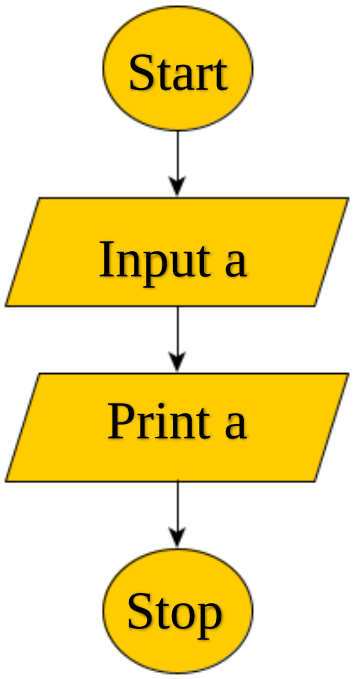
1. C Program to Print an Integer entered by user.

Algorithm –

Step 1. Start.
Step 2. Declare variable a.
Step 3. Inputting the value of a.
Step 4. Print a.
Step 5. Stop.

Flowchart –



Code –

```
 #include <stdio.h>
int main()
{
   int a;
   printf("Enter Number ");
   scanf("%d",&a);
   printf("Enter Number is = %d",a);
}
```

Output –

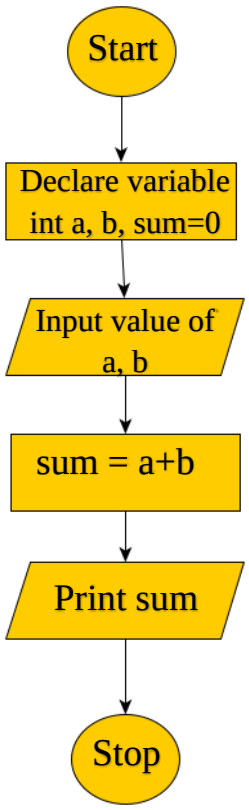
```
Enter Number 9
Enter Number is = 9
```

2. C Program to add two numbers

Algorithm –

Step 1. Start.
Step 2. Declare variable a, b, sum.
Step 3. Inputting the values of a and b.
Step 4. Add a, b and assign the value to the sum.
Step 5. Print sum.
Step 6. Stop

Flowchart –



Code –

```
#include <stdio.h>
int main()
{
   int a,b;
   int sum=0;
   printf("Enter Number 1\n");
   scanf("%d", &a);
   printf("Enter Number 2\n");
   scanf("%d", &b);
   sum=a+b;
   printf("Sum is %d\n", sum);
}
```

Output –

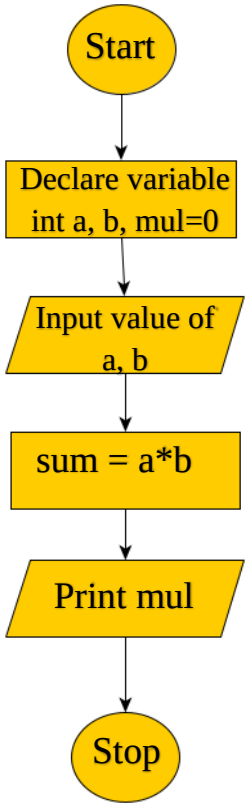
```
Enter Number 1
3
Enter Number 2
7
Sum is 10
```

3. C Program to multiply two numbers

Algorithm –

Step 1. Start
Step 2. Declare variable a, b, mul
Step 3. Inputting the values of a and b
Step 4. Multiply a, b and assign the value to the mul
Step 5. Print mul
Step 6. Stop

Flowchart –



Code –

```
#include <stdio.h>
int main()
{
   int a,b;
   int mul;
   printf("Enter Number 1\n");
   scanf("%d", &a);
   printf("Enter Number 2\n");
   scanf("%d", &b);
   mul=a*b;
   printf("Multiple is %d\n", mul);
}
```

Output –

```
Enter Number 1
7
Enter Number 2
5
Multiple is 35
```

4. C Program to calculate average of 5 given numbers.

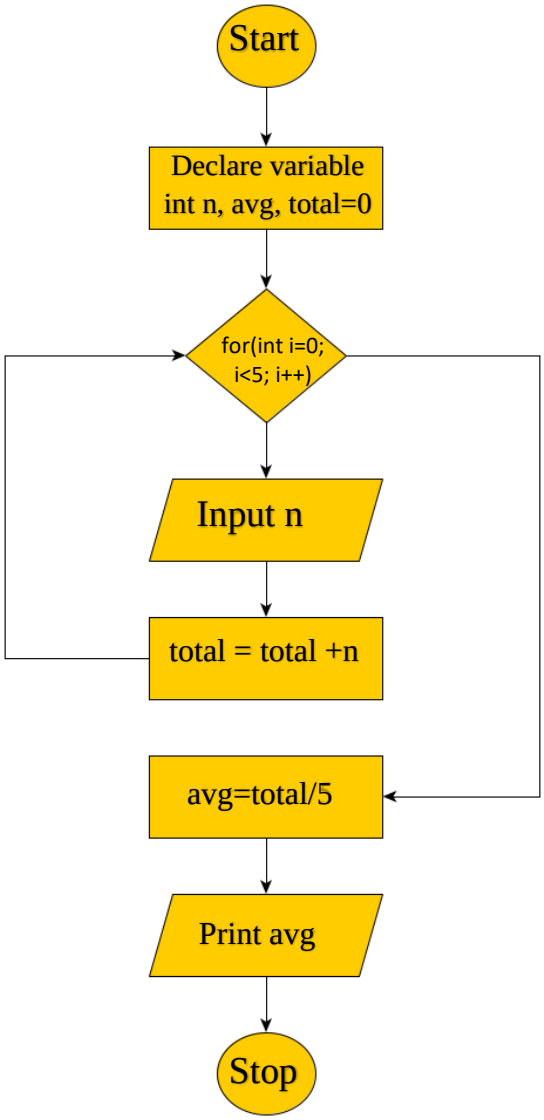
Algorithm –

Step 1. Start
Step 2. Declare variable int total, n, avg.
Step 3. Initialize total = 0
Step 4. Using for loop to input value of n till n<5.

Add the value of n to the total.

Step 5. Divide the value of total by 5 and assign the value to avg.
Step 6. Print the value of avg.
Step 7. Stop.

Flowchart –



Code –

```
#include <stdio.h>
int main()
{
    int total=0;
    int n,avg;
    printf("Enter 5 Number\n");
    for(int i=0; i<5; i++)
    {
        scanf("%d", &n);
        total+=n;
    }
    avg=total/5;
    printf("Average=%d",avg);
}
```

Output –

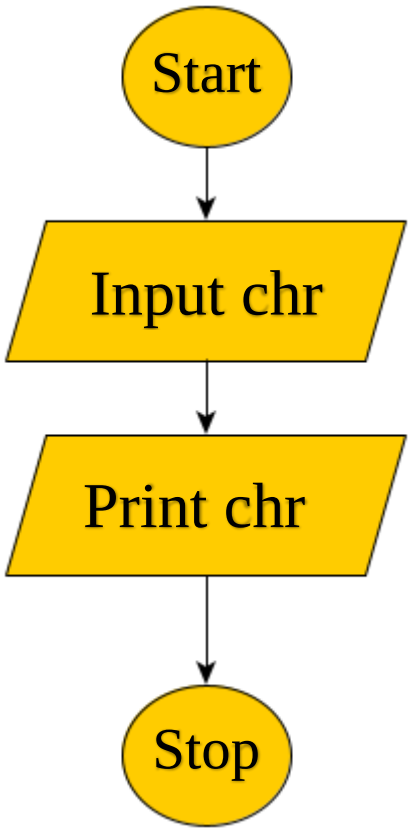
```
Enter 5 Number
1
2
3
4
5
Average=3
```

5. C Program to print ASCII value of character

Algorithm –

Step 1. Start
Step 2. Declare character variable as chr
Step 3. Inputting the value of chr
Step 4. Print the value of chr using “%d” format specifier
Step 5. Stop.

