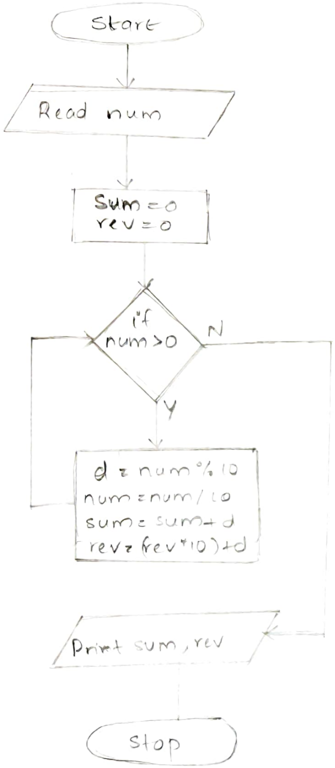


Sum and Reverse of a number

Aim: To find the sum and reverse of a number using C program

Algorithm

Step 1 : start

step 2 : read number num

Step 3 : set sum = 0 and rev = 0

Step 4 : Repeat steps 5 to 8 while num > 0

step 5 : set d = num % 10

step 6 : set num = num / 10

Step 7 : set sum = sum + d

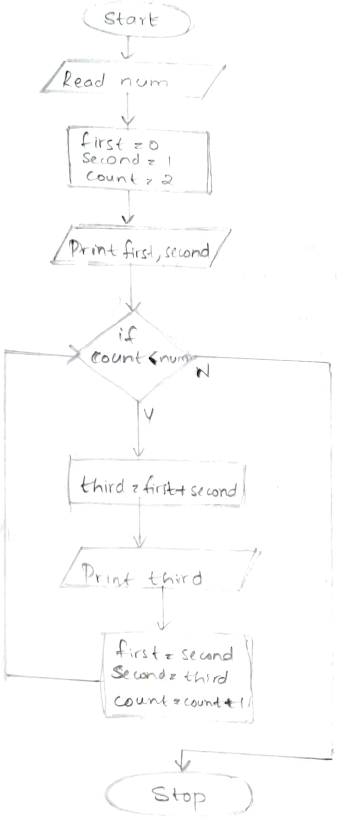
Step 8 : set rev = (rev × 10) + d

Step 9 : print sum, rev

Step 10 : stop

Program

```
#include <stdio.h>
int main () {
    int num;
    int d, rev=0, sum=0;
    printf("Enter the number: ");
    scanf("%d",&num);
    while (num>0) {
        d= num%10;
        num/=10;
        sum+=d;
        rev = (rev*10)+d;
    }
    printf("Reverse of number is %d\n Sum of digits is %d",
            rev, sum);
    return 0;
}
```



FIBONACCI NUMBERS

Aim: To find the first n fibonacci numbers

Algorithm

step 1 : start

step 2 : read the value of num

step 3 : initialize first = 0 , second = 1 and count = 2

step 4 : print first , second

step 5 : Repeat steps 6 to 9 while count < num

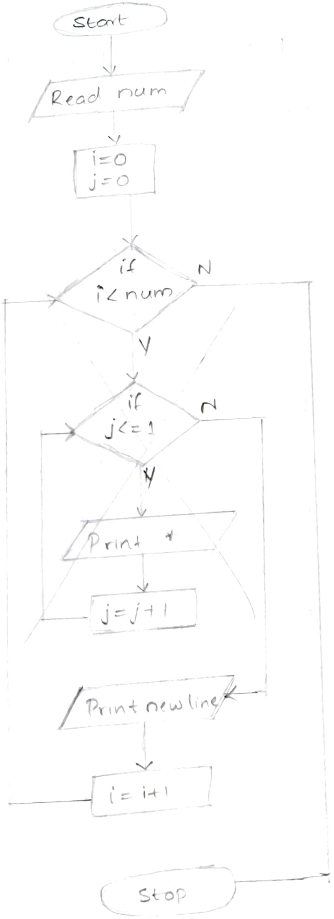
step 6 : set third = first + second

step 7 : print third

step 8 : set first = second , second = third

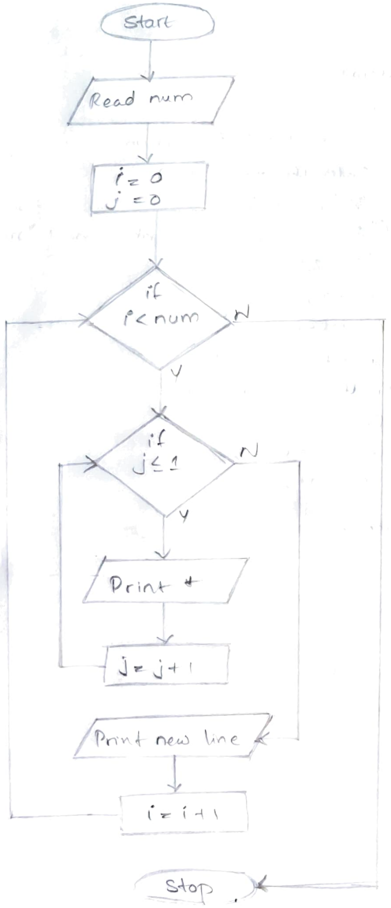
Step 9 : increment count by 1

Step 10 : stop.



