

Problem Statement 1# Write a program to add first seven terms of the following series using a for loop:

$$1/1! + 2/2! + 3/3! + 4/4! + \dots n/n!$$

- Two Operations being carried out:
- Sum of numbers
- Finding factorial

Write a program to add first seven terms of the following series using a for loop:

$$5 + 7 + 8 + 9 + 6 + 13 + 21$$

Sum of n numbers

- Step1 : Input n, num, sum=0
- Step 2: Repeat steps 3 to 5 //for(i=1;i<n; i++)
- Step 3: Enter the number num
- Step 4: sum =sum +num
- Step 5: increment i by 1 and go to step 3
- Step 6 print sum
- Step 7: End

find the factorial value of any number

- Algorithm

Step 1: Enter the number **num** whose factorial u want to calculate

Step 2: count=num

Step 3: product=1

Step 4: Repeat steps while count >= 1

Step 5: product = product * count

Step 6: count - - and go to step 4

Step 7: print product

Step 8: End

```
#include<stdio.h>
int main()
{
    long int product=1;
    int count, num;
    printf("Enter the number whose factorial you want to find");
    scanf("%d", &num);
    for(count=num; count>=1;count--)
    {
        product=product*count;
    }
    printf("factorial of %d is %ld", num, product);
}
```

```
6! = 6*5*4*3*2*1
= product=1 , count =6
Product = 1*6 =6
Product = 6*5 =30
Product = 30*4 = 120
Product = 120*3 =360
Product = 360*2= 720
Product = 720*1= 720
```

$$1/1! + 2/2! + 3/3! + 4/4! + \dots n/n!$$

Algorithm

Step 1: Sum =0.0; num=1;

Step 2: Repeat steps 3 to 6 while num <=7 (i.e for(num=1;num<=7 ;num++))

Step 3: Find factorial of num and store it in product

Step 4: x = num/product;

Step 5: sum = sum + x

Step 6: increment num and go to step 2

Step 7: End

$$1/1! + 2/2! + 3/3! + 4/4! + \dots n/n!$$

Algorithm

Step 1: Sum =0.0; num=1;

Step 2: Repeat steps 3 to 10 while num <=7 (i.e for(num=1;num<=7 ;num++))

Step 3: count=num

Step 4: product=1

Step 5: Repeat steps while count >=1

Step 6: product = product * count

Step 7: count - - and go to step 5

//Calculates Factorial

Step 8: x = num/product;

Step 9: sum = sum + x

Step 10: increment num and go to step 2

Step 11: Print Sum

Step 12: End

Tips:

1. Declare product as long int
2. Declare sum and x as float

Program

```
#include<stdio.h>
int main()
{
    long int product;
    int count, num;
    float sum=0.0, x;
    for(num=1; num<=7; num++)
    {
        product=1;
        count=num;
        for(count=num; count>=1; count--)
        {
            product=product*count;
        }
        x = (float)num/product;
        sum+=x;
    }
    printf("Sum of first 7 no's is %f", sum);
    return 0;
}
```

Factorial
calculation

Problem Statement 2

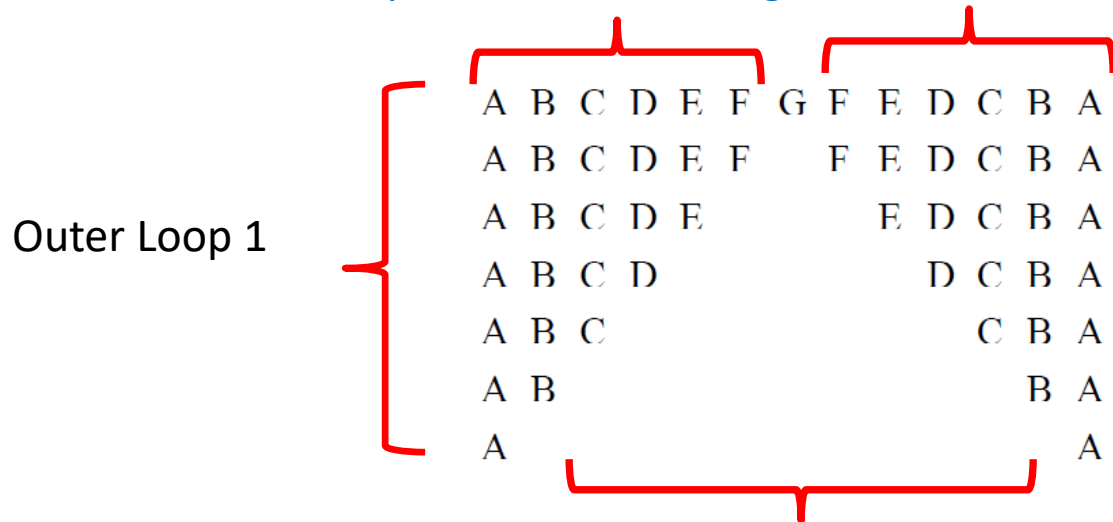
- How many loops are being used ?

```
A B C D E F G F E D C B A
A B C D E F   F E D C B A
A B C D E     E D C B A
A B C D       D C B A
A B C         C B A
A B           B A
A             A
```


