

BLG221E

DATA STRUCTURES

CRN:22450

REPORT OF HOMEWORK3

Submission Date:15.04.12

STUDENT NAME:

STUDENT NUMBER:.....

1.INTRODUCTION

In this homework, the aim was to merge two arrays which are given in files using queues and a stack. 3 file names will be entered by the user to the terminal to execute the program. First two parameters will represent input file names which are "giris1.txt" and "giris2.txt" that include the sorted numbers in increasing manner, the last parameter will be the file name which will include the merged struct's content.

The queue Q1 will contain the numbers in file "giris1.txt", while the second queue Q2 will contain the numbers in file "giris2.txt". After those enqueues are done, these queues will be popped into the stack in an increasing manner, meaning that the bottom of the stack must include the number with the highest value, and the top of the stack must include the smallest one. And meanwhile if equal values from queues are read, only one of them should be pushed to the stack. At the end, the program should pop the stack and write these values to the file, named "sonuc.txt".

2. Development and Operating Environments

Microsoft Visual C++ environment has been used to write the source code, whereas Linux operating system was used to compile and run the program in GNU C++ compiler. It consists of only 1 source code which is named "StudentID_hw3.cpp". To use GNU C++ as our compiler, we need to go Start>Run and write cmd, click OK. In the DOS command prompt, *cd* to the directory where the program is located. To compile, *g++ StudentID_hw3.cpp -o StudentID_hw3.exe* needs to be written, whereas linking the program is via *StudentID_hw3.exe*. And finally the program execution is completed by typing *StudentID_hw3 giris1.txt giris2.txt sonuc.txt*. an example of the outcome is below:

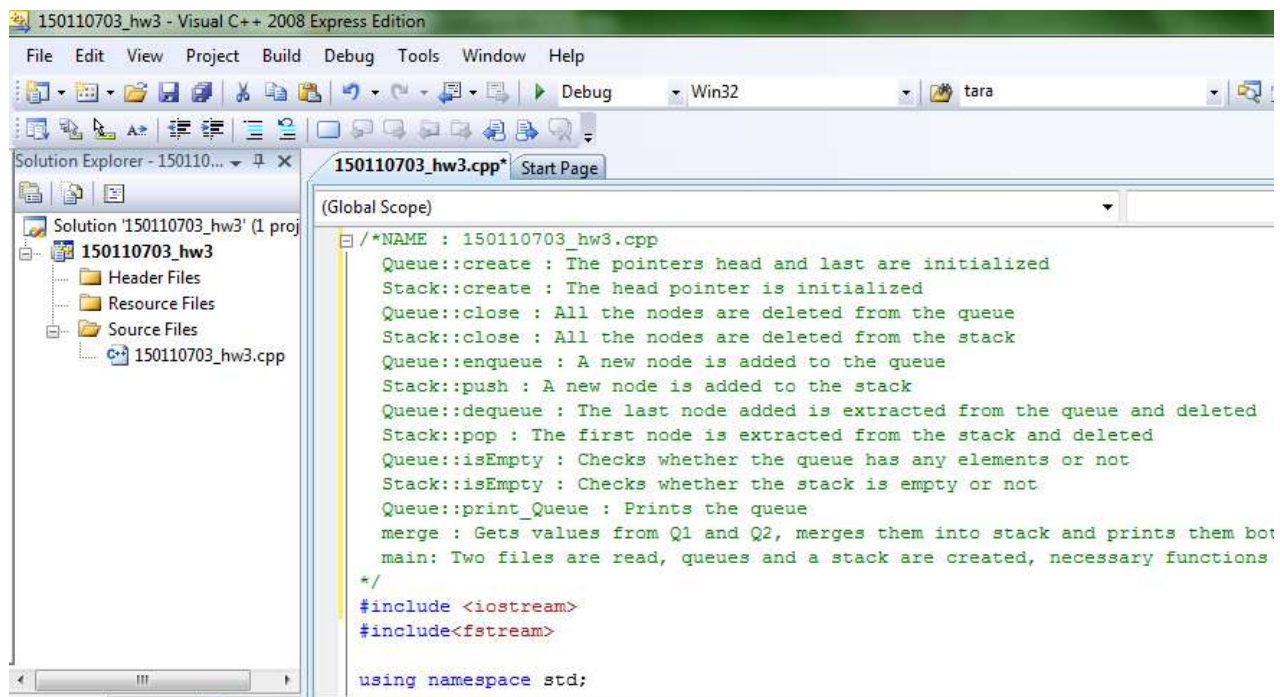
```
C:\Users\nurcan\Desktop>g++ 150110703_hw3.cpp -o 150110703_hw3.exe
C:\Users\nurcan\Desktop>150110703_hw3.exe
150110703_hw3.exe
Queue 1 is empty
Queue 2 is empty
C:\Users\nurcan\Desktop>150110703_hw3 giris1.txt giris2.txt sonuc.txt

Queue 1 is : <600,594,588,582,576,570,564,558,552,546,540,534,528,522,516,510,504,498,492,486,480,474,468,462,456,450,444,438,432,426,420,414,408,402,396,390,384,378,372,366,360,354,348,342,336,330,324,318,312,306,300,294,288,282,276,270,264,258,252,246,240,234,228,222,216,210,204,198,192,186,180,174,168,162,156,150,144,138,132,126,120,114,108,102,96,90,84,78,72,66,60,54,48,42,36,30,24,18,12,6,0,>

Queue 2 is : <700,693,686,679,672,665,658,651,644,637,630,623,616,609,602,595,588,581,574,567,560,553,546,539,532,525,518,511,504,497,490,483,476,469,462,455,448,441,434,427,420,413,406,399,392,385,378,371,364,357,350,343,336,329,322,315,308,301,294,287,280,273,266,259,252,245,238,231,224,217,210,203,196,189,182,175,168,161,154,147,140,133,126,119,112,105,98,91,84,77,70,63,56,49,42,35,28,21,14,7,0,>

Stack is : <0,6,7,12,14,18,21,24,28,30,35,36,42,48,49,54,56,60,63,66,70,72,77,78,84,90,91,96,98,102,105,108,112,114,119,120,126,132,133,138,140,144,147,150,154,156,161,162,168,174,175,180,182,186,189,192,196,198,203,204,210,216,217,222,224,228,231,234,238,240,245,246,252,258,259,264,266,270,273,276,280,282,287,288,294,300,301,306,308,312,315,318,322,324,329,330,336,342,343,348,350,354,357,360,364,366,371,372,378,384,385,390,392,396,399,402,406,408,413,414,420,426,427,432,434,438,441,444,448,450,455,456,462,468,469,474,476,480,483,486,490,492,497,498,504,510,511,516,518,522,525,528,532,534,539,540,546,552,553,558,560,564,567,570,574,576,581,582,588,594,595,600,602,609,616,623,630,637,644,651,658,665,672,679,686,693,700,>
```

