Project 7

**PRIME FACTORIZATION USING AN ARRAY-BASED STACK**

Full Name: Prashul Shrestha

Section #: 2

Project #: 6

Due Date: March 27, 2017

**Design** **Document**

**Introduction**

A prime integer is those factors that are only divisible by itself or by 1. We already printing out the prime numbers in an increasing order. For this program, we are using stack to write out the list of Prime Integer factor of a given number by user in a descending order.

**Data** **Structures**

The programs uses a Class named stack () that stores all the functions and values for stack and an Array named factors[] to hold the Prime factors in stack.

**Functions**

There is an inline function, a destructor and three module function in the class. The inline functions bool empty() returns true when the size of array is 0.

We have used three member functions : one function pushes the factor in an Array, another function pop out an item from an Array and next function isFactor returns Boolean statement by checking if the passed number is 0 or not.

**The Main Program**

The integer are pushed and popped from the main menu. We use push when taking input of the prime factors and pop out to print out the contents of prime factor in a integer. We also use menu to check if the factor is prime or not before pushing it to an Array.

User Document

A prime integer is those factors that are only divisible by itself or by 1. We already printing out the prime numbers in an increasing order. For this program, we are using stack to write out the list of Prime Integer factor of a given number by user in a descending order.

The main program named **main**.**cpp** can be compiled and run, using the code:

**g++ main**.**cpp**

**a.out**

**g++** function will compile the function and make it ready to be run using **a.out**. The function will prompt the following output:

**Ouptut:**

Enter a positive integer (0 to stop): 1776

Prime factors: 1776 = 37 x 3 x 2 x 2 x 2 x 2

Enter a positive integer (0 to stop): 6463

Prime factors: 6463 = 281 x 23

Enter a positive integer (0 to stop): 349856

Prime factors: 349856 = 29 x 29 x 13 x 2 x 2 x 2 x 2 x 2

Enter a positive integer (0 to stop): 352170

Prime factors: 352170 = 43 x 13 x 7 x 5 x 3 x 3 x 2

**Summary**

From this Project I made my concept clearer on using stack on different Platforms. The stack was used to solve problem of factorization and printing it in the sequence in opposite order which could be useful for linked list stacks.

It is better not necessary to use stack to print factors in increasing order. Because in stack, we use it to bring out the last element from the list. And while doing factorization we are storing factors in increasing order and to print out elements in ascending order we just don’t need stack.