

PRELAB PART 1

1. For i in Prime numbers:

 check(Fibonacci)

 check(Lucas)

 Check (mer)i

Fibonacci(n):

Set term 0=0

Set term 1=1

If (n<=1) : return n

Take n and get (n-1) +n-2) within the length of n

Lucas(n):

term1=2

term2=1

Return n-1+n-2

meri(n):

Return 2^{n-1}

2. For i in length of n:

 Go into Fib,lucas,meri function

 If i== return value prime

 Then print

PRELAB PART 2

BitVector ADT:

BV_create:

Allocate space in the struct->vector

Allocate length

BV_delete:

free(v)

BV_getlength:

Return Vector of length

BV_set:

Place 1 at desire position of i/8

BV_clear:
Put ~1 at desire position of i/8

BV_get:
Get value at i/8 position

BV_set_all:
For i in length of input:
Set_bit to 1

2.To avoid memory leak, I would use a loop to go through each index in array and free(), then i would set them to NULL

3.I guess the only bad thing about the sieve code, is its $O(n^2)$?
I would probably not set them all to 1's and just leave it with zeros

PSUEDOCODE:

Int main{
Create BitVector

getopt()--Take in argument in command line with flags

-s=prime function, -p=palindrome, -n=number length

Case 's':
Call sieve function
For each prime number in vector array \leq number length:
 Check Lucas, check Fibo, check Meri:
 Print if the prime number is a part of the above functions

Case 'p'
Call sieve function
For each prime in vector array \leq length:
 Call Convert(i) function
 Check if Palindrome:
 If palindrome then Print

Convert(n):
While $n > 0$:
 $n \% \text{base}$ wanted

Save input in array
n/base wanted
return