Flowchart –



Code –

```
#include <stdio.h>
int main()
{
   char chr;
   printf("Enter character to find its ASCII value\n");
   scanf("%c", &chr);
   printf("ASCII value is = %d", chr);
}
```

Output –

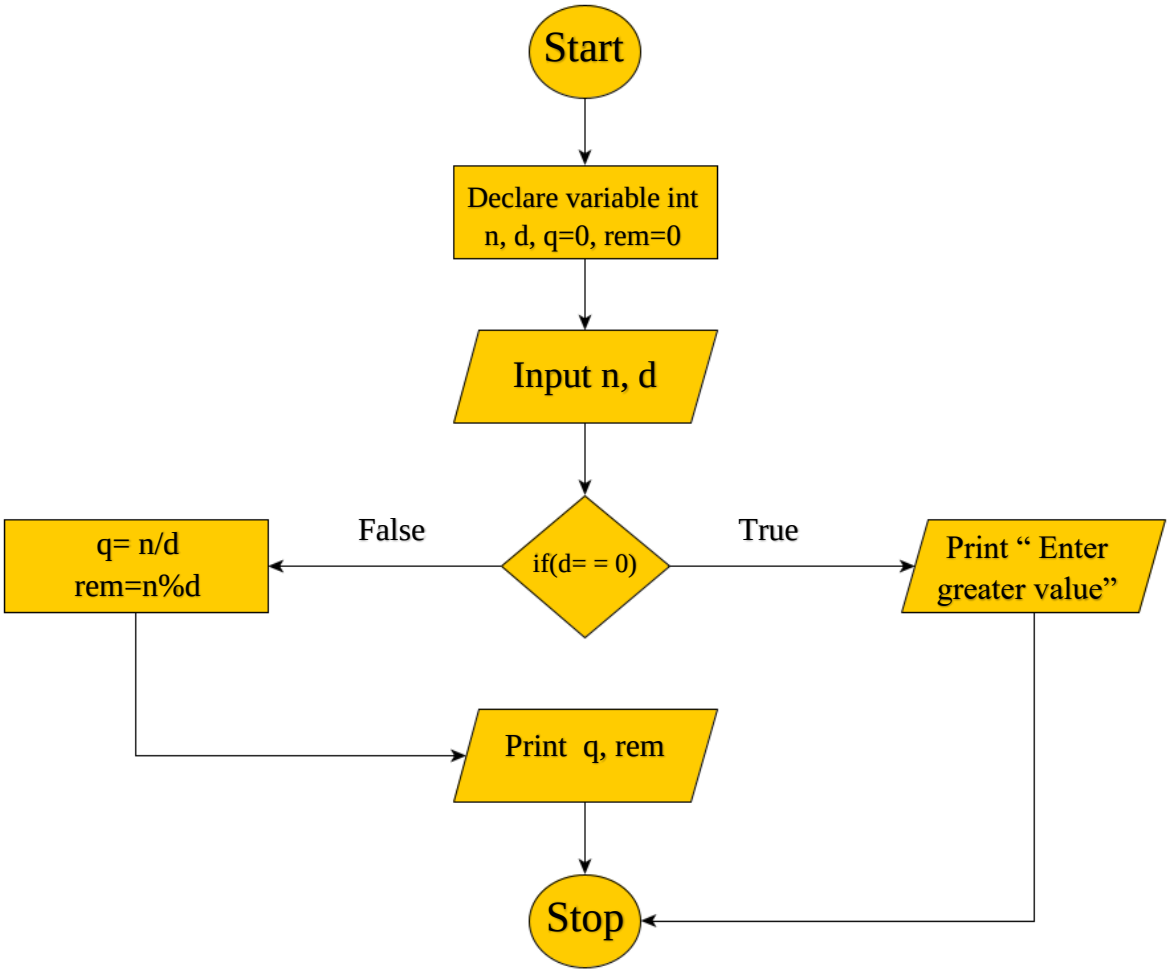
```
Enter character to find its ASCII value
E
ASCII value is = 69
```

6. C Program to print quotient and remainder if two integers are input by the user.

Algorithm –

Step 1. Start
Step 2. Declare variable n, d, q, rem.
Step 3. Initialize the value of q=0, rem=0.
Step 4. Inputting the value of n and d.
Step 4. if(d = = 0)
Print "Enter the value greater than zero"
Else
n will be divided by d and the value will be assigned to q (q = n/d)
rem will store the reminder (rem = n%d)
print the values of q and rem.
Step 5. Stop

Flowchart –



Code –

```
#include <stdio.h>
int main()
{
    int n,d;
    int q=0;
    int rem=0;
    printf("Enter First Number\n");
    scanf("%d", &n);
    printf("Enter Second Number\n");
    scanf("%d", &d);
    if(d==0)
    {
        printf("Enter value greater then ZERO\n");
    }
    else
    {
        q=n/d;
        rem=n%d;
        printf("Quotient = %d, Remainder = %d\n",q,rem);
    }
```

Output –

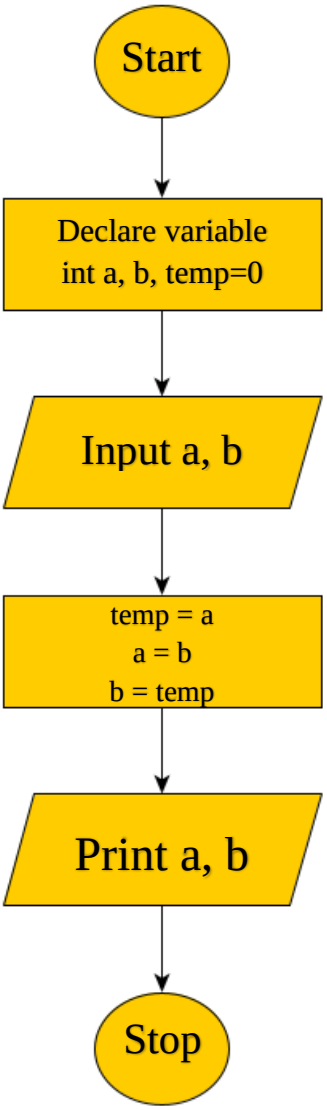
```
Enter First Number
65
Enter Second Number
8
Quotient = 8, Remainder = 1
```

7. C Program to swap two numbers.

Algorithm –

Step 1. Start
Step 2. Declare the int variables a, b.
Step 3. Initialize the value of temp = 0
Step 4. Input the values of a, b from user and assign the values
Step 4. Assign temp = a
a = b
and b = temp
Step 5. Print the values of a and b
Step 6. Stop.

Flowchart –



Code –

```
#include <stdio.h>
int main()
{
    int a,b;
    int temp=0;
    printf("Enter First Number\n");
    scanf("%d", &a);
    printf("Enter Second Number\n");
    scanf("%d", &b);
    temp=a;
    a=b;
    b=temp;
    printf("After Changing\n");
    printf("First Number is %d and Second Number is %d", a,b);
}
```

