

CS 1723, Data Structures
Fall Semester, 1998
Programming Assignment 2, Due September 18, 1998
Palindrome Primes

The assignment: For this assignment, you are to write a program which finds all integers less than 20000 which are prime numbers and which are base 10 palindromes at the same time. An integer is *prime* if no other integers divide evenly into it. (29 and 47 are primes, while 51 is not, since 3 divides evenly into 51.) An integer is a *palindrome* (to base 10) if its digits are the same written forwards or backwards. (13231 is a palindrome, while 13321 is not. 13331 is both a prime and a palindrome.)

Checking that an integer is prime: This is fairly easy. Write a function with prototype

```
int isprime(int n);
```

`isprime` should check each integer `i` less than or equal to `sqrt(n)` to see if `i` divides evenly into `n`. This can be done by checking if `(n%i) == 0`. Don't forget to include `<math.h>` and to use `-lm` at the end of the `lint` and `cc` commands. Of course return 0 or false if `n` is not a prime and return 1 if `n` is a prime.

Checking that an integer is a palindrome: For this part you are required to use a stack and a queue, and to use the following function. (There are easier ways to check if an integer is a palindrome, but for this assignment we are interested in implementing a stack and a queue.)

```
/* ispalin: checks base 10 digits of n, to see if
 * they are the same forwards and backwards.
 * Uses both a stack and a queue.
 */
int ispalin(int n)
{
    int digit, digs, digq;
    int returnval = 1;
    while (n != 0) {
        digit = n%10;
        push(digit);
        insert(digit);
        n = n/10;
    }
    while (!emptys()) {
        digs = pop();
        digq = delete();
        if (digs != digq) returnval = 0;
    }
    return returnval;
}
```

`ispalin` generates the decimal digits of `n` one at a time, pushes each digit on a stack and also inserts each digit into a queue. Later when each digit is popped and removed, respectively, the queue's digits will come out in the order they went in, while the stack's digits will come out in the opposite order. Thus when popping and removing, if all the digits are the same then the number must be a base 10 palindrome.

The stack: For your stack, you must use two separate files, `stack.h` and `stack.c`, and these can be almost the same as what was handed out in class, except that the `StackType` should be `int`, and the `empty` and `full` functions should be renamed `emptyq` and `fullq` (so that they won't be the same as those for the queue.).

The queue: The queue should be an array-based circular queue modeled after the stack. You can copy and modify the stack files to get two new files `queue.h` and `queue.c` for the queue. The queue header file `queue.h` should look like:

```
/*
 * queue.h -- queue header file
 */
typedef int QueueType;
QueueType delete(void);
void insert(QueueType);
int emptyq(void);
int fullq(void);
```

The size of the array used for this queue must be no more than 10, so that you will be demonstrating that the wrap-around aspect of your queue works.

The whole program: Name the file with the main function `palin.c`. It also includes the functions `isprime` and `ispalin`. Your whole program should consist of 5 separate files. Here is a simple makefile that you could use. (Don't forget to use a "TAB" character before the `cc` and the `lint` below.)

```
palin: palin.c stack.c stack.h queue.c queue.h
      cc -g -o palin palin.c stack.c queue.c -lm
lint:
      lint -m -u palin.c stack.c queue.c -lm
```

In the main function you should check integers from 2 to 20000 to see which ones are both primes and palindromes. (Hint: there are none between 1000 and 10000, but, for example, 17471 is both a prime and a palindrome.)

What to turn in: Just turn in a listing with the five files `palin.c`, `stack.c`, `stack_int.h`, `queue.h`, and `queue.c` along with the integer answers (I got 46 of them) concatenated together.