# Additional Examples of While Loops

# Example 1

- **Objective:** To read *N* integers from the user one by one, and output the largest number among the input values.
  - The value of N will be read from the user.
  - The N numbers are separated by whitespace characters.

# Example 1: Solution

- It can be identified as:
  - Aggregate data to generate a certain statistic
- Plan:
  - Use a variable to remember the largest value. Let's name this variable **`max`**.
  - Read the input values one by one.
  - For each input value, if its value is larger than **`max`**, replace the value of **`max`** by this input value.
  - How do we choose an initial value for **`max`** so that its value will get updated when we encounter a larger input?
    - Approach 1: Select a very small value
    - Approach 2: Use the 1st input value

# Example 1: Solution (Approach #1)

```
1   #include <stdio.h>
2   #include <limits.h>
3
4   int main(void) {
5       int max,            // To remember the largest input value
6           input,          // To hold the input value temporarily
7           N,              // # of input to be read from the user
8           i;
9
10      printf("N = ? ");
11      scanf("%d", &N);
12
13      max = INT_MIN;      // Let "max" be a very small value first,
14                          // we will update its value when we
15                          // encounter a larger input.
16
17
```

Must add "`#include <limits.h>`" to use **`INT_MIN`** (a predefined named constant that represents the smallest possible value of type `int`)

# Example 1: Solution (Approach #1)

```
18      i = 0;
19      while (i < N) {
20          scanf("%d", &input);
21          if (input > max)
22              max = input;
23          i++;
24      }
25      printf("The largest number is %d.\n", max);
26      return 0;
27  }
```

```
N = ? 5
6
2
-10
99
11
The largest number is 99.
```

# Example 1: Solution (Approach #2)

```
1   #include <stdio.h>
2
3   int main(void) {
4       int max,            // To remember the largest input value
5           input,          // To hold the input value temporarily
6           N,              // # of input to be read from the user
7           i;
8
9       printf("N = ? ");
10      scanf("%d", &N);
11
12      scanf("%d", &input);
13      max = input;        // So far we have seen only one input,
14                          // so let it be the largest number.
15                          // We will update "max" when we
16                          // encounter a larger input.
17
```

# Example 1: Solution (Approach #2)

```
18      i = 0;
19      while (i < N-1) { // Process the remaining N-1 input values
20          scanf("%d", &input);
21          if (input > max)
22              max = input;
23          i++;
24      }
25      printf("The largest number is %d.\n", max);
26      return 0;
27  }
```

```
N = ? 5
6
2
-10
99
11
The largest number is 99.
```

# Example 2

- **Objective:** To compute the sum of digits of a positive integer.

- This example aims to show how to use a while-loop to achieve the objective by repeatedly:
  - Extracting the last digit from the number,
  - Adding the value of the extracted digit to a variable, and
  - Removing the last digit from the number.

- What loop cases are involved?
  - Basic Case #2 - Indefinite Repetition (we won't know in prior how many digits there are)
  - Aggregate data (sum of the digits)

# Example 2: Solution

```
1   int num;
2
3   printf("num = ? ");
4   scanf("%d", &num);
5
6   while (num > 0) {           // Loop until no digit left
7       printf("%d\n ", num % 10);
8       num = num / 10;         // Remove the last digit from num
9
10  }
11
12
13
14
15
16
17
```

Do NOT try to tackle a complicated looping problem all at once. Do it **step by step**.

Let's start by extracting digits from a number

Do you see why we can accomplish that with the few lines above?

**Note**: We assume the input value is a non-negative integer.

# Example 2: Solution

```
1   int num;
2   int digitSum;                   // To store the sum of digits of "num"
3
4   printf("num = ? ");
5   scanf("%d", &num);
6
7   digitSum = 0;
8   while (num > 0) {               // Eventually num will become 0
9                                   // inside the loop
10
11      digitSum += num % 10;       // Add the last digit of num to digitSum
12      num = num / 10;             // Remove the last digit from num
13  }
14
15  printf("The sum of digits is %d.\n", digitSum);
16
17
```

After confirming that we are extracting digits correctly, we add logic to calculate the digit sum.
Inserted code are in yellow.

**Note**: We assume the input value is a non-negative integer.