The look on the compiler Visual C++:



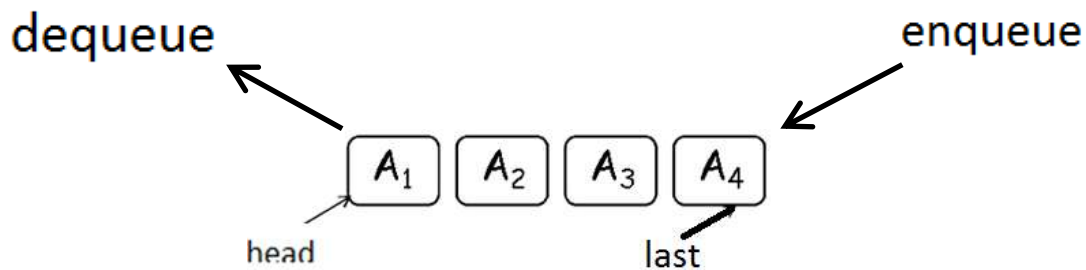
3. Data Structures and Variables

In the program, the local variables are:

- fileToRead1 : First argument entered right after the program name. Used to get the file named "giris1.txt". It is of char type.
- fileToRead2 : The argument taken after fileToRead. It is used to get the file named "giris2.txt". Its type is character.
- fileToWrite : The last argument entered to the terminal to create a file to write the stack on, named "sonuc.txt"
- file1 : Variable used to read data from the file named "giris1.txt" as input streams. It is of type ifstream.
- file2 : Variable used to read from the corresponding file "giris2.txt" as input streams. Its type is ifstream.
- myfile : The variable to write the data to the file "sonuc.txt" as output streams. It is of type ofstream.
- Q1 : Variable of type Queue to enqueue the values in the file "giris1.txt"
- Q2 : Variable of type Queue to enqueue the values in the file "giris2.txt"
- S1 : Variable of type Stack to merge the values in the queues on.