Output –

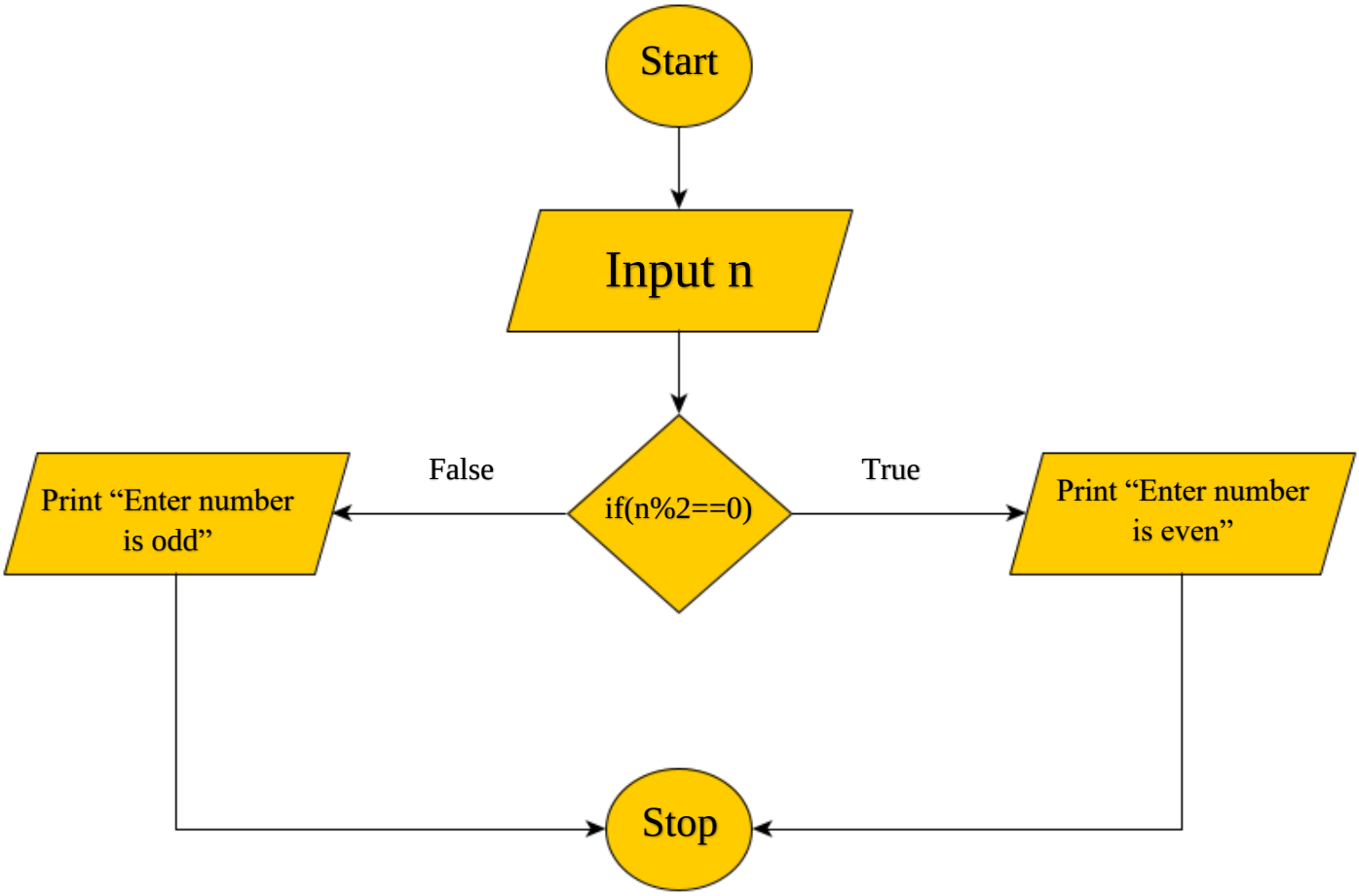
```
Enter First Number
9
Enter Second Number
4
After Changing
First Number is 4 and Second Number is 9
```

8. C Program to check if number is odd or even

Algorithm –

Step 1. Start
Step 2. Declare integer variable n
Step 3. Inputting the value of n
Step 4. if n%2 == 0
Print "Entered number is even"
else
Print "Entered number is odd"
Step 5. Stop

Flowchart –



Code –

```
#include <stdio.h>
int main()
{
   int n;
   printf("Enter Number\n");
   scanf("%d", &n);
   if(n%2==0)
   {
      printf("Number is even\n");
   }
   else
   {
      printf("Number is odd\n");
   }

}
```

Output –

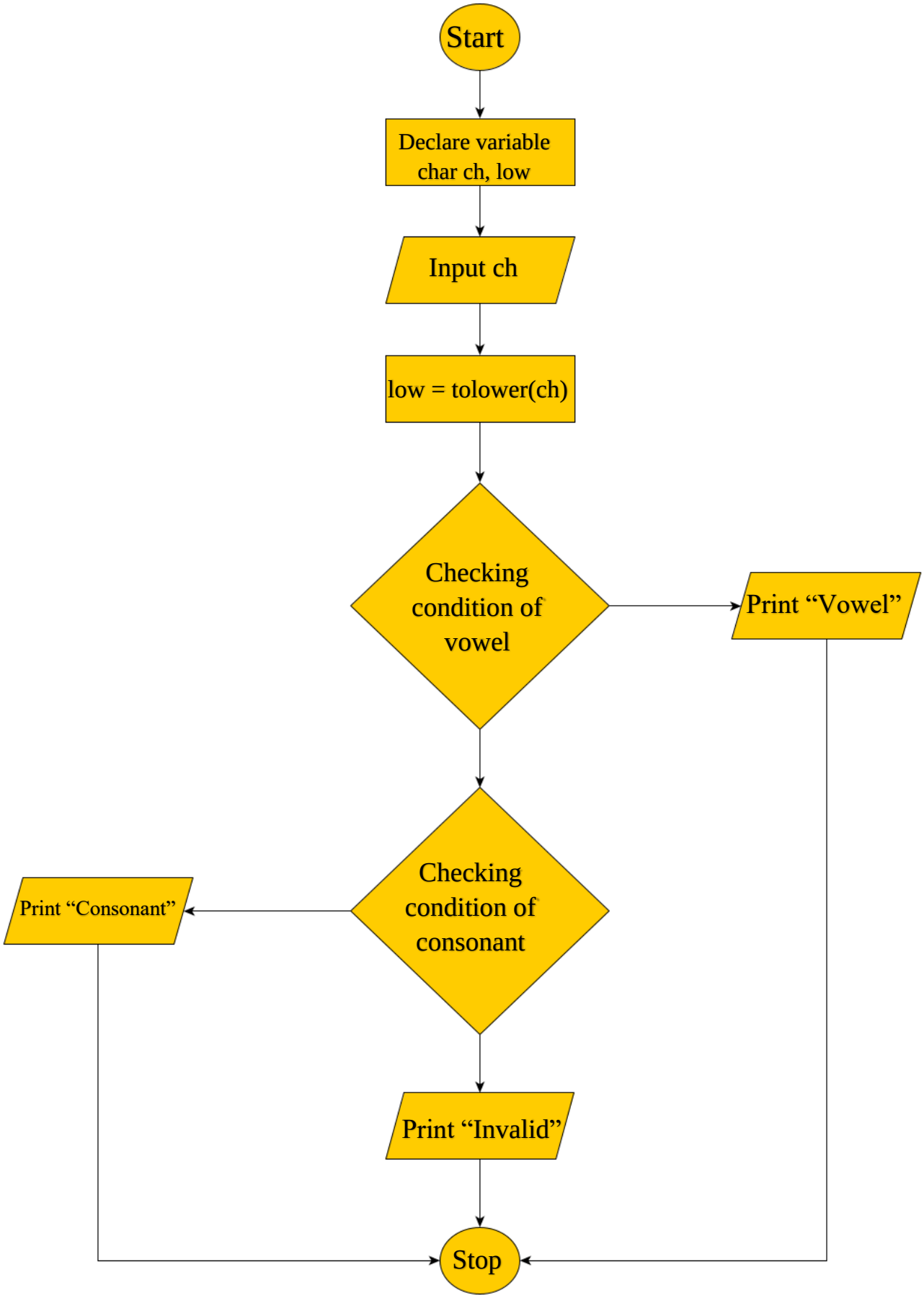
```
Enter Number
3
Number is odd
```

9. C Program to check if entered character is vowel or consonants

Algorithm –

Step 1. Start
Step 2. Declare the character variable ch and low
Step 3. Inputting the value of ch
Step 4. Convert the ch to lowercase (low=tolower(ch))
Step 5. if(low=='a' || low=='e'|| low=='i' || low=='o' || low=='u')
print "Entered character is vowel"
else if (low>='a' && low<='z')
print "Entered character is a consonant "
else
print “Invalid input”
Step 6. Stop

Flowchart –


Start
Declare variable char ch, low
Input ch
low = tolower(ch)
Checking condition of vowel
Print “Vowel”
Checking condition of consonant
Print “Consonant”
Print “Invalid”
Stop

Code –

```
#include <ctype.h>
int main()
{
   char ch,low;
   printf("Enter character\n");
   scanf("%c", &ch);
   low=tolower(ch);
   if(low=='a' || low=='e'|| low=='i' || low=='o' || low=='u')
   {
      printf("Entered character is vowel\n");
   }
   else if(low>='a' && low<='z')
   {
      printf("Entered character is consonants\n");
   }
   else
   {
      printf("Invalid input");
   }
```

