For loop structure
for Loop Structure

General form of a for loop:

```plaintext
for(expression1; expression2; expression3)
statement1; /* execute this statement */
```
for Loop Structure

General form of a for loop:

```plaintext
for(expression1; expression2; expression3)
  statement1;  /* execute this statement */
```

The control variable: i

- `expression1` initializes the variable controlling the loop
  
  ```plaintext
  i = 0;
  ```

- `expression2` is the condition for continuing the loop
  
  ```plaintext
  i <= 10;
  ```

- `expression3` step to increase or decrease the control variable
  
  ```plaintext
  i++/* same as i=i+1 */
  ```
for Loop Structure

General form of a for loop:

```
for(expression1; expression2; expression3)
    statement1; /* execute this statement */
```

- **expression1** *initializes* the variable controlling the loop
  
  ```
  i = 0;
  ```

- **expression2** is the **condition** for continuing the loop
  
  ```
  i <= 10;
  ```

- **expression3** **step: increments** the control variable
  
  ```
  i++ /* same as i=i+1 */
  ```

- Note that there is **NO** semicolon after **expression3**! or after the closing parenthesis
Compound for Loop Structure

- To execute more than one statement in the for loop, enclose them in curly braces {  }

```cpp
for(expression1; expression2; expression3)
{
    statement1;     /* execute this statement */
    statement2;     /* execute this statement */
    ...
}
```
for Loop Structure – Flow Chart

\[ \text{for} (\text{expression1}; \text{expression2}; \text{expression3}) \]

- **expression1**: Initializes the loop control variable:
  - ex. \( i = 0; \)
- **expression2**: Tests the loop control variable to see if it is time to quit looping:
  - ex. \( i < 10; \)
- **statement**: Increments the loop control variable:
  - ex. \( i++ \)
for statement

```
for (i = 0; i < 10; i++) {
    printf("count at %d\n");
}
```

Pseudo-code:

1. Evaluate **expression1**: initialize i to 0
2. Test **expression2**: is i < 10?
   - If true:
     - Execute **loop body**: `printf("...")`
     - Execute **expression3**: `i++`
     - Go to (2)
   - If false: exit loop
for (i = 1 ; i < 10; i = i+1)
{
    statement 1;
    statement 2;
    ...
}

for keyword
control variable name
final value of control variable
change of control variable
for ( i = 1 ; i < 10; i = i*2)
{
    statement 1;
    statement 2;
    ...
}

1. control variable name
2. final value of control variable
3. change of control variable
int k = 10;

for (i = 1; i < k; i = i + 1)
{
    statement 1;
    k *= 2;

}
Example (I): read and sum 10 numbers

```c
#include <stdio.h>
main()
{
    int i, a, sum;
    sum = 0;
    for (i = 0; i < 10; i++)
    {
        Body
    }
    printf("total is %d", sum);
}
```
Example (I): read and sum 10 numbers

```
#include <stdio.h>
main()
{
    int i, a,sum;
    sum = 0;
    for (i=0; i < 10; i++)
    {
        printf("enter number \n");
        scanf("%d", &a);
        sum = sum + a;
    }
    printf("total is %d", sum);
}
```
Example (II): largest of 10 numbers

```c
#include <stdio.h>
int main()
{
    int i, a, max;
    printf("enter number \n");
    scanf("%d", &a);
    max = a;
    for (i=1; i < 10; i++)
    {
        printf("enter number \n");
        scanf("%d", &a);
        if (a > max)
        {
            max = a;
        }
    }
    printf("largest is %d", max);
}
```
Example: largest of 10 numbers

```c
#include <stdio.h>
main()
{
    int i, a, max;
    printf("enter number \n");
    scanf("%d", &a);
    max = a;
    for (i=1; i< 10; i++)
    {
        printf("enter number \n");
        scanf("%d", &a);
        if (a > max)
            max = a;
    }
    printf("largest is %d", max);
}
```

Can we move this into inside of the for loop?
Example: largest of 10 numbers

```
#include <stdio.h>
main()
{
    int i, a, max;
    printf("enter number \n");
    scanf("%d", &a);
    max = ?;
    for (i=1; i < 10; i++)
    {
        printf("enter number \n");
        scanf("%d", &a);
        if (a > max)
            max = a;
    }
    printf("largest is %d", max);
}
```
Example: largest of 10 numbers

```c
#include <stdio.h>
#include <limits.h>

main()
{
    int i, a, max;
    printf("enter number \n");
    scanf("%d", &a);
    max = INT_MIN;
    for (i=1; i < 10; i++)
    {
        printf("enter number \n");
        scanf("%d", &a);
        if (a > max)
            max = a;
    }
    printf("largest is %d", max);
}
```