Program

```
#include <stdio.h>
int main(){
    int first=0, second=1, third ,num ;
    printf("Enter the value : ");
    scanf(" %d" , &num);
    printf ("First %d fibonacci numbers are : ",num);
    printf ("%d, %d" , first , second );
    int count=2 ;
    while (count<num ) {
        third = first +second ;
        printf(", %d ",third );
        first = second;
        second = third ;
        count++;
    }
    return 0;
}
```



3

PYRAMID USING ASTERISK

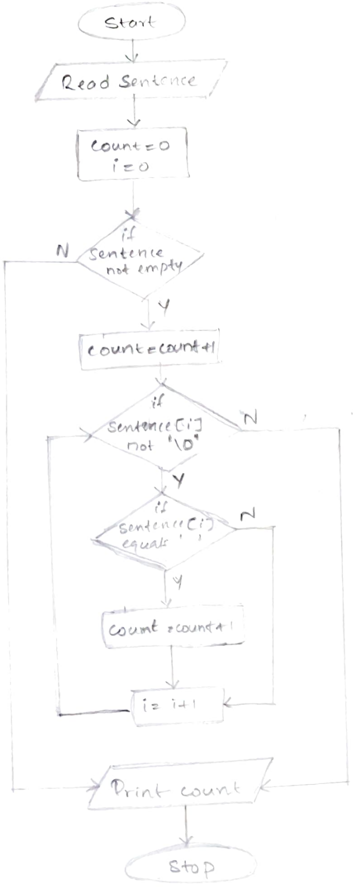
Aim : To create a pyramid using asterisk (*) in C

Algorithm

```
Step 1 : start
Step 2 : declare i,j, num
Step 3 : read number num
Step 4 : initialize i=0 , j=0
Step 5 : repeat steps 6 to 10 while i< num
Step 6 : repeat steps 7,8 while j<=1
Step 7 : print *
Step 8 : increment j by 1
Step 9 : print new line
Step 10 : increment i by 1
Step 11 : stop
```

Program

```
#include <stdio.h>
int main () {
    int num;
    printf("Enter the number : ");
    scanf ( "%d", &num);
    int i =0;
    while (i<num) {
        int j=0;
        while(j<=i ){
            printf(" * ");
            j++;
        }
        printf('\n');
        i++;
    }
    return 0;
}
```



COUNTING THE NUMBER OF WORDS IN A SENTENCE

Aim : To count the number of words in a given sentence using C

Algorithm

Step 1 : Start

step 2 : declare sentence[100] ,count=0, i=0

step 3 : read sentence

Step 4 : repeat steps 5 to 9 if sentence is not empty

Step 5 : increment count by 1

Step 6 : repeat steps 7, 8 until the end of the sentence

Step 7 : if sentence[i] equals ' ', increment count by 1

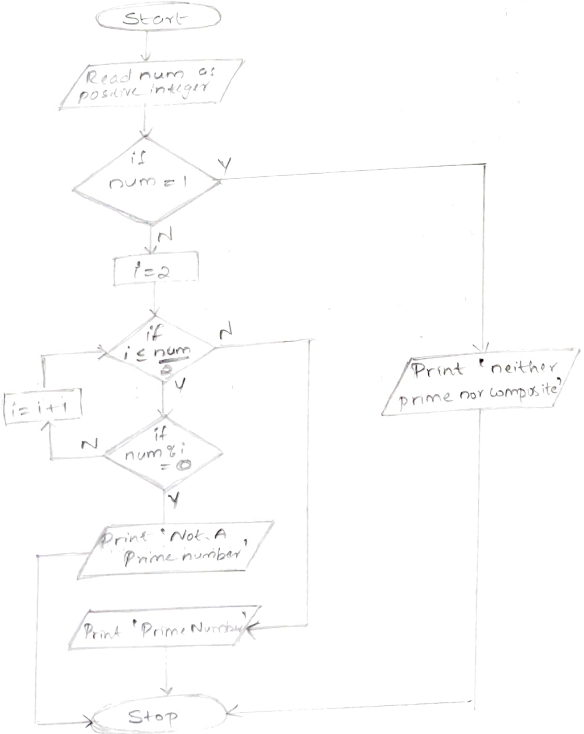
Step 8 : increment i by 1

Step 9 : Print count as number of words

Step 10 : stop.

Program

```
#include <stdio.h>
#include <string.h>
int main() {
      char sentence[100];
      int count = 0;
      printf("Enter the sentence : ");
       gets(sentence);
       if (sentence[0] != NULL) {
            count ++ ;
             for (int i=0; sentence[i] != '\0'; i++ ){
                   if (sentence[i] == ' ')
                         count ++ ;
             }
        }  printf("%d", count);
      return 0;
}
```


Start
Read num as positive integer
if num = 1
Y
N
i=2
if i ≤ num/2
N
Y
i = i+1
N
if num % i = 0
Y
Print 'Not A Prime number'
Print 'Prime Number'
Print 'neither prime nor composite'
Stop

PRIME NUMBER

Aim : To cheek if a number entered is prime or not in C

Algorithm

```
Step 1 : Start
Step 2 : read positive integer as num
step 3 : if num is 1 , Print "neither prime nor composite"
Step 4 : else , repeat steps 5 to 10
Step 5 : initialize i=2
step 6 : repeat steps 7 to 9 while i<=num/2
step 7 : repeat steps 8,9 while num mod i is zero
Step 8 : print "not a prime number"
Step 9 : goto step 11
Step 10 : print " prime number
Step 11 : Stop
```

Program

```
#include<stdio.h>
int main() {
    int num;
    printf("Enter a positive integer: ");
     scanf("%d",&num);
    if (num==1) {
        printf(" 1 is neither prime nor composite");
    }
    else {
        for (int i=2 ; i<=num/2 ; i++ ) {
            if (num%i ==0){
                printf (" %d is not a prime number", num);
                exit(0);
            }
        }
        printf(" %d is a prime number", num);
    }
    return 0;
}
```