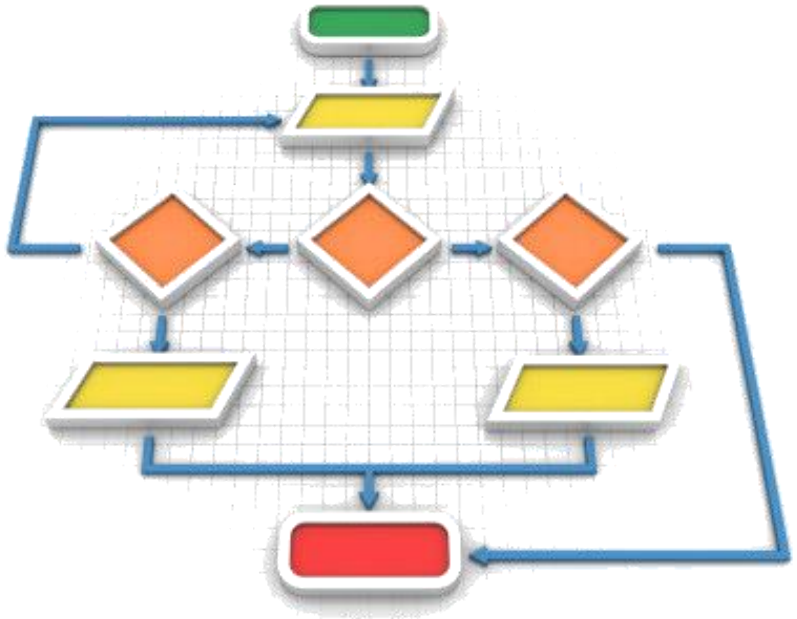
S9_2

Control Structures-
Branching & Looping



Session Objectives

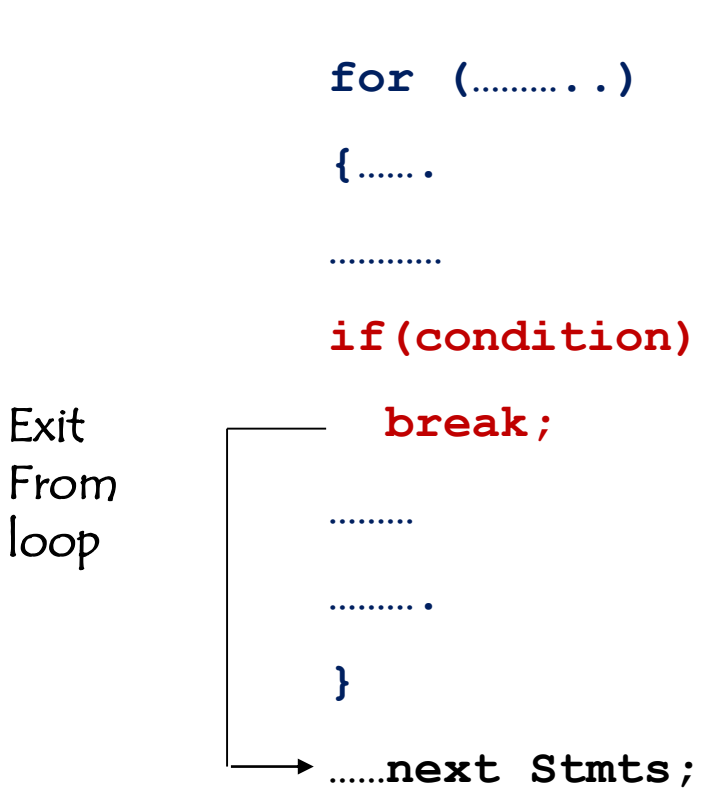
- To learn and appreciate the following concepts
  - The `break` with `for` statement
  - The `continue` with `for` statement
  - Problems on Control Structures

