Scientific Programming

Anand Kamble

19th September 2023

1 Introduction

We will be writing this program in ‘C’ and will be compiled using the compiler gcc.

2 Programming Style

Each function mentioned in this report below is written in its own file which is named ’Q’ followed by the question number. e.g. File which includes the function ’Question1’ will be named ’Q1.c’. These files are then imported into the ’main.c’ and the main function will call all of the functions.

Below are the formatting conventions used for this code.
Casing: Camel-Case
Indentation: 4 spaces
Line break: LF

3 Functions

3.1 Question 1

The function named ’Question 1’ can be found in the file ’Q1.c’. This function reads a file into the memory and then counts the times the character ’a’ and the word ’the’ appears in the text file. The paragraph which is used for this demonstration is generated from the website randomwordgenerator.com [1]
Output:

```
27 ’a’s and 5 ’the’s are found inside the text.
Total number of words in this file is 96.
```

3.2 Question 2

The function named 'Question 2' can be found in the file 'Q2.c' and it creates two $10 * 10$ matrices. The matrices are named A and B, and their elements are calculated by,

$$A[i][j] = i + j + 1.0 \ \& \ B[i][j] = 1/(i + j + 1.0)$$

For the above equations, we can optimize the code by calculating,

$$x = i + j + 1.0$$

and then substituting in A and B.

$$A[i][j] = x \ \& \ B[i][j] = 1/x$$

Output:

```
File matrices.dat successfully created.
```

3.3 Question 3

This function which is written in file 'Q3.c', reads the binary file which includes the two matrices A and B created using the function 'Question 2'. After successfully reading the file it multiplies the two matrices and stores the answer in another matrix named 'C'. In the end calculates the average of all the elements in the matrix 'C'.
Output:

```
Average of all elements of C : 11.687714
```

3.4 Question 4

This implements the function named 'reverse' which prints the elements of an array in reverse order starting from the last element. This is a recursive function that iteratively invokes itself until it exhausts the available elements.
Output:

```
10.000000
9.000000
8.000000
7.000000
6.000000
5.000000
4.000000
3.000000
2.000000
1.000000
```

3.5 Question 5

We are creating a sequence where each element is the sum of the last three elements. In this problem, we are using the type long long to store out integers since we will need higher precision. With the precision provided by the type long long, we can calculate up to 72 terms in the sequence. After the 72nd term, the limit of the long long int is exceeded and hence later terms cannot be calculated.[2]
Output:

```
72nd term in the sequence is 5887809802922991460.
```

3.6 Question 6

In this function, we are creating a trinary tree and inserting 50 random digits in it which results in a tree with 50 nodes. The number of nodes are calculated by the function 'TotalNumberOfNodes', which is a recursive function and invokes itself whenever the root node is not null.[3]
Output:

```
Number of nodes in the trinary tree = 50
```

3.7 Question 7

In this function, we are allocating one block of memory for storing a 2D array, where the first 10 elements are the starting addresses of each row.
Output:

```
ADDRESS OF A[0] : 00AC7848
ADDRESS OF A[1] : 00AC784C
DIFF BETWEEN A[0] & A[1] : 1
ADDRESS OF A[0][3] : 00AC7854
ADDRESS OF A[1][3] : 00AC787C
DIFF BETWEEN A[0][3] & A[1][3] : 10
```

4 Execution

To execute the program, we need to compile the main.c file by running,

```
gcc main.c
```

After successfully compiling, you should get an executable file. It is not recommended to compile files that are named like Q1, Q2,..., Q7 since those files do not include a main method.

Note: This code has been tested on Windows by compiling using GCC v6.3.0.

References

[1] ChatGPT conversation:
https://chat.openai.com/share/e1cccee6-7ca4-4ef7-9afd-6a4ab208f66a.

[2] Data Types in C
https://www.freecodecamp.org/news/data-types-in-c-integer-floating-point-and-void-explai

[3] ChatGPT conversation:
https://chat.openai.com/share/b1c25d3e-b84f-4318-a2ff-94d7b3837b16