- Four loops are being used

Inner Loop 1 for incrementing

Inner Loop 3 for decrementing



Inner Loop 2 for printing spaces

- Outer loop 1

```
for(i=1;i<=7;i++)
```

- Inner loop 1

```
for(ch = 'A';ch<= 'G';ch++)
```

```
{
```

```
printf("%c", ch)
```

```
}
```

```
ch--;
```

```
A B C D E F G F E D C B A
A B C D E F   F E D C B A
A B C D E     E D C B A
A B C D       D C B A
A B C         C B A
A B           B A
A             A
```

Outer loop and Inner Loop 1

```
#include<stdio.h>
int main()
{
    char ch = 'A', x = 'G';
    int i,j;
    for(i=0;i<=6;i++)
    {
        for(ch= 'A'; ch<=x;ch++)
        {
            printf("%c ", ch);
        }
        x= x-1;
        printf("\n");
    }
}
```

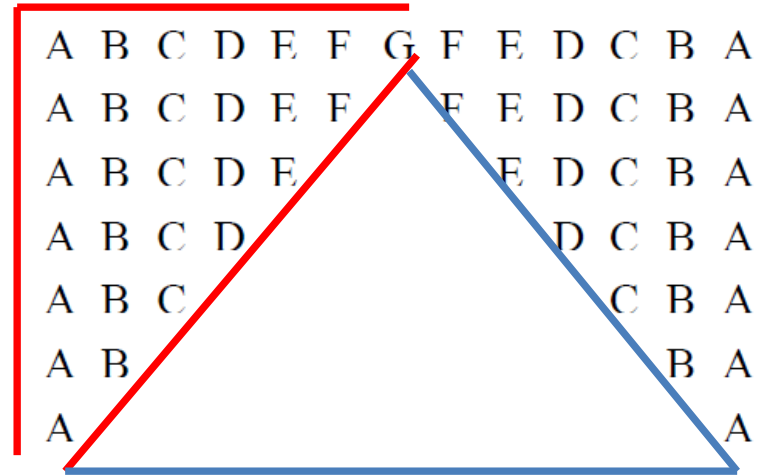
A	B	C	D	E	F	G	F	E	D	C	B	A
A	B	C	D	E	F	F	E	D	C	B	A	
A	B	C	D	E		E	D	C	B	A		
A	B	C	D			D	C	B	A			
A	B	C				C	B	A				
A	B					B	A					
A						A						

Outer loop and Inner Loop 2

```
#include<stdio.h>
int main()
{
    char ch = 'A', x = 'G';
    int i,j;
    for(i=0;i<=6;i++)
    {
        for(ch= 'A'; ch<=x;ch++)
        {
            printf("%c ", ch);

        }

        if(i!=0)
        {
            for(j=1;j<= 2*i-1;j++)
            {
                printf(" ");
            }
        }
    }
}
```