Output –

```
Enter character
E
Entered character is vowel
```

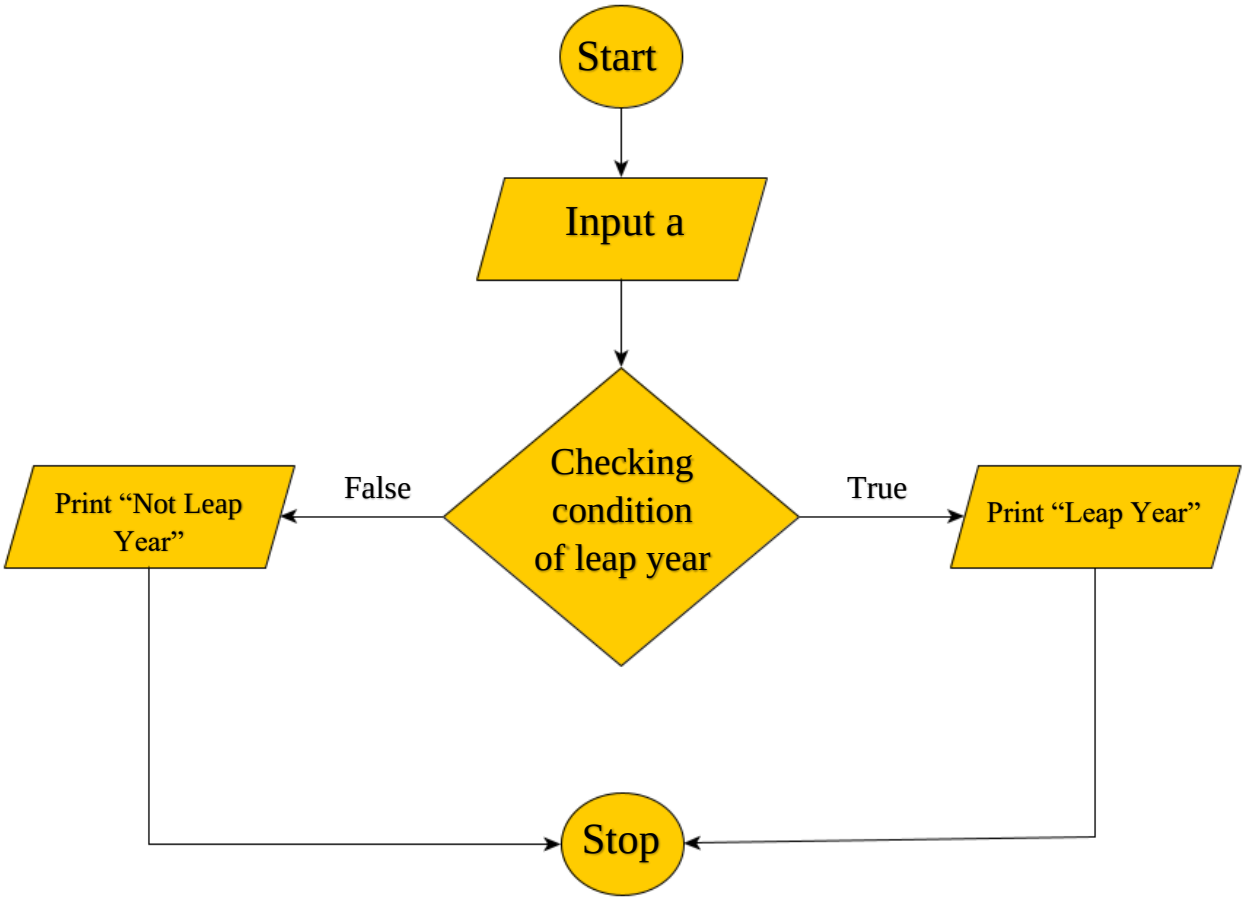
10. C Program to check if given year is leap year or not

Algorithm –

Step 1. Start
Step 2. Declare variable int a
Step 3. Inputting the value of a.
Step 4. if(a%100!=0 && a%4==0)

print "It's a leap year"
else
print "Not a leap year"
Step 5. Stop.

Flowchart –



Code –

```
#include <stdio.h>
int main()
{
   int a;
   printf("Enter Year\n");
   scanf("%d", &a);
    if(a%100!=0 && a%4==0)
   {
      printf("It’s a leap Year\n");
   }
   else
   {
      printf("Not a Leap Year\n");
   }

}
```

Output –

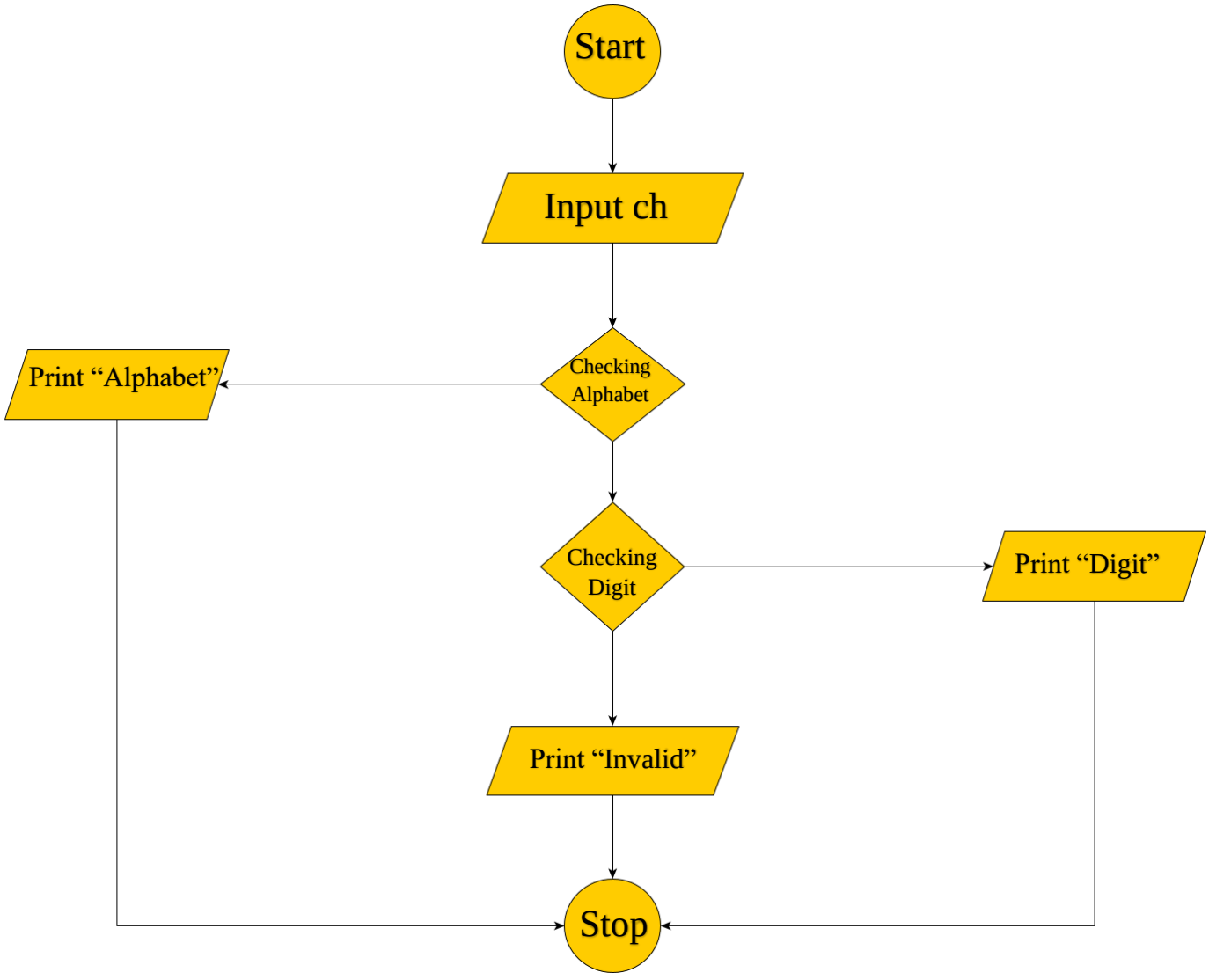
```
Enter Year
2012
Leap Year
```

11. C Program to check if given character is digit or alphabet

Algorithm –

Step 1. Start.
Step 2. Declare variable ch.
Step 3. Initialize the value of ch.
Step 4. If ch is between 64 and 91 OR ch is between 98 and 123 print “Entered character is alphabet”
Step 5. If ch is between 47 and 58 print “Entered character is digit”
Step 6. else print “Invalid input”
Step 7. Stop.

