

# Practicing while loops

## Table of Contents

- [1. README](#)
  - [1.1. Emacs tip](#)
- [2. The while statement](#)
  - [2.1. Simple example](#)
  - [2.2. Countdown example](#)
  - [2.3. Infinite loops](#)
  - [2.4. Printing table of square](#)
  - [2.5. Summing numbers](#)

## 1 README

- Practice workbook for while loops in C.
- Practice workbook for do..while loops in C.
- See `cc100/6_loops/README.org` in GitHub for script/solutions

### 1.1 Emacs tip

To **not** see the emphatic characters like `~` or `*` or `/` in the Org file text, run the following code chunk (or put the code in your `/.emacs` file): if successful, you should see `"t"` in the minibuffer.

```
(setq-default org-hide-emphasis-markers t)
```

If you don't put it in your `/.emacs` file, the command will only work for the current Emacs session.

## 2 The while statement

### 2.1 Simple example

Run the program with different values of `n`.

```
int i = 1, n = 10;
while ( i < n ) {
    i = i * 2;
    printf("%d < %d ?\n", i, n);
}
```

```
2 < 10 ?
4 < 10 ?
8 < 10 ?
16 < 10 ?
```

### 2.2 Countdown example

```
int i = 10;
while ( i > 0 ) {
    printf("T minus %d and counting\n", i);
    i--;
}
printf("i = %d\n", i);
```

```
T minus 10 and counting
T minus 9 and counting
T minus 8 and counting
T minus 7 and counting
T minus 6 and counting
T minus 5 and counting
T minus 4 and counting
T minus 3 and counting
T minus 2 and counting
T minus 1 and counting
i = 0
```

- What does this program do?
- What's the difference between `--i` and `i--`?
- Create a more concise version of the code:

```
int i = 10;
while ( i > 0 ) {
    printf("T minus %d and counting\n", i--);
    //      i--;
}
printf("i = %d\n", i);
```

```
T minus 10 and counting
T minus 9 and counting
T minus 8 and counting
T minus 7 and counting
T minus 6 and counting
T minus 5 and counting
T minus 4 and counting
T minus 3 and counting
T minus 2 and counting
T minus 1 and counting
i = 0
```

## 2.3 Infinite loops

```
while (1)
    puts("Still running...\n");
```

- Tangle the code, compile and run it on the CMD line
- `C-c C-v t => tangling`
- Why don't you see any output in Emacs?

## 2.4 Printing table of square

1. Declare integer variables `i` and `n`
2. Scan `n`
3. Initialize `i` to 1
4. Write a `while` statement that
  - prints `i` and `i * i`
  - increments `i` by one
5. Run the program (input file is already there)

```
int i, n;
scanf("%d", &n);
i = 1;
while ( i <= n ) {
    printf("%10d%10d\n", i, i * i);
    i++;
}
```

```
1      1
2      4
3      9
4     16
5     25
```

## 2.5 Summing numbers

- The program 1 below is only missing the `while` statement.

1. Use `n != 0` as the controlling expression
2. Inside the loop,
  - sum `um` with `sum += n`
  - scan the next number `n`
3. Run the code block with the preset input file
4. Tangle the file, compile and run it on the CMD line

```
int n, sum = 0;

printf("Enter integers (0 to terminate).\n");
scanf("%d", &n);
while ( n != 0 ) {
    sum += n; // sum up
    scanf("%d", &n); // scan n
}

printf("The sum is %d\n", sum);
```

```
Enter integers (0 to terminate).
The sum is 107
```

Created: 2022-04-25 Mon 09:19