Outer loop	Inner Loop 2
------------	--------------

Iteration 0	0 space
-------------	---------

Iteration 1	1 space
-------------	---------

Iteration 2	3 space
-------------	---------

Iteration 3	5 space
-------------	---------

Iteration 4	7 space
-------------	---------

Iteration 5	9 space
-------------	---------

Iteration 6	11 space
-------------	----------

j=1 ;j<= 2*i-1; j++

Outer loop and Inner Loop 3

```
#include<stdio.h>

int main()
{
    char ch = 'A', x = 'G';
    int i,j;
    for(i=0;i<=6;i++)
    {
        for(ch= 'A'; ch<=x;ch++)
        {
            printf("%c ", ch);

        }

        if(i!=0)
        {
            for(j=1;j<= 2*i-1;j++)
                printf(" ");

            if( i==0)
                x=x-1;
            for(ch= x; ch>='A';ch--)
            {
                printf("%c ", ch);
            }
            if(i>0)
                x= x-1;
            printf("\n");
        }
    }
}
```

```

#include<stdio.h>
int main()
{
    char ch = 'A', x = 'G';
    int i,j;
    for(i=0;i<=6;i++)
    {
        for(ch= 'A'; ch<=x;ch++)
        {
            printf("%c ", ch);
        }

        if(i!=0)
        {
            for(j=1;j<= 2*i-1;j++)
            {
                printf(" ");
            }

            if( i==0)
            {
                x=x-1;
            }
            for(ch= x; ch>='A';ch--)
            {
                printf("%c ", ch);
            }
            if(i>0)
            {
                x= x-1;
                printf("\n");
            }
        }
    }
}

```

```

A B C D E F G F E D C B A
A B C D E F      F E D C B A
A B C D E          E D C B A
A B C D              D C B A
A B C                  C B A
A B                      B A
A                          A

```

Inner Loop 1

Inner Loop 2

Inner Loop 3

Why??

Why??

Why??

Outer Loop 1

```

#include<stdio.h>
int main()
{
    char ch = 'A', x = 'G';
    int i,j;
    for(i=0;i<=6;i++)
    {
        for(ch= 'A'; ch<=x;ch++)
        {
            printf("%c ", ch);
        }

        if(i!=0)
        {
            for(j=1;j<= 2*i-1;j++)
            printf(" ");

            if( i==0)
            x=x-1;
            for(ch= x; ch>='A';ch--)
            {
                printf("%c ", ch);
            }
            if(i>0)
            x= x-1;
            printf("\n");
        }
    }
}

```

```

A B C D E F G F E D C B A
A B C D E F      F E D C B A
A B C D E          E D C B A
A B C D              D C B A
A B C                  C B A
A B                      B A
A                          A

```

Inner Loop 1

$i \neq 0$ is set because spaces are getting printed for second iteration. In first iteration when $i=0$, no space is printed

Inner Loop 2

Only in the first step a decremented value is printed after G got printed so this has to be handled as a special case

Inner Loop 3

If condition is put otherwise after the first iteration x would have got decremented twice

Outer Loop 1

Problem Statement 3

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

Write a program to determine whether a number is prime or not. A prime number is one, which is divisible only by 1 or itself.

Step 1: Enter the num

Step 2: set $i=2$;

Step 3: repeat steps 4 to 5 while $i \leq \text{num}-1$

Step 4: if($\text{num}\%i==0$)

 break and exit the loop

 else

$i=i+1$;

Step 5: go to step 3

Step 6: if($i==\text{num}$)

 Print num is a prime number

Step 7: End

Write a program to determine whether a number is prime or not. A prime number is one, which is divisible only by 1 or itself.

```
main( )
{
    int  num, i ;

    printf ( "Enter a number " ) ;
    scanf ( "%d", &num ) ;

    i = 2 ;
    while ( i <= num - 1 )
    {
        if ( num % i == 0 )
        {
            printf ( "Not a prime number" ) ;
            break ;
        }
        i++ ;
    }

    if ( i == num )
        printf ( "Prime number" ) ;
}
```

Problem Statement 4

- Write a program to generate all prime numbers between 1 to 1000

Write a program to generate all prime numbers between
1 to 1000

Algorithm

Step 1 $j=1$;

Step 2: Repeat steps 3 to 6 while $j \leq 1000$ (i.e. for($j=1; j \leq 1000 ; j++$))

Step 3: $num=j$

Step 4: Check whether num is prime or not

Step 5: Print num if prime

Step 6: increment j and go to step 2

Step 7: End

Algorithm

Step 1 $j=1$;

Step 2: Repeat steps 3 to 6 while $j \leq 1000$ (i.e. for($j=1; j \leq 1000 ; j++$))

Step 3: $num=j$

Step 4: set $i=2$;

Step 5: repeat steps 6 to 7 while $i \leq num-1$

Step 6: if($num \% i == 0$)

 break and exit the loop

 else

$i=i+1$;


Step 7: go to step 5

Step 8: if($i==num$)

 Print num is a prime number

Step 9: increment j and go to step 2

Step 10: End



Inner loop checking
prime number