Flowchart –



Code –

```
#include <stdio.h>
int main()
{
   char ch;
   printf("Enter character\n");
   scanf("%c", &ch);
   if(ch>64 && ch<91 || ch>98 && ch<123)
   {
      printf("Entered character is alphabet\n");
   }
   else if( ch>47 && ch<58)
   {
      printf("Entered character is digit\n");
   }
   else
   {
      printf("Enter Valid input\n");
   }
}
```

Output –

```
Enter character
u
Entered character is alphabet
```

12. C Program to convert decimal number to binary number

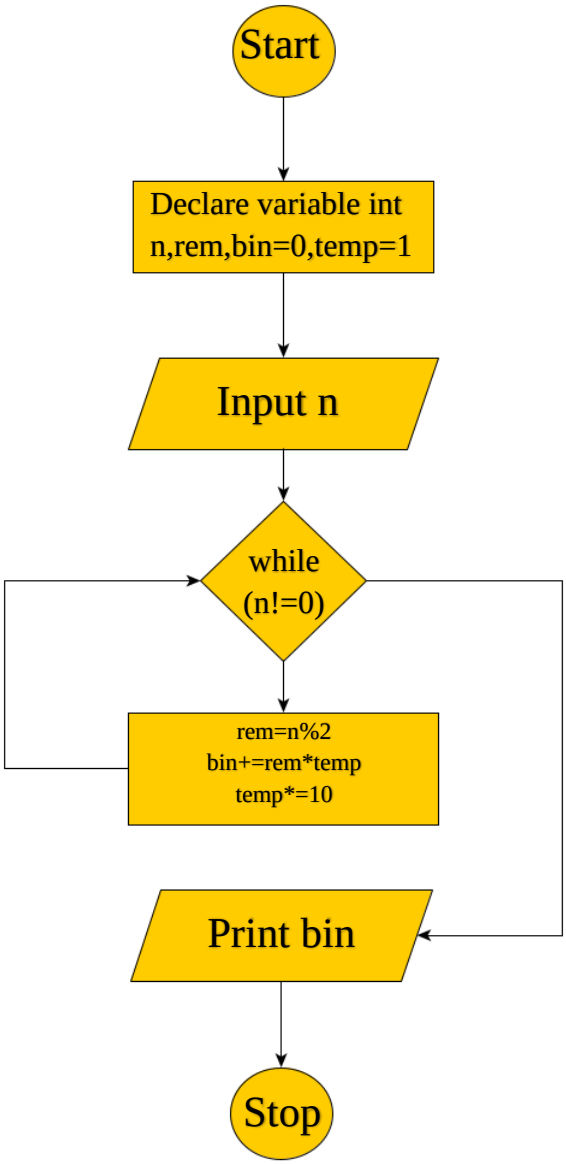
Algorithm –

Step 1. Start.
Step 2. Declare variable n, rem, bin, temp.
Step 3. Initialize the value of bin=0, temp=1.
Step 4. Inputting the value of n.
Step 5. Using while loop for repeating till n=0.

Finding reminder of n (rem=n%2)
Storing value of reminder in bin.

Step 6. Print the value of bin.
Step 7. Stop.

Flowchart –



Code –

```
#include <stdio.h>
int main()
{
   int n, rem;
   int bin=0;
   int temp=1;
   printf("Enter Number\n");
   scanf("%d", &n);
   while(n!=0)
   {
      rem=n%2;
      bin=bin+rem*temp;
      temp=temp*10;
      n=n/2;
   }
   printf("Binary = %d", bin);
}
```

Output –

```
Enter Number
20
Binary = 10100
```

13. C Program to convert decimal number to hexadecimal number

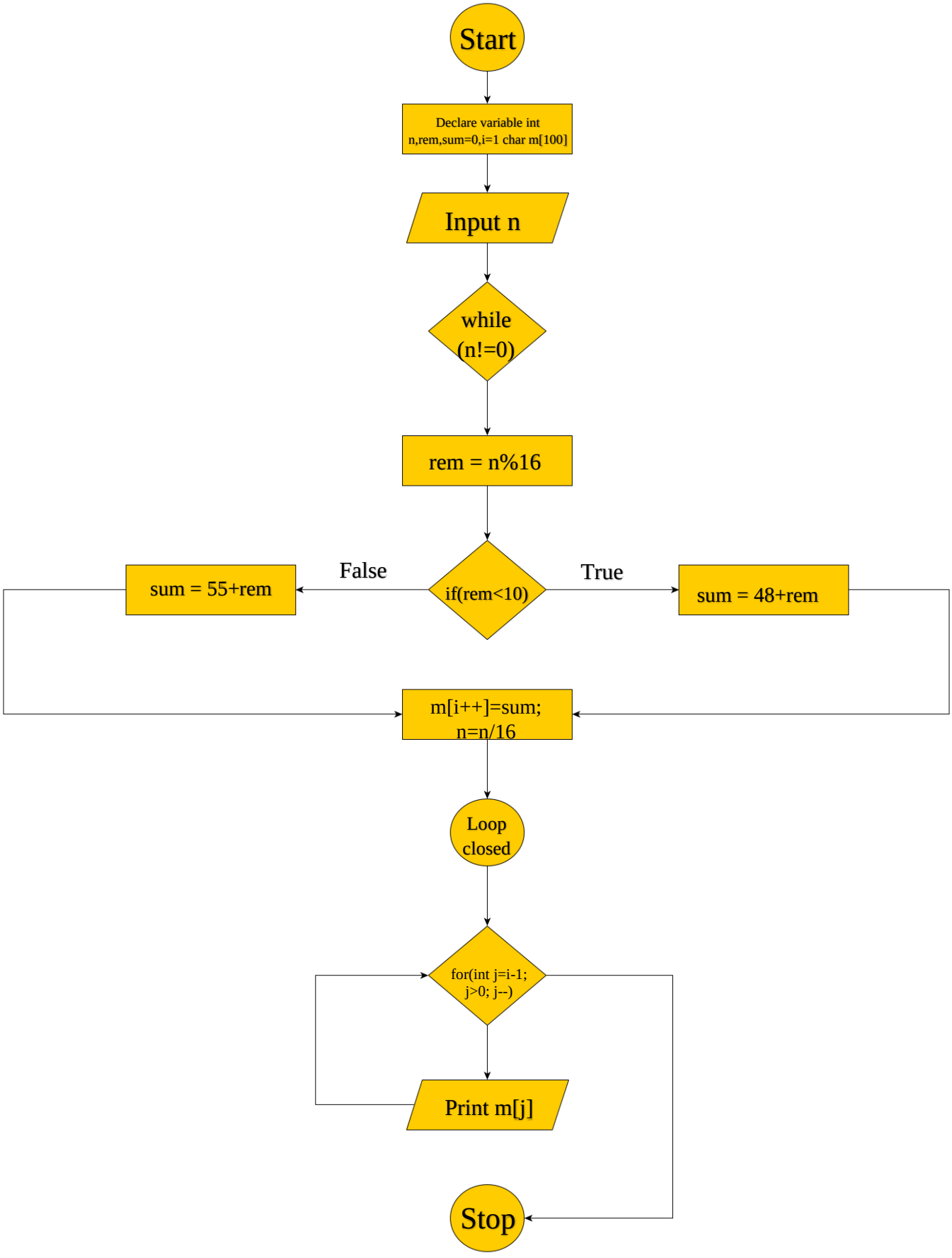
Algorithm –

Step 1. Start.
Step 2. Declare variable n, rem, sum, i, m.
Step 3. Initialize the value of sum=0, i=1, m[100].
Step 4. Inputting the value of n.
Step 5. Using loop till n=0.

