while Repetition Statement

A repetition statement (also called a looping statement or a loop) allows the programmer to specify that a program should repeat an action while some condition remains true.

The statement contained in the while repetition statement constitutes the body of the while, which can be a single statement or a block.

Eventually, the condition will become false (or will be 0). At this point, the repetition terminates, and the first statement after the repetition statement executes.
As an example of C++'s `while` repetition statement, consider a program segment designed to find the first power of 3 larger than 100. Suppose the integer variable `product` has been initialized to 3. When the following `while` repetition statement finishes executing, `product` contains the result:
When the while statement begins execution, the value of product is 3.

Each repetition of the while statement multiplies product by 3, so product takes on the values 9, 27, 81 and 243 successively.

When product becomes 243, the while statement condition product <= 100 becomes false. This terminates the repetition, so the final value of product is 243. At this point, program execution continues with the next statement after the while statement.
Example: Find the first power of 3 larger than 100.

```c
#include "stdio.h"
#include "conio.h"
int main()
{
    int product = 3;
    while (product <= 100)
    {
        product = product * 3;
    }
    printf("%d", product);
    getch();
    return 0;
}
```

Example: Find the sum of integers 1 to 1000

```c
#include "stdio.h"
#include "conio.h"
int main()
{
    long int top = 0;
    int i = 1;
    while (i<=1000) {
        top = top + i;
        i = i + 1;
    }
    printf("%ld", top);
    getch();
    return 0;
}
```
Example  Find the factorial of 10

```c
#include"stdio.h"
#include"conio.h"
int main()
{
    float f=1;
    int i=1,N=10;
    
    while ( i<=N ) {
        f=f*i;
        i=i+1;
    }
    printf("%.0f", f);
    getch();
    return 0;
}
```

Example  Find the summation of N numbers given by user

Algorithm of this problem as follows

S.1. \( i = 1, sum = 0 \)
S.2. take number
S.3. \( sum = sum + number \)
S.4. \( i = i + 1 \)
S.5. if \( i \leq N \) then go S.2.
S.6. show \( sum \)
Find the summation of numbers given by user until the user give up.

```c
#include "stdio.h"
#include "conio.h"
int main()
{
    int sum=0;
    int i=1,N,number;
    printf("how many numbers do you have ");
    scanf("%d", &N);

    while (i<=N) {
        printf("give an integer ");
        scanf("%d", &number);
        sum = sum + number;
        i=i+1;
    }

    printf("%d", sum);
    getch();
    return 0;
}
```
Euclidian Algorithm

S.1. take numbers, \( N_1, N_2 \)
S.2. if \( N_1 > N_2 \) then interchange \( N_1 \) and \( N_2 \)
S.3. if \( N_1 \leq 0 \) go S.6.
S.4. \( N_2 = N_2 - N_1 \)
S.6. show \( N_2 \)

```c
#include "stdio.h"
#include "conio.h"
int main()
{
    int N1,N2,temp;
    printf("give numbers ");
    scanf("%d,%d", &N1,&N2);
    while (N1>0) {
        if (N1>N2) { //interchanging
            temp=N1;
            N1=N2;
            N2=temp;
        }
        N2=N2-N1;
    }
    printf("%d", N2);
    getch();
    return 0;
}
```