

In-Class Assignment 2

1. Static Variable

Write a C program that uses a static variable to count the number of times a function is called. Display the count each time the function is invoked.

```
PS C:\Users\wes\github-repos\cs3150_fall2023\In-Class 2> .\wshakespear_part1.exe
Running the function 14836 times
The value of times_called is 14836
```

2. Dynamic Memory Allocation

Create a program that dynamically allocates memory for an integer array of user-defined size. Prompt the user for the array size and then input the elements. Finally, print the array in reverse order.

```
PS C:\Users\wes\github-repos\cs3150_fall2023\In-Class 2> .\wshakespear_part2.exe
How large should the array be? 10
array[ 0]: 100
array[ 1]: 101
array[ 2]: 102
array[ 3]: 103
array[ 4]: 104
array[ 5]: 105
array[ 6]: 106
array[ 7]: 107
array[ 8]: 108
array[ 9]: 109
Here is the data in the array:
    Index    0      Value:  100
    Index    1      Value:  101
    Index    2      Value:  102
    Index    3      Value:  103
    Index    4      Value:  104
    Index    5      Value:  105
    Index    6      Value:  106
    Index    7      Value:  107
    Index    8      Value:  108
    Index    9      Value:  109
```

3. Array Sum

Write a C function that calculates the sum of all elements in an integer array. Pass the array and its size as parameters to the function.

```
PS C:\Users\wes\github-repos\cs3150_fall2023\In-Class 2> .\wshakespear_part3.exe
Testing the sum function for the folowing data
{ 1, 2, 3, 4, 5, 6, 7, 8, 9, }
The sum is: 45

Testing the sum function for the folowing data
{ 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, }
The sum is: 232
```

4. Factorial Function

Implement a recursive function to calculate the factorial of a non-negative integer. The function should take an integer as input and return its factorial.

```
PS C:\Users\wes\github-repos\cs3150_fall2023\In-Class 2> .\wshakespear_part4.exe
To find the factorial enter a positive integer (0 to exit): 1
The factorial of 1 is 1
To find the factorial enter a positive integer (0 to exit): 3
The factorial of 3 is 6
To find the factorial enter a positive integer (0 to exit): 20
The factorial of 20 is 210
To find the factorial enter a positive integer (0 to exit): 100
The factorial of 100 is 5050
To find the factorial enter a positive integer (0 to exit): 0
```

In-Class Assignment 2

5. Array Manipulation

Write a program that allows the user to perform various operations on an integer array such as finding the maximum, minimum, and average values, and reversing the array.

```
Testing the functions with the following data
{ 1, 2, 3, 4, 5, 6, 7, 8, 9, }
The maximum is: 9
The minimum is: 1
The average is: 5
The reversed array is now:
{ 9, 8, 7, 6, 5, 4, 3, 2, 1, }

Testing the functions with the following data
{ 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, }
The maximum is: 89
The minimum is: 1
The average is: 21
The reversed array is now:
{ 89, 55, 34, 21, 13, 8, 5, 3, 2, 1, 1, }
```

6. Static vs. Dynamic Memory Allocation

Explain the difference between static and dynamic memory allocation in C, and provide an example illustrating when to use each.

Static:

- Static memory is used when the size of the information is known before compilation, and when the value stored in memory needs to stay allocated for the duration of the program.

```
1 // In-Class 2 - part 6 written by Weston Shakespear on 9/21/2023
2 // Explain the difference between static and dynamic memory allocation in C,
3 // and provide an example illustrating when to use each.
4
5 #include <stdio.h>
6
7 // Function for static memory demonstration
8 void log_message(char * message);
9
10 // Variable for static memory demonstration
11 int message_number = 1;
12
13 void main()
14 {
15     // Static memory demonstration
16     for (int i = 0; i < 10; i++)
17     {
18         log_message("Here is a sample log message");
19     }
20 }
21
22 void log_message(char * message)
23 {
24     printf("[%2d] - '%s'\n", message_number++, message);
25 }
```

```
PS C:\Users\wes\github-repos\cs3150_fall2023\In-Class 2> .\wshakespear_part6_static.exe
[ 1] - 'Here is a sample log message'
[ 2] - 'Here is a sample log message'
[ 3] - 'Here is a sample log message'
[ 4] - 'Here is a sample log message'
[ 5] - 'Here is a sample log message'
[ 6] - 'Here is a sample log message'
[ 7] - 'Here is a sample log message'
[ 8] - 'Here is a sample log message'
[ 9] - 'Here is a sample log message'
[10] - 'Here is a sample log message'
```

In-Class Assignment 2

Dynamic:

- Static memory is used when the size of the information is not known before compilation. Depending on how much information needs to be stored, the memory is allocated at runtime.

```
1 // In-Class 2 - part 6_dynamic written by Weston Shakespear on 9/22/2023
2 // Explain the difference between static and dynamic memory allocation in C,
3 // and provide an example illustrating when to use each.
4
5 #include <stdio.h>
6 #include <stdlib.h>
7
8 void main()
9 {
10     int items = 0;
11     printf("Please enter the total daily transactions: ");
12     scanf("%d", &items);
13
14     float * transactions = malloc(items * sizeof(float));
15
16     for (
17         int transaction_counter = 0;
18         transaction_counter < items;
19         transaction_counter++
20     )
21     {
22         float value = 0.0;
23         printf("Please enter transaction %4d: ", transaction_counter);
24         scanf("%f", &value);
25
26         *(transactions + transaction_counter) = value;
27     }
28
29     float balance = 0.0;
30     printf("Please enter the daily starting balance: ");
31     scanf("%f", &balance);
32
33     printf("\n");
34     printf("| # | BALANCE | WITHDRAW | DEPOSIT | \n");
35
36     for (
37         int transaction_counter = 0;
38         transaction_counter < items;
39         transaction_counter++
40     )
41     {
42         float transaction = *(transactions + transaction_counter);
43         balance += transaction;
44         printf("| %2d | %10.2f |", transaction_counter, balance);
45
46         if (transaction < 0.0)
47         {
48             printf(" %10.2f | | \n", transaction);
49         }
50         else
51         {
52             printf(" | %10.2f | \n", transaction);
53         }
54     }
55
56     free(transactions);
57 }
```

In-Class Assignment 2

```
PS C:\Users\wes\github-repos\cs3150_fall2023\In-Class 2> .\wshakespear_part6_dynamic.exe
Please enter the total daily transactions: 5
Please enter transacton    0: 1.00
Please enter transacton    1: 500.00
Please enter transacton    2: -39.56
Please enter transacton    3: -28.45
Please enter transacton    4: 1400.01
Please enter the daily starting balance: 100.00
```

#	BALANCE	WITHDRAW	DEPOSIT
0	101.00		1.00
1	601.00		500.00
2	561.44	-39.56	
3	532.99	-28.45	
4	1933.00		1400.01

7. Matrix Transposition

Create a function that takes a two-dimensional integer array representing a matrix and returns its transpose (rows become columns and vice versa).

```
PS C:\Users\wes\github-repos\cs3150_fall2023\In-Class 2> .\wshakespear_part7.exe
Transposing the following matrix:
    0      1      2      3
    4      5      6      7
    8      9     10     11

Result:
    0      4      8
    1      5      9
    2      6     10
    3      7     11

Transposing the following matrix:
20834  13307  5333
 6663   3612 30225
17420   4410 27197
11195   5917 17683
11216   8944  1016
29780  29913  6830
 7794  12555 26240

Result:
20834   6663  17420  11195  11216  29780   7794
13307   3612   4410   5917   8944  29913  12555
 5333   30225  27197  17683   1016   6830  26240
```

8. Prime Number Check

Write a function that checks if a given integer is a prime number or not. Return 1 if it's prime and 0 otherwise.

```
PS C:\Users\wes\github-repos\cs3150_fall2023\In-Class 2> .\wshakespear_part8.exe
Enter an integer to test of the absolute value is prime (0 to exit): 10
The number 10 is not prime
Enter an integer to test of the absolute value is prime (0 to exit): 11
The number 11 is prime
Enter an integer to test of the absolute value is prime (0 to exit): -13
The number 13 is prime
Enter an integer to test of the absolute value is prime (0 to exit): 56
The number 56 is not prime
Enter an integer to test of the absolute value is prime (0 to exit): 57
The number 57 is not prime
Enter an integer to test of the absolute value is prime (0 to exit): 101
The number 101 is prime
Enter an integer to test of the absolute value is prime (0 to exit): 0
```

9. String Reversal

Implement a function that reverses a string in-place. Pass the string as a character array to the function.

In-Class Assignment 2

```
PS C:\Users\wes\github-repos\cs3150_fall2023\In-Class 2> .\wshakespear_part9.exe
Reversing the string: 'Weston Shakespear'
The reversed string is: 'raepsekaHs notseW'

Reversing the string: 'I Love Computer Science'
The reversed string is: 'ecneicS retupmoC evoL I'
```

10. Recursive Fibonacci

Write a C program to calculate the nth Fibonacci number using a recursive function. Prompt the user for the value of n and display the result.

```
PS C:\Users\wes\github-repos\cs3150_fall2023\In-Class 2> .\wshakespear_part10.exe
To find the fibonacci number enter a positive integer (0 to exit): 1
Fibonacci number 1 is 1
To find the fibonacci number enter a positive integer (0 to exit): 2
Fibonacci number 2 is 1
To find the fibonacci number enter a positive integer (0 to exit): 3
Fibonacci number 3 is 2
To find the fibonacci number enter a positive integer (0 to exit): 4
Fibonacci number 4 is 3
To find the fibonacci number enter a positive integer (0 to exit): 5
Fibonacci number 5 is 5
To find the fibonacci number enter a positive integer (0 to exit): 10
Fibonacci number 10 is 55
To find the fibonacci number enter a positive integer (0 to exit): 40
Fibonacci number 40 is 102334155
To find the fibonacci number enter a positive integer (0 to exit): 0
```

In-Class Assignment 2