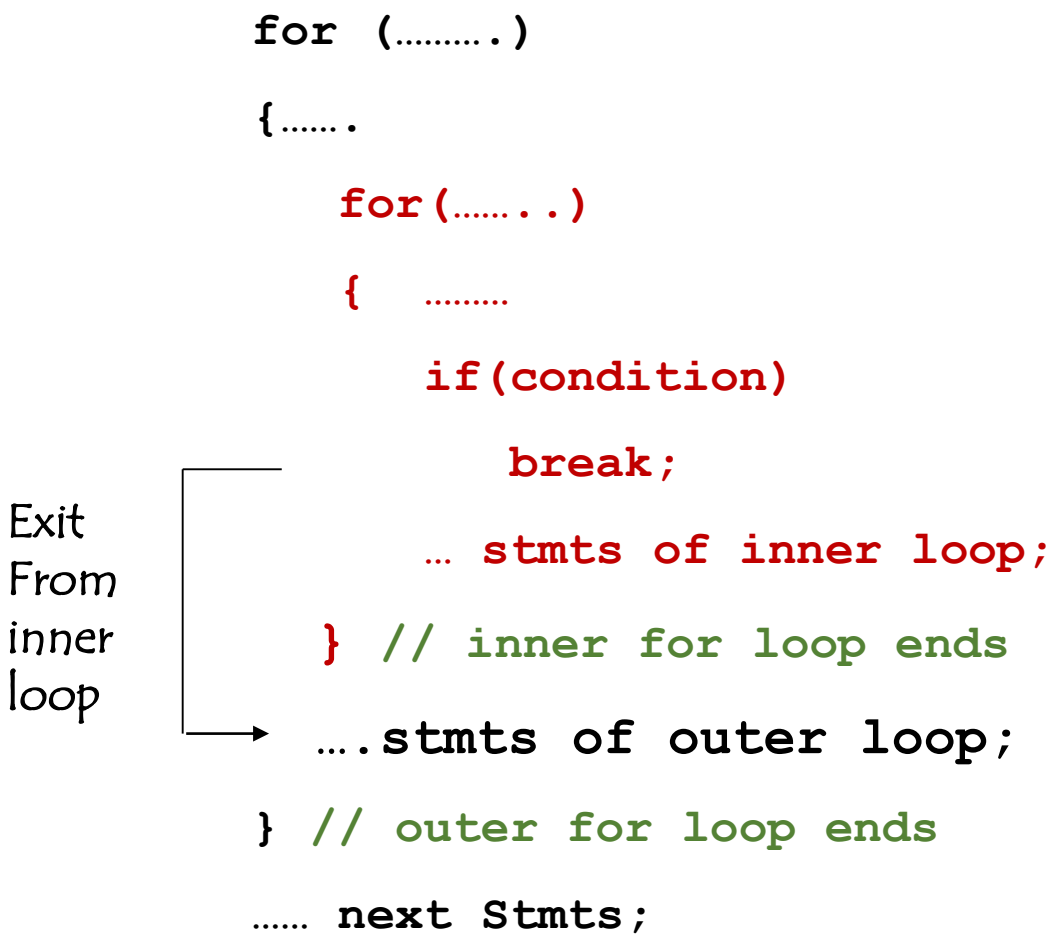
Session Outcomes

At the end of session student will be able to learn and understand

- The `break` with `for` statement
- The `continue` with `for` statement
- Problems on Control Structures

