Loops

- Performing a task or a number of tasks for a known number of times.
  - while
  - do while
  - for

while Loop

- Syntax:
  ```
  while ( condition ){
          /* BODY OF WHILE LOOP is repeated */
  }
  ```

Description:

1. Condition is first checked:
   - So long the condition is TRUE then the body of the loop is executed and the loop repeats (goes back to step 1)
   - If condition is FALSE then the body of the while loop is NOT executed and loop is terminated (does not go back to step 1).

```
/* Example: Print integer numbers from 1 to 10 using while*/
#include <stdio.h>
#define         NUMBER      10

main( ){
        int     counter= 1;

      /*Print the numbers 1 through 10 */
        while(counter <= NUMBER){
                printf("Value of counter = %d\n", counter);
                counter = counter + 1;
        }
        printf("The value of counter after the while loop is over: %d\n", counter);
}
```

```
/* Read integer numbers from standard input and then add them up using while
loop. Also print how many integer numbers where entered */
#include <stdio.h>

main( ){
        int     number, sum=0;
        char    Repeat= 'Y';

      /* Get numbers until the user types 'n' or 'N'*/
        while( (Repeat != 'N') && (Repeat != 'n') ){
                printf("Pls enter an integer number,\n");
                printf ("And then 'Y' if there are more numbers,\n");
```

```
                printf (“Or ‘N’ if there is no other numbers.\n”);

                scanf(“%d %c”, &number, &Repeat);
                sum = sum + number;

                printf (“\n”);
        }       /* End of while loop. */
       /* Print sum of the numbers */
        printf(“The total sum of numbers is: %d.\n”, sum);
}
```

Q1) Do we know how many numbers the user may input before the program gets executed?

Q2) What happens if the user input any character other than Y or N? i.e., s, d, C, 4, 5?

/* Exercise: Read integer numbers from standard input and then add them up so long as the integer number is a positive number. Also count and print number of integer numbers.*/

```
#include <stdio.h>

main( ){
        int     number, cntr=1, sum=0;

        /* Get numbers until the user inputs a negative number*/
        puts(“Pls enter a POS. integer (to end input any NEG. number): “);
        scanf(“%d”, &number);

        while( number >= 0 ){
                sum = sum + number;
                cntr = cntr + 1;
                printf (“Pls enter a POS integer (any NEG ends loop): \n“);
                scanf(“%d”, &number);
                printf (“\n”);
        }
       /* Print sum of the entered numbers */
        printf(“%d positive integer numbers are entered.\n”, cntr);
        printf(“The total sum of inputted numbers is: %d.\n”, sum);
}
```

do while Loop

- Similar to while-loop
- However,
  - **while** tests the loop condition at the top of the loop; but
  - **do while** tests the condition at the end of the loop
    - Regardless of the condition, the body of the do while loop is executed at least once.
- Syntax:

```
do{
        statement;
        …
        …
}while (condition);
```

Note: So long as the condition is TRUE, the body of the loop will be repeated.

Example:

```
/* This program prints integer numbers from 0 to 10 using do while. */
#include <stdio.h>
#define         NUMBER      10
main( ){
        int     counter=0;

       /*Print numbers until number is greater than 10*/
        do{
                printf("counter is %d\n", counter);
                counter++;
        }while(counter <= NUMBER);

        printf("The value of the counter after the loop is over: %d.\n", counter);
}
```

Q) What is the value of counter after the loop is over?

```
/* Exercise: Read integer numbers and add them up. Also print the number of
integer numbers entered. DO THIS AS LONG AS THE NUMBER ENTERED IS
POSITIVE */
#include <stdio.h>

main( ){
        int     number=0, cntr=-1, sum=0;
        do{
                sum += number;
                cntr++;
                printf (“Pls enter a POS. integer (any NEG to end): \n“);
                scanf(“%d”, &number);
                printf (“\n”);
        } while( number >= 0 );
       /* Print sum of the numbers */
        printf(“%d positive integer numbers are entered.\n”, cntr);
        printf(“The total sum of the numbers is: %d.\n”, sum);
}
```

for Loop

- General Syntax:

```
                for(index; condition; change)
                {
                        /* BODY OF for LOOP */
                }
```

Note: If the body includes only ONE statement, no need to use the braces.

STEPS TO EXECUTE A FOR LOOP

1. index is initialized,
2. condition is checked:
   If the condition is TRUE, then the body of for loop is executed.
   Otherwise, if the condition is FALSE, then exit loop.
3. index is changed then go back to step 2.