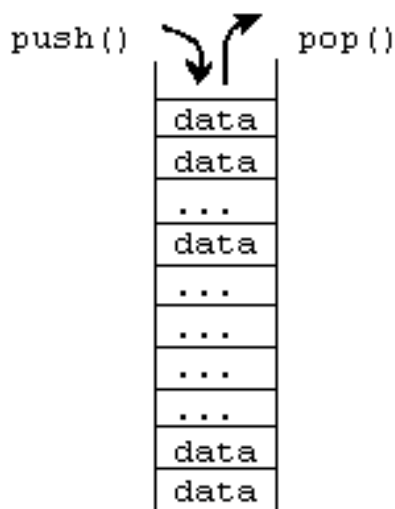
As you see, 2 types of data structures were used in this homework, queue and stack. The struct framing for queue is as follows with its 2 pointer variables and member functions:

```
struct Queue{
    PROGRAM CODES
};
```

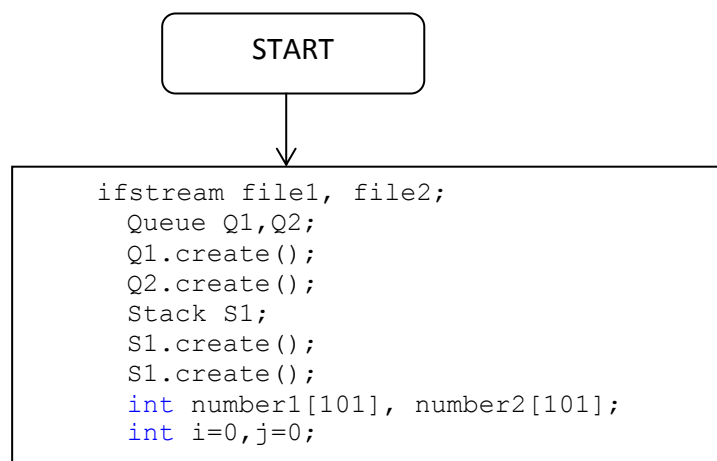


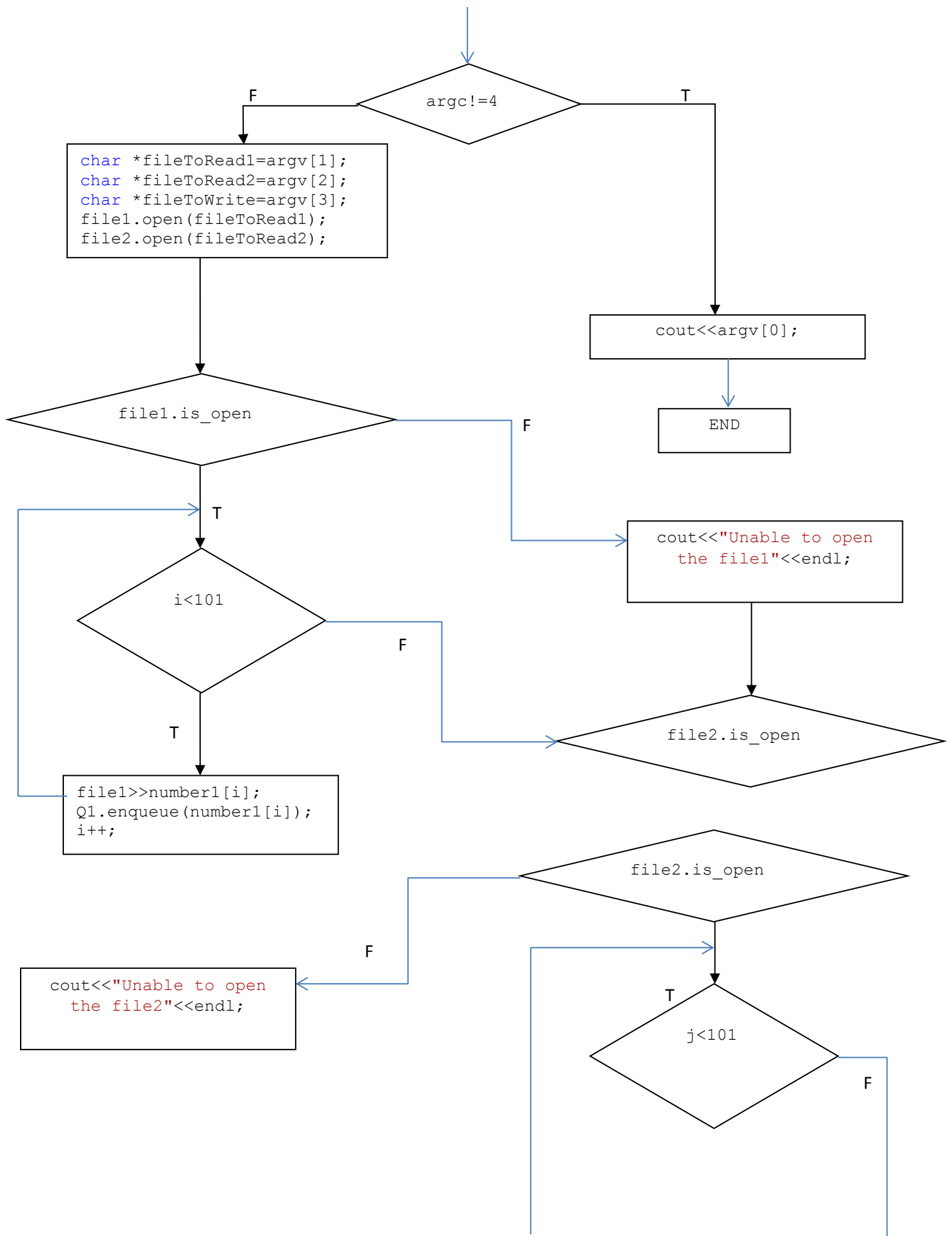
The struct framing for Stack with its one pointer and member functions:

