Prelab 2

Shiyu Wang

Pre 2.1) Printing the value of variables using printf statements is one simple debugging approach. However, it is inefficient. Describe how and why it is inefficient.

Printing the value of variables is Ad-Hoc. Although it has advantages that no special debugging tools needed, it requires intimate knowledge of code and expected values. Also it needs frequent re-compile and execute cycles. Moreover, inserted code can be buggy.

Pre 2.2) Define in your own words (or as guided by the textbook) what is a debugger?

A debugger is a program or tool used to test and debug other programs. Breakpoints and single-stepping are common method that used by debugger.

Pre 2.3) What is the gcc compiler flag for enabling debug symbols?

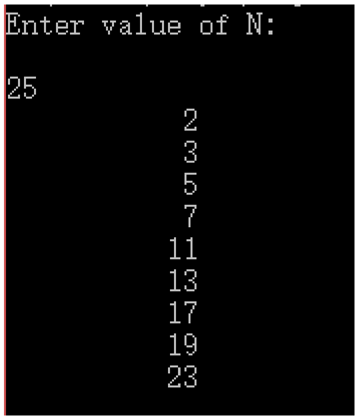
gcc -g to compile with debugging flags, for use with gdb.

Pre 2.4) What is the difference between a syntax error and a programming bug?

A syntax error is usually easy to find and fix, such as input code is not legal, or errors caught by compiler and other translation mechanism.

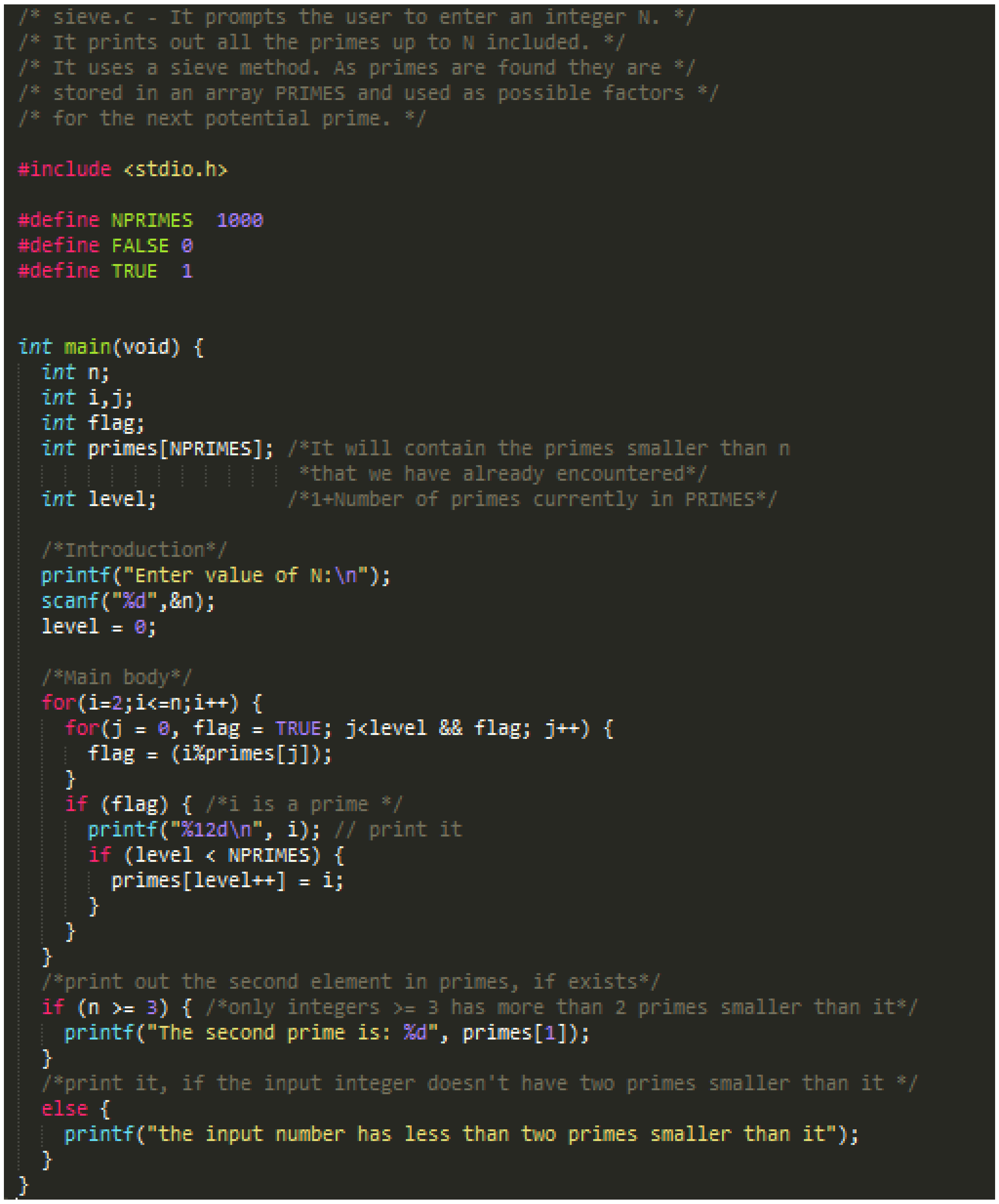
A programming bug is an error, flaw, failure or fault in a program that causes it to produce an incorrect or unexpected result. This is usually hard to find or fix. So we often use debugger to help us find them.