Example:

```
/* This program prints to standard output integer numbers from 1 to 10*/
#include <stdio.h>
#define         MAX             10
main( ){
        int             counter;

        for(counter= 1; counter <= MAX; counter = counter + 1)
                printf("Counter is %d. \n", counter);

        printf(“The counter after the loop is finished: %d.\n”, counter);
}
```

AFTER EXECUTION (EITHER PROGRAM):

```
        Counter is 1.
        Counter is 2.
        …
        Counter is 9.
        Counter is 10.
        The counter after the loop is finished: 11
```

Q1: Can you justify the final value of the counter (11)?

Post-Increment: i.e.

```
count = count + 1;
count++;
```

Exercise:

```
/* Write a program that prints to standard output integer numbers from 1 to 10
using post-increment and for loop. */
#include <stdio.h>
#define     MAX         10
main( ){
        int         count;

        for(count = 1; count <= MAX; count++)
                printf("count = %d\n", count);
}
```

Example:

```
/* EMPTY LOOP or void loop*/
#include <stdio.h>
#define     MAX         100

main( ){
        int     count;
        for(count = 1; count <= MAX; count++);
}
```

Sum Algorithm

- Objective is to calculate the sum of integer numbers from 1 to 10.
- i.e.

  1+2+3+…+9+10
- One method:
  - Start with SUM = 0, then
  - Add 1, now SUM = 1; SUM = 0+1, then
  - Add 2, now SUM = 3, SUM = 0+1+2, then
  - …, then
  - Add 10, Finally SUM = 55, SUM = 0+1+2+…+10 = 55

- Another method:
  - Use a “for” loop, rather than writing lines and lines of codes.

Exercise:

```
/* Write a program that adds numbers from 1 through 10 using a for loop.*/
#include<stdio.h>
#define     MAX         10

main( ){
	int		cnt, SUM=0;

	/* Calculate sum of numbers 1 to 10 and print the result */
	for(cnt=1; cnt<= MAX; cnt++)
		SUM=SUM+cnt;

	printf("1+2+3+…+8+9+10= %d", SUM);
}
```

Factorial Algorithm

- Objective is to calculate the product of integer numbers from 1 to 9.
- Same as finding factorial of a number.
- i.e.,

  $n! = 1 \times 2 \times \ldots \times n\text{-}1 \times n$
- One method:
  - Start with PROD = 1, PROD = 1, then
  - Multiply by 2, now PROD = 12, PROD = 1x2, then
  - Multiply by 3, now PROD = 6, PROD = 1x2x3
  - …, then
  - Multiply by 9, now PROD = 3,628,800, PROD = 1x2x…x9
- Another method:
  - Use a “for” loop.

Exercise:

```
/* Write a program that calculates the factorial of an integer number using for
loop. The integer number is entered from standard input. */

#include<stdio.h>

main( ){
	int PROD=1;
	int cnt, number;

	/* Input the number from user */
	printf("Pls enter an integer number: ");
	scanf("%d", &number);

	/* Calculate the factorial using for loop, print the result */
	for(cnt=1; cnt<=number; cnt++)
		PROD=PROD*cnt;

	printf("%d is  %d", number, PROD);
}
```

Nested Loops

- A loop within another loop
- Most often has applications when dealing with matrices

Example:

```
/* MULTIPLE NUMBERS 1 THROUGH 10 BY 2, 10 TIMES
, PRINT NUMBERS IN A 10X10 MATRIX    */
#include <stdio.h>
#define         ROWS        10
#define         COLUMNS     10
main( ){
        int     counter = 0;
        int     i, j;

        for (i = 1; i <= ROWS; i++){
                for (j = 1; j <= COLUMNS; j++){
                        printf("%5d", i*j);
                        counter++;
                }
                puts("");
        }
        printf("These loops were executed for a total of %d.\n", counter);
}

/* Exercise: Read an integer and separate the digits. */
#include<stdio.h>

main( ){
        int     num, digit;

       /*Input number from the user*/
        printf("Pls enter an integer number: ");
        scanf("%d", &num);

        printf("digits in reverse order are: \n");

        /*Divided number by 10 until the remainder is 0*/
        while( num != 0){
                digit=num%10;
                num=num/10;
                printf("%d, ", digit);
        }
        puts("");
}
```