Exiting a loop with `break` statement in `for` statement

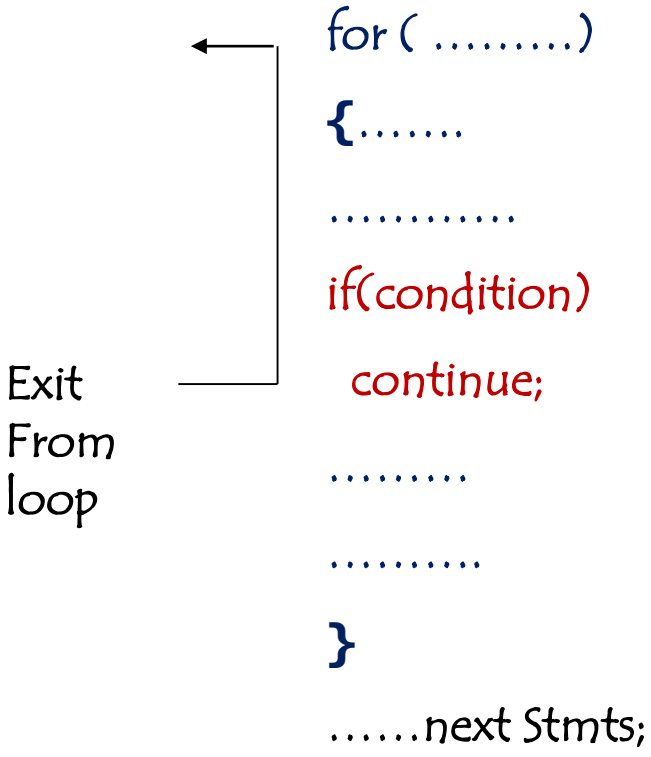


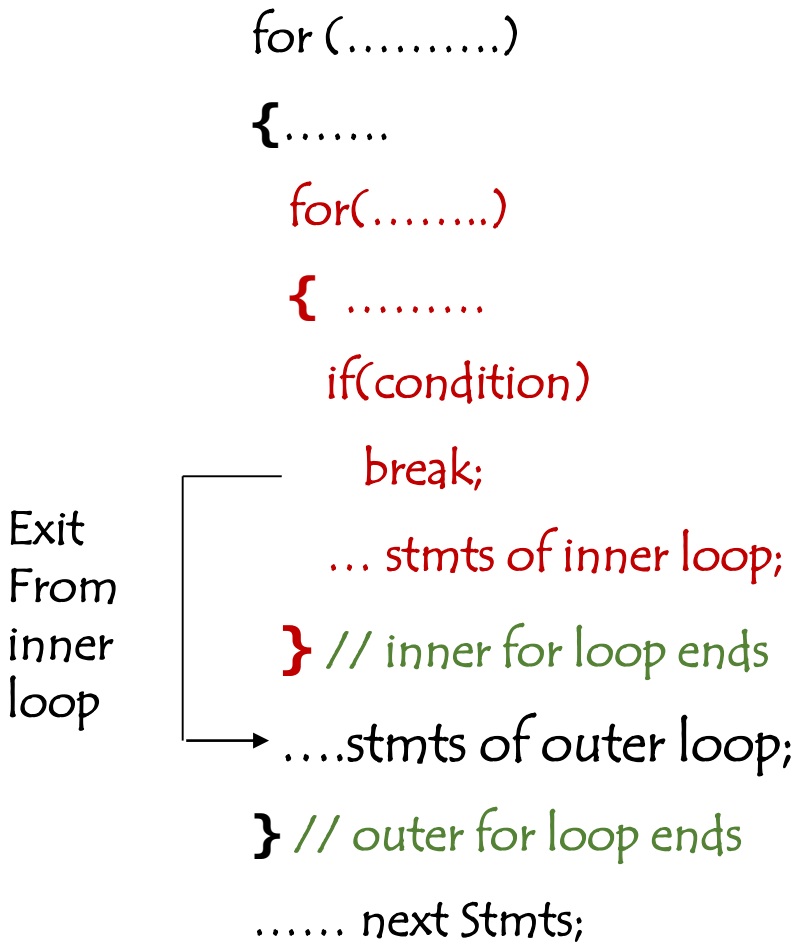


Check whether given number is prime or not - example

```
int  j, prime=1;
scanf("%d",&N);
 for( j=2; j<N; j++ )
      {
           if( (N % j) == 0)
           {
              prime=0;
              break; /* break out of for loop */
           }
       }
        if (prime == 1)
     printf("%d is a prime no",N);
     else
     printf("%d is a not a prime no",N);
```

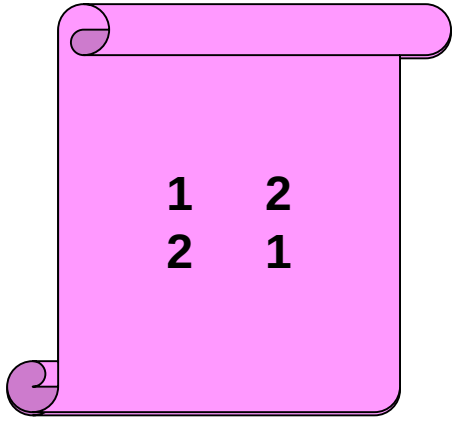
Skipping a part of loop – `continue` in `for` statement