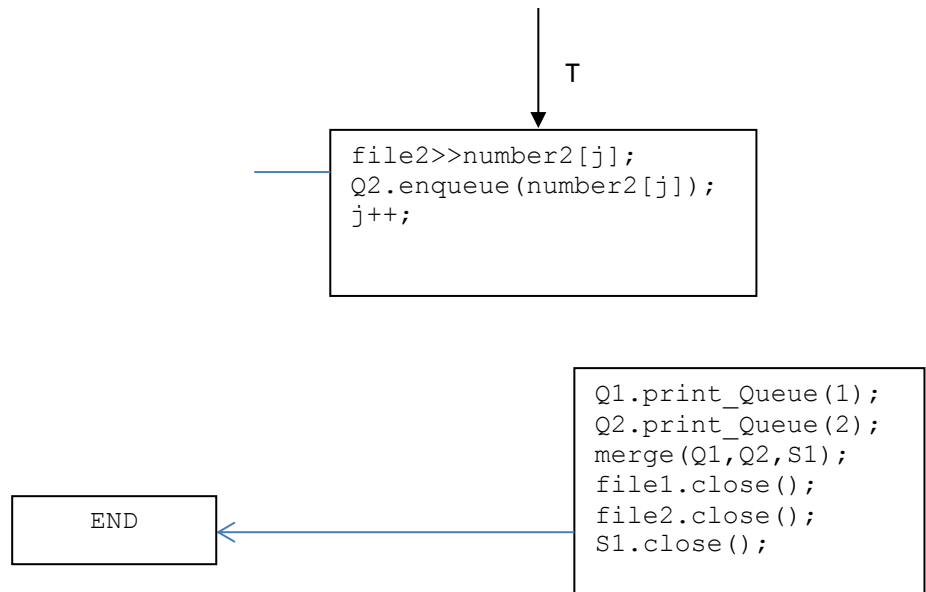
```
struct Stack{
    PROGRAM CODES
};
```



4. Program Flow







Now, let's examine the merge function in detail with its pseudocode:

```

/*****
* Function Name: merge
* Aim: Gets values from Q1 and Q2, merges them into stack and prints them
both in console and file "sonuc.txt"
* Input Parameter: The two queues to be merged and the stack to push those
values in
* Output Parameter: None
* Data Structures: Queue, Stack
*****/
  
```

```

void merge(Queue Q1, Queue Q2, Stack S1){
    PROGRAM CODES
}
  
```

Below, you see the output file, containing the merged data written in the stack:

```

Stack is : {0,6,7,12,14,18,21,24,28,30,35,36,42,48,49,54,56,60,63,66,70,72,77,78,84,
90,91,96,98,102,105,108,112,114,119,120,126,132,133,138,140,144,147,150,154,156,161,
162,168,174,175,180,182,186,189,192,196,198,203,204,210,216,217,222,224,228,231,234,
238,240,245,246,252,258,259,264,266,270,273,276,280,282,287,288,294,300,301,306,308,
312,315,318,322,324,329,330,336,342,343,348,350,354,357,360,364,366,371,372,378,384,
385,390,392,396,399,402,406,408,413,414,420,426,427,432,434,438,441,444,448,450,455,
456,462,468,469,474,476,480,483,486,490,492,497,498,504,510,511,516,518,522,525,528,
532,534,539,540,546,552,553,558,560,564,567,570,574,576,581,582,588,594,595,600,602,
609,616,623,630,637,644,651,658,665,672,679,686,693,700,}
  
```

5. Conclusion

In this homework, I have become more familiar with the concept of 2 data structures, queues and stacks. I had the chance to intensify my knowledge about their structures. And also dealing with file operations is always useful since it keeps my memory fresh about them.