Finding reminder of n (rem=n%16)
If reminder is smaller then 10 then sum store (48 + reminder).
If reminder is greater then 10 then sum store (55 + reminder).
Value of sum store in array (m[i++]=sum).

Step 6. Using inverse loop to print the value of array. Print “m[j]”.
Step 7. Stop.

Flowchart –


Start
Declare variable int n,rem,sum=0,i=1 char m[100]
Input n
while (n!=0)
rem = n%16
if(rem<10)
False
True
sum = 55+rem
sum = 48+rem
m[i++]=sum; n=n/16
Loop closed
for(int j=i-1; j>0; j--)
Print m[j]
Stop

Code –

```
#include <stdio.h>
int main()
{
   int n;
   int rem;
   int sum=0;
   int i=1;
   char m[100];
   printf("Enter Decimal Number\n");
   scanf("%d", &n);
   while(n!=0)
   {
      rem=n%16;
      if(rem<10)
      {
         sum=48+rem;
      }
      else
      {
         sum=55+rem;
      }
      m[i++]=sum;
      n=n/16;
   }
   for(int j=i-1; j>0; j--)
   {
      printf("%c",m[j]);
   }
}
```

Output –

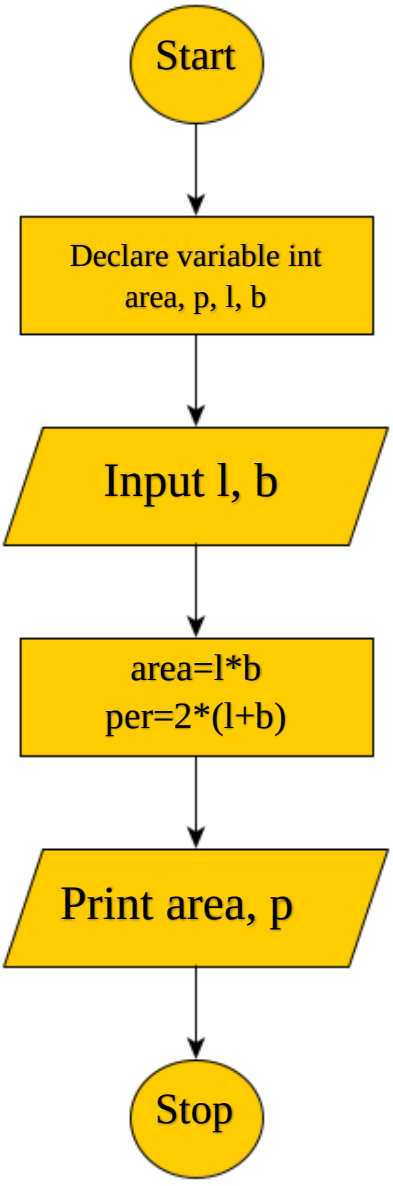
```
Enter Decimal Number
180
B4
```

14. C Program to compute the area and perimeter of a rectangle.

Algorithm –

Step 1. Start.
Step 2. Declare variable area, p, l, b.
Step 3. Inputting the value of l, b.
Step 4. Appling formula of area of rectangle (area=l*b).
Step 5. Appling formula of perimeter of rectangle (per=2*(l+b)).
Step 6. Print the value of area and perimeter.
Step 7. Stop.

Flowchart –



Code –

```
#include <stdio.h>
int main()
{
    int area, p;
    int l,b;
    printf("Enter Length\n");
    scanf("%d", &l);
    printf("Enter Breadth\n");
    scanf("%d", &b);
    area=l*b;
    p=2*(l+b);
    printf("Area=%d",area);
    printf("\nPerimeter=%d",p);
}
```

Output –

```
Enter Length
10
Enter Breadth
7
Area=70
Perimeter=34
```

15. C Program to convert decimal number to octal number.

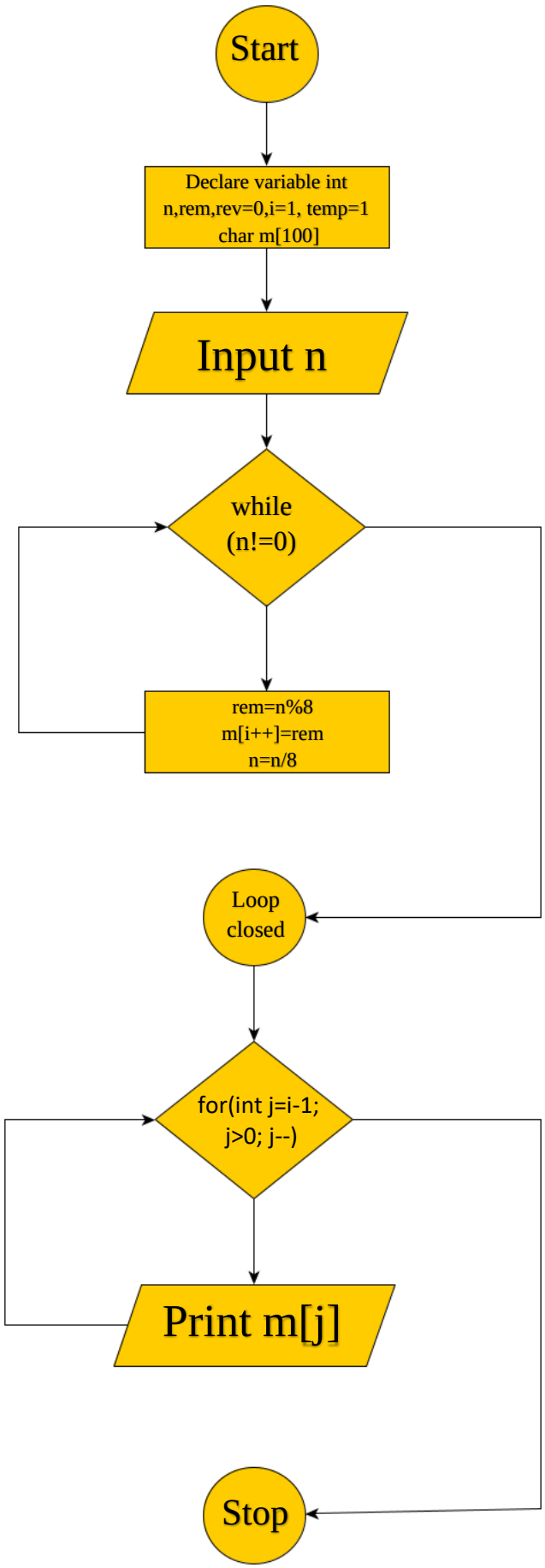
Algorithm –

Step 1. Start.
Step 2. Declare variable n, rem, rev, i, m, temp.
Step 3. Initialize the value of rev=0, temp=1, i=1, m[100].
Step 4. Inputting the value of n.
Step 5. Using loop till n=0.

Finding reminder of n (rem=n%8)
Value of reminder store in array (m[i++]=rem).

Step 6. Using inverse loop to print the value of array. Print “m[j]”.
Step 7. Stop.

