

ECE 71 – Engineering Computations in C

Professor Kriehn – Fall 2016

Code Due By: Midnight on Tuesday, October 25, 2016

Writeup Due By: Class on Thursday, October 27, 2016

HOMEWORK #21 – Game of Craps

Write a program that simulates the game of craps, which is played with two dice. On the first roll, the player wins if the sum of the dice is 7 or 11. The player loses if the sum is 2, 3, or 12. Any other roll is called the “point”, and the game continues. On each subsequent roll, the player wins if he or she rolls the point again. The player loses by rolling a 7. Any other roll is ignored and the game continues. At the end of the each game, the program will ask the user whether or not to play again. When the user enters a response other than ‘y’ or ‘Y’, the program will display the number of wins and losses and then terminate.

Specifications:

Write the program using three functions: `main`, `roll_dice`, and `play_game`. Use the following function prototypes:

```
int roll_dice(void);
bool play_game(void);
```

`roll_dice` should generate two random numbers, each between 1 and 6, and return their sum. Use the `rand` function to generate random numbers.

`play_game` should play one craps game (calling `roll_dice` as many times as necessary to determine the outcome of each dice roll); it will return `TRUE` if the player wins and `FALSE` if the player loses. `play_game` is also responsible for displaying messages showing the results of the player’s dice rolls.

`main` will call `play_game` repeatedly, keeping track of the number of wins and losses, and displaying the “You win!” and “You lose!” messages. At the beginning `main`, after displaying a “Welcome to Craps” message, you should prompt the user to enter a random seed (to satisfy the grader program).

Finally, use the `exit` function and the `EXIT_SUCCESS` constant to exit your program (which are both defined in `<stdlib.h>`). From this point forward, each of your programs should use the `exit` function to terminate.

If you execute the program, the following information should be displayed:

```
~> hw21.o
Welcome to Craps
Please enter a random seed: 23
You rolled: 4
Your point is: 4
```

You rolled: 8
You rolled: 8
You rolled: 5
You rolled: 6
You rolled: 11
You rolled: 7
You lose!
Play again?: n

Wins: 0 Losses: 1
~>

~> hw21.o
Welcome to Craps
Please enter a random seed: 63
You rolled: 7
You win!
Play again? y
You rolled: 7
You win!
Play again? y
You rolled: 10
Your point is: 10
You rolled: 4
You rolled: 7
You lose!
Play again? n

Wins: 2 Losses: 1
~>

~> hw21.o
Welcome to Craps
Please enter a random seed: 152
You rolled: 5
Your point is: 5
You rolled: 3
You rolled: 5
You win!
Play again?: q

Wins: 1 Losses: 0
~>

HOMEWORK #22 – Statistical Functions

Write a program that prompts the user to enter a set of N double precision floating point numbers, scans them into a 1-D array, and calculates various statistical properties of the array of numbers, including the maximum, minimum, average, variance, standard deviation, standard error, and median.

Below are the mathematical definitions of interest. If you have an array $x[]$ with N elements:

Maximum – The maximum finds the maximum of the array elements $x[i]$, where i ranges from 0 to $N-1$.

Minimum – The minimum finds the minimum of the array elements $x[i]$, where i ranges from 0 to $N-1$.

Average – To find the average \bar{x} , sum the array elements $x[i]$ and divide by the number of elements N :

$$\bar{x} = \sum_{i=0}^{N-1} x[i] / N$$

Variance – To find the population variance σ^2 , sum the square of the difference between each array element $x[i]$ with the average \bar{x} , and divide by the number of elements N .

$$\sigma^2 = \sum_{i=0}^{N-1} (x[i] - \bar{x})^2 / N$$

Standard Deviation – To find the population standard deviation σ , take the square root of the variance:

$$\sigma = \sqrt{\sigma^2}$$

Standard Error – To find the standard error, divide the standard deviation by the square root of the number of elements in the array N :

$$SE = \frac{\sigma}{\sqrt{N}}$$

Media – The median finds the middle value of an array of elements. For an odd number of elements, the median is the central array element, after the array has been numerically sorted. For an even number of elements, the median is the average of the two central elements, after the array has been numerically sorted.

NOTE: For this function to work, you will also need a *sort()* function that sorts the elements in ascending order.

Specifications:

Use the following function prototypes:

```
double maximum(int n, double a[n]);
double minimum(int n, double a[n]);
double average(int n, double a[n]);
double variance(int n, double a[n]);
double std_dev(int n, double a[n]);
double std_err(int n, double a[n]);
double median(int n, double a[n]);
void quicksort(double a[], int low, int high);
int split(double a[], int low, int high);
```

Prompt the user to enter the number of elements for the array and use that information to create a variable length array. Then use a for loop to prompt the user to enter in a number for each element of the array, and scan in the appropriate numbers. Your program should then call the appropriate functions and print the results to the screen using a `%.2f` format. Finally, use the `exit` function to end the program. You must use functions to solve this problem! If you execute the program, the following information should be displayed:

```
~> hw22.o
```

```
Enter the number of elements in the array: 3
```

```
Enter element 0: 3
```

```
Enter element 1: -1
```

```
Enter element 2: 4
```

```
The maximum is: 4.00
```

```
The minimum is: -1.00
```

```
The average is: 2.00
```

```
The variance is: 4.67
```

```
The standard deviation is: 2.16
```

```
The standard error is: 1.25
```

```
The median is 3.00
```

```
~> hw22.o
```

```
Enter the number of elements in the array: 4
```

```
Enter element 0: -5.2
```

```
Enter element 1: 2.6
```

```
Enter element 2: 6.3
```

```
Enter element 3: 19.3
```

```
The maximum is: 19.30
```

```
The minimum is: -5.20
```

```
The average is: 5.75
```

```
The variance is: 78.43
```

```
The standard deviation is: 8.86
```

```
The standard error is: 4.43
```

The median is: 4.45

HINTS: For the maximum and minimum, initially set the maximum/minimum to the 0th element of the array. Then compare that value against the other elements and store find the true maximum/minimum. Use the math library and the `pow` function for the variance. Use the math library and the `sqrt` function for the standard deviation and the standard error. The median function must call the sort function.