

Remember to include honor code.

How to submit?

CS 551 Assignment 0

Handling Bitmaps and Pointers in C.

Due Friday September 2, 2016 at 10PM

Do we always have to use sizeof?

Write a C program called goldbach.c. The program takes one command line argument "N" and computes a bitmap of primes less than N by using a sieve method. After the bitmap is constructed, print out the number of primes found, and then read in even integers from stdin until eof is reached. For each integer use the bitmap to decompose the integer into the sum of two odd primes, where the smaller prime is as large as possible. Output the decomposition in a display similar to the demo program /home/cs551000/share/goldbach including the total number of decompositions.

Notes:

- (1) The bitmap in fact will be represented as a doubly linked list of segments each containing 8192 bits. Define a segment using the structure:

```
typedef struct _seg {
    int bits[256];
    struct _seg *next, *prev;
} seg;
```

See sample.c for a longer sample. You should dynamically allocate enough segments to represent all odd numbers $\leq N$.

- (2) Initially all bits should be set to 0. Use the sieve method for marking ON the bits corresponding to composite numbers less than or equal to N. (Note: The sieve method marks on multiples of primes, and is done when it finishes marking multiples of primes not exceeding \sqrt{N}).

- (3) When you complete building the bitmap, take one pass through it counting the 0 bits, and print out

The number of odd primes less than or equal to N is: XXXX

- (4) To decompose the number K, start I at 3 and increase by 2's using the bitmap to test "I" and "K-I" for primality. "I" increasing by 2's mean moving 1 bit up for I, and 1 bit down for "K-I". This is why the linked list is doubly linked. Keep increasing I counting the solutions and print the solution with the largest I less than or equal to K/2 as well as the number of solutions.

- (5) You must write the 3 functions:

```
*seg whichseg (int j)
int whichint (int j)
int whichbit (int j)
```

which locate the "coordinates" of the bit corresponding to the number j. This is used in the sieve when marking the non-primes. In the Goldbach portion it is only to locate the starting point for I and K-I, since the "prev" and "next" links are used to walk backwards and forwards through the bitmap. Since whichseg is mainly used with increasing j, use the last returned value to implement it efficiently.

- (6) Note, N is NOT the number of segments required.

- (7) Do not provide any prompts or require any user interaction beyond entering the numbers. Do not ask if the user is done, you must detect EOF. The input could be coming from a redirected file or the keyboard. The program needs not know which. We will discuss how EOF is indicated from the keyboard in class.

Don't change the exact message

Torun: a.out Test-number -> arg

Head of list:

```
struct seg * head
struct seg * p
head = p;
```

```
p = malloc(1024 * sizeof(seg));
head = p;
```

256 x 32 bits = 1024 bytes
256 x 4 bytes = 1024 bytes

To use, (eg. set 0)
p->bits[0] = 0;

Compile error

I, prime 1 + prime 2

```
whichbit = seg - modular / 32;
#define SetBit(A, b) (A[(b/32)] |= (1 << (b%32)))
```

```
#define ClearBit(A, b) (A[(b/32)] &= ~(1 << (b%32)))
#define TestBit(A, b) (A[(b/32)] & (1 << (b%32)))
```

(main scanf) -> EOF