Flowchart –


Start
Declare variable int
n,rem,rev=0,i=1, temp=1
char m[100]
Input n
while
(n!=0)
rem=n%8
m[i++]=rem
n=n/8
Loop
closed
for(int j=i-1;
j>0; j--)
Print m[j]
Stop

Code –

```
#include <stdio.h>
int main()
{
    int n,rem;
    int rev=0, temp=1;
    int i=1;
    int m[100];
    printf("Enter Decimal Number\n");
    scanf("%d",&n);
    while(n!=0)
    {
        rem=n%8;
        m[i++]=rem;
        n=n/8;
    }
    for(int j=i-1; j>0; j--)
    {
        printf("Octal is =%d", m[j]);
    }
}
```

Output –

```
Enter Decimal Number
200
Octal is = 310
```

16. C Program to convert binary number to decimal number

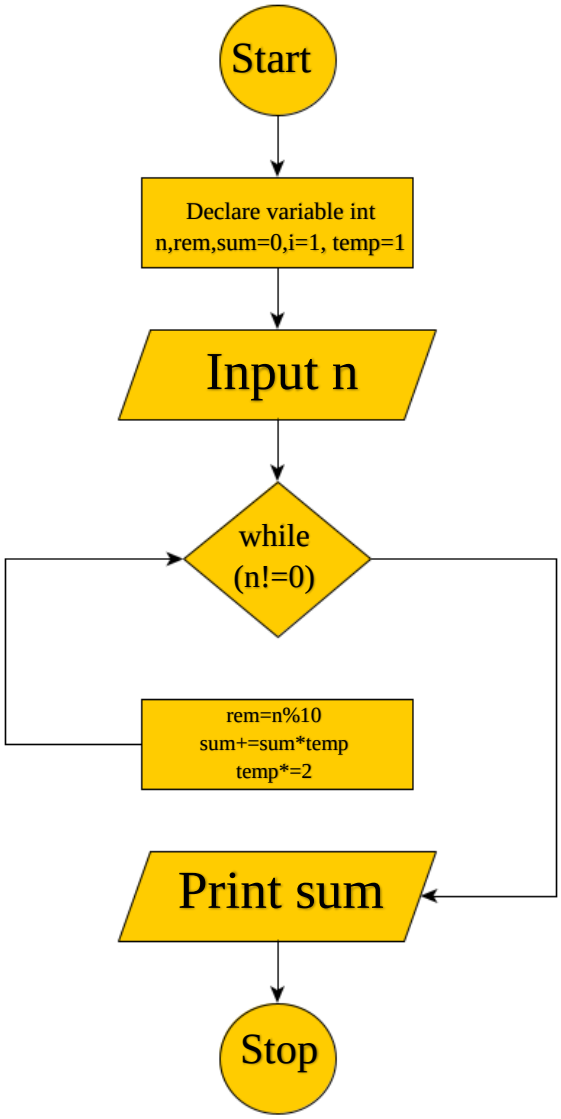
Algorithm –

Step 1. Start.
Step 2. Declare variable n, rem, sum, temp.
Step 3. Initialize the value of sum=0, temp=1.
Step 4. Inputting the value of n.
Step 5. Using while loop for repeating till n=0.

Finding reminder of n (rem=n%2)
Storing value of reminder in sum.

Step 6. Print the value of sum.
Step 7. Stop.

Flowchart –



Code –

```
#include <stdio.h>
int main()
{
    int n,rem,sum=0,temp=1;
    printf("Enter Number\n");
    scanf("%d", &n);
    while(n!=0)
    {
        rem=n%10;
        sum=sum+rem*temp;
        temp=temp*2;
        n=n/10;
    }
    printf("Decimal is = %d", sum);
}
```

Output –

```
Enter Number
101010
Decimal is = 42
```

17. C Program to convert hexadecimal number to decimal number

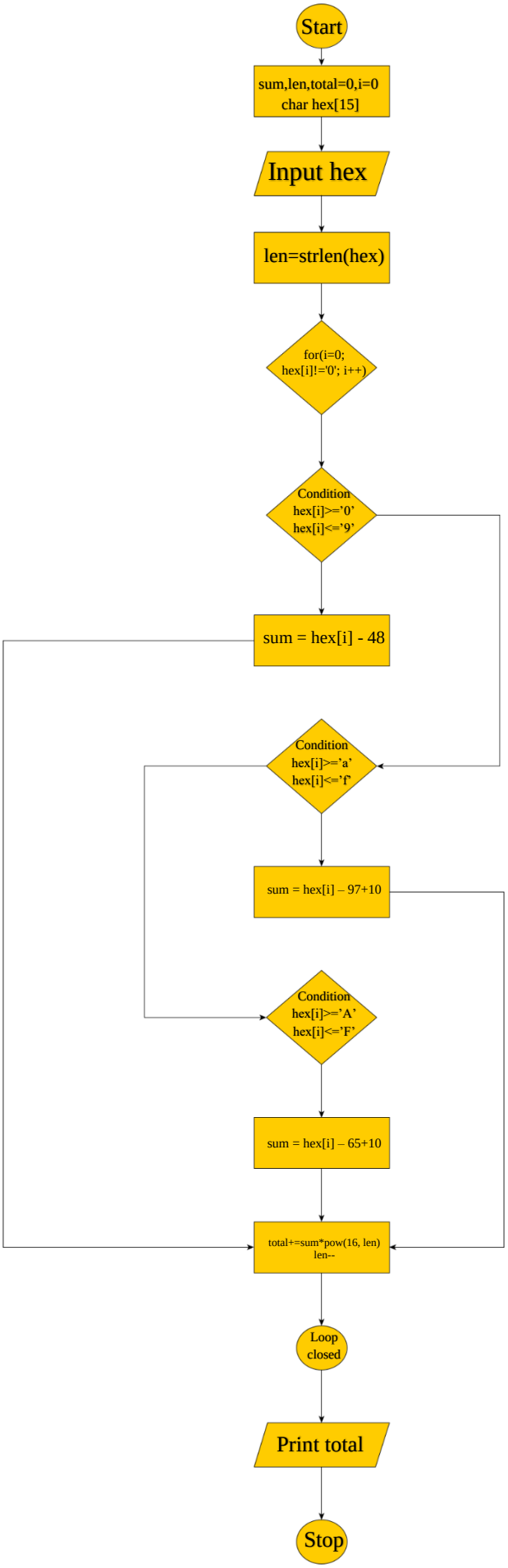
Algorithm –

Step 1. Start.
Step 2. Declare variable hex, sum, i, len, total.
Step 3. Initialize the value of total=0, i=1, hex[15].
Step 4. Inputting the value of hex.
Step 5. Storing the length of hex.
Step 6. Using for loop till hex[i]='0'.

If hex[i] greater then equal to '0' and hex[i] smaller then equal to '9' then sum = hex[i] – 48.
If hex[i] greater then equal to 'a' and hex[i] smaller then equal to 'f' then sum = hex[i] – 97 +10.
If hex[i] greater then equal to 'A' and hex[i] smaller then equal to 'F' then sum = hex[i] – 65 +10.
Total store the value of sum by multiply power 16 (total=total + sum*pow(16, len)).

Step 6. Print the value of total.
Step 7. Stop.

Flowchart –


Start
sum,len,total=0,i=0
char hex[15]
Input hex
len=strlen(hex)
for(i=0;
hex[i]!='0'; i++)
Condition
hex[i]>='0'
hex[i]<='9'
sum = hex[i] - 48
Condition
hex[i]>='a'
hex[i]<='f'
sum = hex[i] – 97+10
Condition
hex[i]>='A'
hex[i]<='F'
sum = hex[i] – 65+10
total+=sum*pow(16, len)
len--
Loop
closed
Print total
Stop

Code –

```
#include <stdio.h>
#include <math.h>
#include <string.h>
int main()
{
   char hex[15];
   int i=0;
   int sum,len;
   int total=0;
   printf("Enter Hexa\n");
   gets(hex);
   len=strlen(hex);
   len--;
    for(i=0; hex[i]!='0'; i++)
   {
      if(hex[i]>='0' && hex[i]<='9')
      {
         sum = hex[i] - 48;
      }
      else if(hex[i]>='a' && hex[i]<='f')
      {
         sum = hex[i] - 97 + 10;
      }
      else if(hex[i]>='A' && hex[i]<='F')
      {
         sum = hex[i] - 65 + 10;
      }

      total= total+ sum * pow(16, len);
      len--;
   }
   printf("Decimal Num=%d", total);
}
```