Skipping a part of loop – `continue` in `for` statement

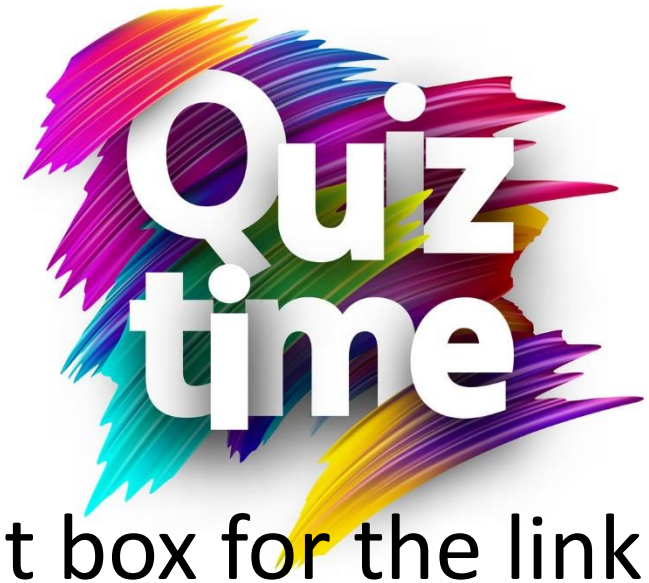
```
for ( i = 1 ; i <= 2 ; i++ )
{
    for ( j = 1 ; j <= 2 ; j++ )
    {
        if ( i == j )
            continue ;
        printf("\n %d\t %d\n",i, j);
    }
}
```



Generate prime numbers between given 2 limits

```
scanf("%d %d",&m, &n);

for( i=m; i<=n; i++) {
      int prime=1;
      for( j=2; j<i; j++ ) {
            if( i % j == 0)
                {
                  prime=0;
                  break; /* break out of inner loop */
                }
        }
       if (prime == 1) printf("%d\t",i);
}
```



Go to posts/chat box for the link to the question

submit your solution in next 2 minutes

The session will resume in 3 minutes

Problems on Control Structures

Factors of a Positive Integer

```
#include <stdio.h>
int main() {
    int num, i;
    printf("Enter a positive integer: ");
    scanf("%d", &num);
    printf("Factors of %d are: ", num);
    for (i = 1; i <= num; ++i) {
        if (num % i == 0) {
            printf("%d ", i);
        }
    }
    return 0;
}
```

For num = 20
OUTPUT
Factors of 20 are: 1 2 4 5 10 20

```
Enter a positive integer: 20
Factors of 20 are: 1 2 4 5 10 20
```

Armstrong no's for a given limit 'n'

Armstrong Number
Σ (cubes of digits)= num
$3^3 + 7^3 + 1^3 = 371$

```
scanf("%d",&lim);
printf("The armstrong numbers are:");
for(n=1;n<lim;n++){
   sum = 0; //initialized for each number
   num = n; //store it for comparison
   while(num>0) {
    dig = num%10; //extracting digits
    sum = sum+pow(dig,3); //sum of cube of digits
    num = num/10; //remaining digits for next iteration
   }
if(sum == n)
    printf("%d\n\t",n);
}
```

```
Enter the limit: 400
The armstrong numbers are:
 1
 153
 370
 371
```

Sine series for a given 'n' terms & angle 'x'

Sine series

$\sin(x) = x - x^3/3! + x^5/5! - \ldots x^n/n!$

```
# define PI 3.141592
scanf("%d %f",&n,&x);
 no=x;
 x=x*PI/180.0; // degrees to radians
 term=x; // first term value
 sum=x; //term stored in sum
  for (i=1;i<=n;i++) //compute & sum for second term onwards
   {
     term= term*(((-1)*x*x)/(2*i*(2*i+1)));
     sum+=term;
   }
printf("Library value of Sin(%.2f) = %.2f ", no, sin(x));
printf("\nSin (%.2f) = %.2f", no, sum);
```

```
Enter the values for number of terms(n) and angle (x):
 5
 90
Library value of Sin(90.00) = 1.00
Computed Sin (90.00) = 1.00
```

Tutorial Problems

1. Write a C program to count number of digits in any number
2. Write a C program to find last and first digit of any number
3. Write a C program to enter any number and print all its factors
4. Write a C program to find LCM of two numbers
5. Write a C program to convert Binary to Octal number

Session 9 Summary

- The `for` Statement
- Nested `for` Loops
- `for` Loop Variants
- The `break` with `for` statement
- The `continue` with `for` statement
- Problems on Control Structures