INTRODUCTION TO PROGRAMMING IN C - PART 3

So far
* Sequential construct
* Conditional construct

Today
* Iterative construct
Iterative constructs

* While

Flow diagram:

```
while (x <= 10) {
    printf("x=%d\n", x);
    x = x + 1;
}
```
* For

Flow diagram:

Example:

```c
for (x = 0; x <= 10; x=x+1)
    printf("x=%d\n", x);
```
*Equivalence of while and for:*

**While**

```
Condition
   True
     ↓
Statement
```

**For**

```
Initialization
      ↓
Condition
    True
      ↓
Statement
      ↓
Update
```

False

False
Example:

```c
x = 0;
while (x <= 10)
{
    printf("x=%d\n", x);
    x = x + 1;
}
```

for (x = 0; x <= 10; x=x+1)
```
    printf("x=%d\n", x);
```

Note: in practice, we use the for loop when we know how many iterations to do, and use while when this is undetermined.
Increment/decrement shortcuts:

```c
x++;  // Equivalent to x=x+1; */
x--;  // Equivalent to x=x-1; */
```

Break and continue

```c
break;  // Will cause loop to terminate */
continue;  // Skip rest of code in loop and start executing next iteration */
```
Example:

```c
int i=0;
while (i<10)
{
    i++;
    if (i==5)
        continue;
    printf("%d\n",i);
}
```

```c
int i=0;
while (i<10)
{
    i++;
    if (i==5)
        break;
    printf("%d\n",i);
}
```
Examples

*Problem: echo characters from the keyboard into display until '0' (sentinel) is typed

```c
#include <stdio.h>    /* needed for printf and scanf */

int main ()
{
    char inchar;
    scanf("%c", &inchar);
    while (inchar != '0')
    {
        printf("%c\n", inchar);
        scanf("%c", &inchar);
    }
    return 0;
}
```

*Problem: add the first n positive integer numbers and print result to the display, where n is chosen by user
Key insight: \( \text{sum} = 0 + 1 + 2 + \ldots + n \)

```c
#include <stdio.h>  /* needed for printf and scanf */

int main()
{
    int n, i, sum=0;

    printf("Choose n: ");
    scanf("%d", &n);

    for(i=1; i<=n; i++)
    {
        sum+=i;  /* sum = sum+i; */
    }
    printf("The sum of the first %d positive integers is %d\n", n, sum);
    return 0;
}
```