

# Nested Loops and Jumping Statements

**Gagan Deep**  
**Founder & Director**  
**Rozy Computech Services**  
Kurukshetra-136119, M- 9416011599  
Email – [rozygag@yahoo.com](mailto:rozygag@yahoo.com)  
Visit us at : [www.rozyph.com](http://www.rozyph.com)

## Nested Loops

- Loops, like if-else statements, can be nested, one within another.
- The inner and outer loops need not be generated by the same type of control structure.
- It is essential, however, that one loop be completely embedded within the other - there can be no overlap.
- Each loop must be controlled by different index.
- Moreover, nested control structure can involve both loops and if-else statement.
- Thus, a loop can be nested within an if-else statement, and an if-else statement can be nested within a loop.
- The nested structure may be complex as necessary, as determined by the program logic.

## Tables of 1-10

## Program 1-4

```
#include <stdio.h>
void main()
{ int i,j;
  for (i=1; i<=10; i++)
    for(j=1; j<=10; j++)
      printf("%d ", i*j);
```

Prints 1 2 3 ..10 2 4 6 ..20 .....10 20 30 .....100

```
#include <stdio.h>
void main()
{ int i=1,j;
  while(i<=10)
    { for (j=1; j<=10; j++)
      printf(" %d\t", i*j);
      i++; } }
```

```
#include <stdio.h>
void main()
{ int i=1,j=1;
  while(i<=10)
    { while(j<=10)
      { printf(" %d", i*j);
        j++;}
      i++;} }
```

```
#include <stdio.h>
void main()
{ int i,j =1;
  for (i=1; i<=10; i++)
    do
      { printf("%d\n", i*j);
        j++; }
    while(j<=10) ; }
```

## Program 5-6

Check Palindrome Number from 11 to 500

```
#include<stdio.h>
void main()
{int i, r, T, rn=0;
for ( i = 11; i<=500; i++) {
    T=i;
    while (T!=0) {
        r=T%10;  T=T/10 ;
        rn=rn*10+r; }
    if (rn==i)
        printf("%d is Palindrome Number", );
    else
        printf("%d is Non-Palindrome Number");
    } }
```

Check Palindrome Number from 11 to 500

```
#include<stdio.h>
void main()
{int n=11,r, T, rn=0;
while(n<=500) {
    T=n;
    while (T!=0) {
        r=T%10;  T=T/10 ;
        rn=rn*10+r; }
    if (rn==n)
        printf("%d is Palindrome Number", );
    } }
```


# Jumping Statements

- Jumping statements are also known as Loop Control Statements.
- Jumping statements are of different types
  - break
  - continue
  - goto
  - return
  - exit ()

## break Statements

- break statement simply terminates the loop and takes control out of the loop. Here explained break statement for for Loop

```
for(.....)
{
    .....
    .....
    if (condition)
        break;
    .....
    .....
}
```



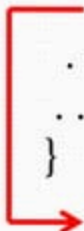
8. Print sum of infinite numbers.

```
#include <stdio.h>
void main ()
{ int a, sum=0;
  for( ; ; )
  { scanf("%d", &a);
    if (a==-999)
      break;
    sum=sum+a; }
  printf("The sum is %d",
    sum); }
```




## break statement for while & do While Loop

```
while(.....)
{
    .....
    .....
    if (condition)
        break;
    .....
    .....
}
```



```
.....
```

```
do
{
    .....
    .....
    if (condition)
        break;
    .....
    .....
}
```



```
while (.....);
.....
```

## Continue Statement

- Continue is used for skipping a part of loop for some condition.
- Continue causes the remaining code inside a loop block to be skipped and causes execution of jump to the top of loop block




```
for(.....)
{
    .....
    .....
    if (condition)
        continue;
    .....
    .....
}
```


```
12 Print 1-10 numbers except 3 and 7
#include <stdio.h>
void main ()
{
    int i;
    for(i=1 ; i<=10 ; i++)
    {
        if ((i==3)|| (i==7))
            continue;
        printf(" %d\t", i);
    } }
```



## continue statement for while & do-while Loops



```
while(.....)
{
    .....
    .....
    if (condition)
        continue;
    .....
    .....
}
```



```
do
{
    .....
    .....
    if (condition)
        continue;
    .....
    .....
}
while (.....);
```

- The goto statement is used to alter the normal sequence of program execution by transferring control to some other part of the program unconditionally/conditionally.
- In its general form, the goto statement is written as
  - goto label;
  - where the **label is an identifier** that is used to label the target statement to which the control is transferred.  
label : statement;
- Each labeled statement within the function must have a unique label, i.e., no two statement can have the same label.

## goto statement

goto label1

.....

.....

label2:

.....

.....

label1:

.....

.....

goto label2

## return statement and exit()

- **return** is an instruction of the language that returns from a function call.
- **exit** is a system call (not a language statement) that terminates the current process.

## Print Prime numbers in between 2-100 Prog. 16

```
#include <stdio.h>
void main ()
{
    int i, j;
    for(i=2; i<100; i++)
    {
        for(j=2; j <= (i/j); j++)
            if(!(i%j))
                break;
        // if factor found, not prime
        if(j > (i/j))
            printf ("%d is prime\n", i);
    }
}
```

## Examples of Pyramid Program 17

```
#include <stdio.h>

void main()
{
    int i,j,l;
    printf("Number of lines: ");
    scanf("%d",&l);
    for(i=1;i<=l;++i)
    {
        for(j=1;j<=i;++j)
            printf("* "); // printf("j ");
        printf("\n");
    }
}
```

```
*           1
**          12
***         123
****        1234
*****       12345
```

## Do Yourself

- Count even and digits of a number
- Sum of even and odd digits of a number
- Check for Armstrong Number
- Fibonacci Sequence
- Prime number when divide upto  $n/2$  and sqrt of  $n$





**THANKS!**

If you have any queries you can  
contact me at :

[rozygag@yahoo.com](mailto:rozygag@yahoo.com)

Visit us at : [www.rozyph.com](http://www.rozyph.com)