Pre 2.5) Compile sieve.c program and execute with entering 25



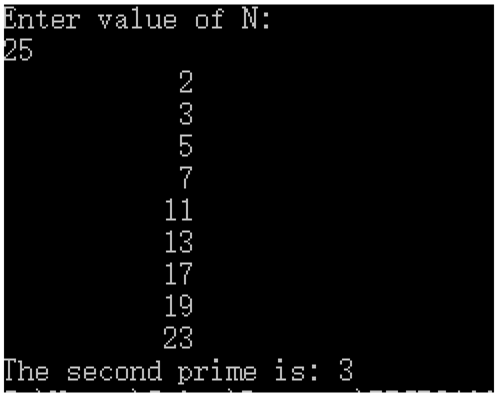
```
Enter value of N:

25
           2
           3
           5
           7
          11
          13
          17
          19
          23
```

Modify the program to include a print statement in the end



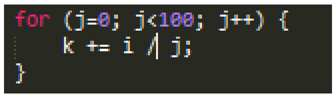
```c
/* sieve.c - It prompts the user to enter an integer N. */
/* It prints out all the primes up to N included. */
/* It uses a sieve method. As primes are found they are */
/* stored in an array PRIMES and used as possible factors */
/* for the next potential prime. */

#include <stdio.h>

#define NPRIMES  1000
#define FALSE 0
#define TRUE  1


int main(void) {
  int n;
  int i,j;
  int flag;
  int primes[NPRIMES]; /*It will contain the primes smaller than n
                        *that we have already encountered*/
  int level;           /*1+Number of primes currently in PRIMES*/

  /*Introduction*/
  printf("Enter value of N:\n");
  scanf("%d",&n);
  level = 0;

  /*Main body*/
  for(i=2;i<=n;i++) {
    for(j = 0, flag = TRUE; j<level && flag; j++) {
      flag = (i%primes[j]);
    }
    if (flag) { /*i is a prime */
      printf("%12d\n", i); // print it
      if (level < NPRIMES) {
        primes[level++] = i;
      }
    }
  }
  /*print out the second element in primes, if exists*/
  if (n >= 3) { /*only integers >= 3 has more than 2 primes smaller than it*/
    printf("The second prime is: %d", primes[1]);
  }
  /*print it, if the input integer doesn't have two primes smaller than it */
  else {
    printf("the input number has less than two primes smaller than it");
  }
}
```

Output:



```
Enter value of N:
25
          2
          3
          5
          7
         11
         13
         17
         19
         23
The second prime is: 3
```

Pre 2.6) The code has an error because of



```
for (j=0; j<100; j++) {
    k += i / j;
}
```

During this loop, The j can be 0 in the beginning, which is i / 0, so it gives an error.