

## CS 12 - Lab 2 Reading Input

In this lab you will be using the functions to read input. Be sure to include all necessary information at the top of the program and use good commenting and style throughout your program.

Write a program that plays a guessing game with users. Your program should allow the user to enter their name then choose between playing or letting a new user start playing.

The game is that a random number is chosen between 1 and 20 and the user has one shot at guessing the number. If they guess it they win \$1000. Your program should print whether they guessed correctly or not and should specify who was playing at the time (so someone else cannot claim the prize).

The option to allow a new user to start playing simply reads in the new user's name. When names are entered, they are not restricted to first name only.

An example run is shown below:

```
Enter name: John Doe
```

```
n - New player  
p - Play game  
q - Quit  
Enter choice: p
```

```
A random number has been drawn between 1 and 20.  
If you can guess it in one try, you win $1000  
Enter your guess: 12
```

```
Sorry, John Doe. The number was 17. Thanks for playing.
```

```
n - New player  
p - Play game  
q - Quit  
Enter choice: p
```

```
A random number has been drawn between 1 and 20.  
If you can guess it in one try, you win $1000  
Enter your guess: 8
```

```
Sorry, John Doe. The number was 9. Thanks for playing.
```

```
n - New player  
p - Play game  
q - Quit  
Enter choice: n
```

Enter your name: Kelsey Lick

n - New player  
p - Play game  
q - Quit  
Enter choice: p

A random number has been drawn between 1 and 20.  
If you can guess it in one try, you win \$1000  
Enter your guess: 4

Congratulations, Kelsey Lick. You have won.

n - New player  
p - Play game  
q - Quit  
Enter choice: q

The following is an explanation of generating random numbers. The `rand()` function returns an integer between 1 and `MAX_INT`. You must include the standard library header file `stdlib.h` in order to use this function. To generate random numbers within a specific range use the following formula,

```
num = (rand()%num_items) + base_num
```

where *num\_items* is the number of values in the range and *base\_num* is the lowest value. For example, to get values between 50 and 60 you would use the following expression,

```
int num = (rand()%11) + 50;
```

`rand()` actually generates a set of random numbers. Each run of the program will generate the same set of numbers in the same order, it only looks random within that run of the program.

A seed value can be used to generate a unique set of random numbers. Each seed value produces its own set. A seed value for the `rand()` function can be set by using the following function.

```
srand(seed);
```

The seed value is a long int. The default seed value is zero, which is why each program run produces the same set of random numbers. To generate a unique set for each run of the program, a random seed value should be chosen. A way of setting a pseudo-random seed value is to use the *time* at the beginning of the program execution. You must include the time header file `time.h` in order to use this function. An example is shown below.

```
srand(time(NULL));  
int num = (rand()%11) + 50;
```