cc-practice-while

Table of Contents

1 README

- Practice workbook for `while` loops in C.

2 The `while` statement

2.1 Simple example

Insert a print trace statement in the `while` loop to print the values of `i` and `n`, then run the program for different values of `n`.

—— SOLUTION ——

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

2.2 Countdown example

- Write a program that counts down from `i=10` and prints both the counter variable and the end value `n=10`.
- Use a compound operator `i--` for counting down
- Change the operator to `--i` and check if there's a difference

—— SOLUTION ——

```
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
```

- Create a more concise version of the code by pulling the counting statement into the `printf` statement.

  —— SOLUTION ——

```
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

- Let's produce an infinite loop!

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

- Tangle the code in 1, compile and run it on the command line
- Remember `C-c C-v t` to tangle
- Why don't you see any output in Emacs when you run this code?

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)

—— SOLUTION ——

```
echo "10" > ./src/square_input
cat ./src/square_input
```

```
int i, n;

printf("Enter number of rows:\n");
scanf("%d", &n);

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

```
Enter number of rows:
         1         1
         2         4
         3         9
         4        16
         5        25
         6        36
         7        49
         8        64
         9        81
        10       100
```

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 an input file that contains integers ending in 0, e.g. `5 10 15 0`. When the last element of the list is reached, the loop ends.

4. If you tangle the file, compile and run it on the command line, you can use the input file, too, like here, where `file` is the executable.

```
$ ./file < input
```

```c
int n, sum = 0;

printf("Enter integers (0 to terminate).\n");
scanf("%d", &n);
_______________ // sum up
_______________ // scan n
}

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

—— SOLUTION ——

```
echo 8 12 25 0 > "./src/sum_input"
cat "./src/sum_input"
```

```
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 45
```

Created: 2022-06-14 Tue 14:53