Output –

```
Enter Hexa
D6
Decimal Num=214
```

18. C Program to convert octal number to decimal number

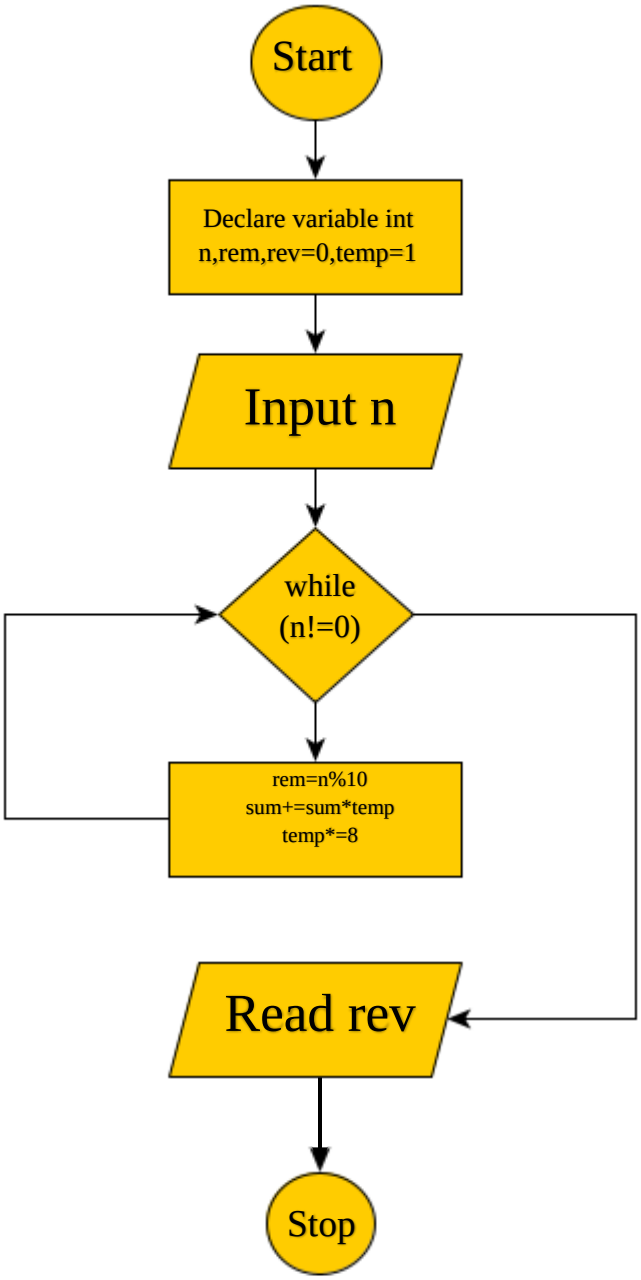
Algorithm –

Step 1. Start.
Step 2. Declare variable n, rem, rev, temp.
Step 3. Initialize the value of rev=0, temp=1.
Step 4. Inputting the value of n.
Step 5. Using while loop for repeating till n=0.

Finding reminder of n (rem=n%10).
Storing value of reminder in rev.

Step 6. Print the value of rev.
Step 7. Stop.

Flowchart –



Code –

```
#include <stdio.h>
int main()
{
    int n, rem;
    int rev=0, temp=1;
    printf("Enter Numer\n");
    scanf("%d", &n);
    while(n!=0)
    {
        rem=n%10;
        rev=rev+rem*temp;
        temp=temp*8;
        n=n/10;
    }
    printf("Decimal=%d", rev);
}
```

Output –

```
Enter Numer
167
Decimal=119
```

19. C Program to generate multiplication table

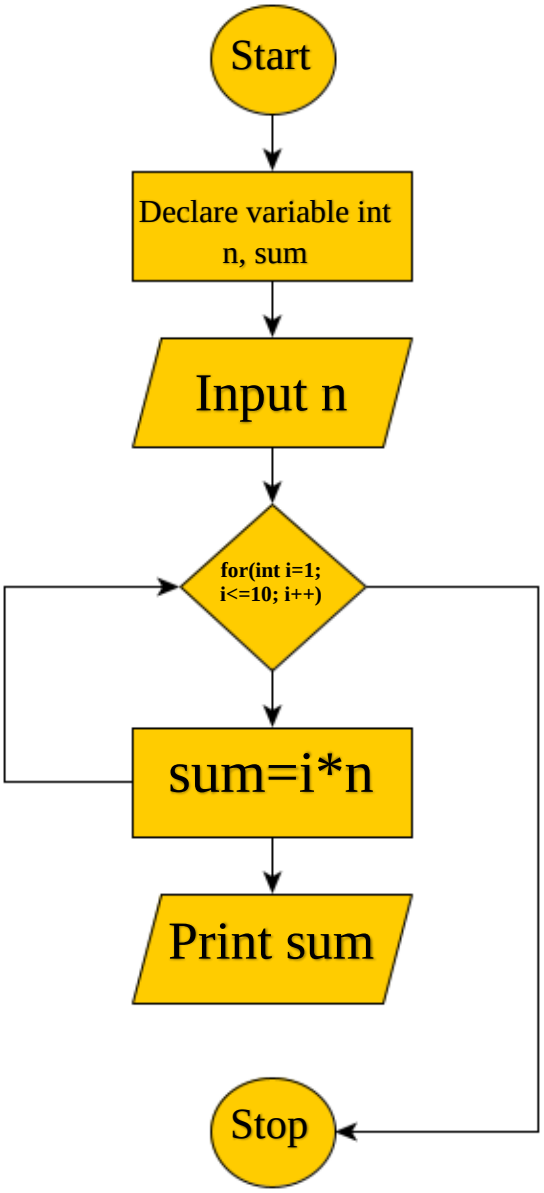
Algorithm –

Step 1. Start.
Step 2. Declare variable n, sum.
Step 3. Inputting the value of n.
Step 4. Using while loop for repeating till i<=10.

Sum storing the multiplication of n and i (sum=i*n).
Print the value of sum.

Step 5. Stop.

Flowchart –



Code –

```
#include <stdio.h>
int main()
{
    int n, sum;
    printf("Enter Number\n");
    scanf("%d", &n);
    for(int i=1; i<=10; i++)
    {
        sum=i*n;
        printf("\n%d", sum);
    }
}
```

Output –

```
Enter Number
8

8
16
24
32
40
48
56
64
72
80
```

20. C Program to print fibonacci series.

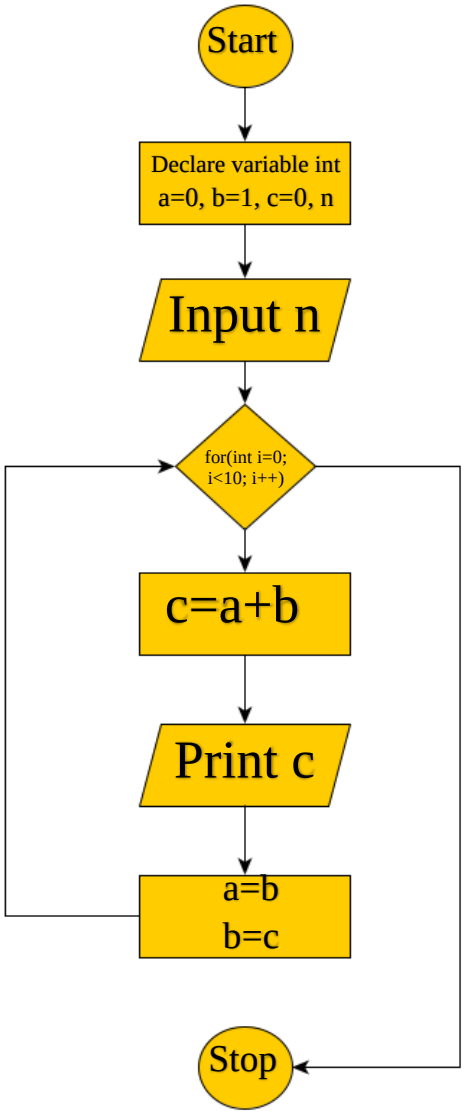
Algorithm –

Step 1. Start.
Step 2. Declare variable a, b, c.
Step 3. Inputting the value of a=0, b=1, c=0.
Step 4. Using while loop for repeating till i<8.

c storing the addition of a and b (c=a+b).
Print the value of c.
Changing the value of a into b then b into c.

Step 5. Stop.

Flowchart –



Code –

```
#include <stdio.h>
int main()
{
    int a=0, b=1, c=0;
    int n;
    printf(“Enter length of series”);
    scanf(“%d”, &n);
    for(int i=0; i<n; i++)
    {
        c=a+b;
        printf("\n%d", c);
        a=b;
        b=c;
    }
}
```

Output –

```
1
2
3
5
8
13
21
34
```