

# INTRODUCTION TO SCIENTIFIC AND ENGINEERING COMPUTATION

## Assignment 2

March 2018

**Objective:** Tracking the game score of a dart player.

### Description



The dart board consists of twenty numbered segments. Landing a dart in a segment scores the number value for that segment. For example, if the dart lands in one of the yellow areas between the number 17 and the center, the player scores 17 points.

The outer ring (just below the numeric values) is called the “double ring” and doubles the segment number. For example, if the dart lands in the green area right above the number 17, the player scores 34 points. The inner ring is called the “triple ring” and triples the segment value. For example, if the dart lands in the inner green area between the number 17 and the center, the player scores 51 points.

There are two circles at the center. The outer circle is called the “outer bull” and is worth 25 points. The inner circle is called the “inner bull” and is worth 50 points.

The player has to reach a target number of points, typically 301 or 501. The player starts with this point value and points get deduced with every throw. But to start scoring, the player has to land a dart in the double ring. Any dart landing in any other area before that happens doesn’t score any points.

Once a dart lands in the double ring, the points get deduced and subsequent throws will be scored regularly. For example, if the target value is 301 and the player throws a single 17 (or a triple 20, or a bull's eye, ...) the points remain 301. If the player throws a double 3, the points get down to 295. In the next throw, a triple 11 will reduce the points to 262, and so on.

To end the game, the player has to reach the exact point value 0, and again with a throw in the double ring. If the needed value is exceeded or the remaining point value is 1, the points don't get counted. For example, if the remaining points of a player is 14 and the player throws a 16, the points remain 14. Similarly, scoring a single 14 or a single 13 causes the points to remain 14. If the player throws a double 7, the game is ended.

## Assignment

Write a C program that will get the target value and a sequence of throws from the user, and tracks the point value along the game. The user will input each throw by giving its segment and ring. For example "17 D" will represent a double 17, "8 T" will represent a triple 8, and "15 S" will represent a single 15. The outer and inner bulls will be represented by an "O" and an "T", respectively, and in those cases, the segment value will be ignored.

1. Write a function that takes the segment and ring of a throw as parameters and returns the point value for the throw.
2. Write a function that takes the current points, and the segment and ring for a throw as parameters and returns the remaining points after that throw according to the rules described above. This should cover the starting and ending of the game.
3. Using these functions, write a `main` function that interacts with the user and prints out the score after every throw until the game is ended.
4. Make sure to properly document your functions (purpose, parameters, etc) as shown in the class and the slides.

Below is an example run:

```
Target: 101

Throw: 18 S
Points: 101
Throw: 7 T
Points: 101
Throw: 9 D
Points: 83
Throw: 1 0
```

```
Points: 58
Throw: 10 T
Points: 28
Throw: 15 D
Points: 28
Throw: 12 S
Points: 16
Throw: 16 S
Points: 16
Throw: 8 D
Points: 0
```

- If you need, you can add extra parameters to the functions described above.
- You don't need error checking on user input; you can assume that the user is typing the throws properly.

## Rules

- Your source code file has to have the name "assignment2.c".
- Your program will be compiled using the following command on a Linux system. If it cannot be compiled and linked using this command, it will not be graded (failed submission).

```
gcc -std=c99 -Wall -Werror assignment2.c -o assignment2
```

- Your program will be checked using an automatic checker. Therefore, make sure you print the messages exactly as given in the example runs.
- Do NOT use any construct that hasn't been covered in the course before this week, such as using arrays. Also use no C++ features such as cout and cin.
- Do NOT use any external functions except for printf and scanf.
- Make sure your coding style is proper and consistent. Use the clang-format tool if necessary. Don't use any variable names in a language other than English.
- This is an individual assignment. Collaboration in any form is NOT allowed. No "working together", no sharing code in any form including showing code to your classmates to give them ideas.
- All the code you submit must be your own. Don't copy/paste any piece of code from any resource including anything you've found on the Internet.

- The assignments will be checked for plagiarism using both automated tools and manual inspection. Any assignment involving plagiarism and/or infringement of intellectual property will be not be graded and is subject to further disciplinary actions.