Aleesha Nageer
CSE 13S
November 8, 2020

CSE 13S – Fall 2020
Assignment 4: Bit Vectors and Primes
Design Document

Prime numbers are numbers that are only divisible by 1 and itself. In this lab, we utilize bit vectors to isolate prime numbers from a range of integers, inputted by the user, and decipher whether they are Mersenne, Lucas, or Fibonacci prime numbers.

The inputs to the program are:

- -s: This prints out all of the prime numbers for the value inputted and labels them according to if they are Mersenne, Lucas, or Fibonacci prime numbers
- -p: This prints out all of the palindromic prime numbers in bases 2, 9, 10, and 24.
- -n <value>: This specifies the range and the largest value that the sieve program should run through to find all the prime numbers

PRE-LAB pt. 1

1. Assuming you have a list of primes to consult, write pseudo-code to determine if a number is a Fibonacci prime, a Lucas prime, and/or a Mersenne prime.

Fibonacci

```
Calculate Fibonacci Numbers
N1 = 0
N2 = 1
Loop twice to add first two numbers, then the next two, and so on…
Check if Fibonacci number = prime number
If they are equal, return true
Else if the Fibonacci number is greater than the prime number, return false
```

Lucas

```
Calculate Fibonacci Numbers
N1 = 2
N2 = 1
Loop twice to add first two numbers, then the next two, and so on…
Check if Fibonacci number = prime number
If they are equal, return true
Else if the Lucas number is greater than the prime number, return false
```

Mersenne

```
Input prime numbers in 2^n – 1 equation for every prime
Check if Mersenne prime number = prime number
If they are equal, return true
Else if the Mersenne prime number is greater than the prime number, return false
```

2. Assuming you have a list of primes to consult, write pseudo-code to determine if a number in base 10 is a palindrome. Note that the technique is the same in any base

```
First, reverse the number
While number doesn’t equal 0
Calculate the remainder of the number divided by 10
Reversed Number = Reversed Number * 10 + Remainder
Divide number by 10

Once Number equals 0, check to see if original number equals reversed number
If they do, the number is a palindrome!
Else, it is not a palindrome.
```

PRE-LAB pt. 2

1. Implement each BitVector ADT function.
- I have implemented each BitVector ADT function utilizing shifts, and/or operators, masking, and several other bitwise functions.

2. Explain how you avoid memory leaks when you free allocated memory for your BitVector ADT.
- I avoided memory leaks when freeing allocated memory by ensuring the vectors as well as the bits within the vectors were accounted for.

3. While the algorithm in sieve() is correct, it has room for improvement. What change would you make to the code in sieve() to improve the runtime?
- Currently, the sieve algorithm has a runtime of O(n^2) do to the nested for loop. In order to improve the runtime, I would find a way to include only one loop for sieving the prime numbers from the range provided.

NOTE:

This assignment was quite difficult for me and required much redesigning for my pseudocode. I had to rewrite how I solved the BitVector code several times. I also need to update my bv code since I still have to add conditional statements to filter out the prime numbers. In addition to this, I wish I could update how I found the palindromes for each base. I know there is a way to have it all in one function, but I was running out of time and could not wrap my brain around the best way of going about the problem. I also am aware that I do not properly print out the palindrome for base 24 and still have yet to solve the problem of converting my integers to ASCII characters.