ES112 Spring 2003-2004 Final

1 Product of Cosines *(25 points)*

Your task in this question is to write a **function** that calculates the value of the following product:

$$\prod_{k=3}^{n} \cos\left(\frac{\pi}{k}\right) = \cos\left(\frac{\pi}{3}\right)\cos\left(\frac{\pi}{4}\right)\cdots\cos\left(\frac{\pi}{n}\right)$$

The prototype of the function should be:

```
double cosineProduct(int n);
```

Here, n is the n given in the above formula. You *are* allowed to use math library functions.

2 String to Integer *(25 points)*

Write a C **function** that takes a string as an argument, and returns its numeric value as an integer. We will assume that the string contains a positive integral number, and consists of digits only. So, the string passed to the function could be “2134”, “12308”, “72”, or “007”. The prototype should be:

```
int stringToInteger(const char *s);
```

You *may not* use any C library functions in the implementation of this function. (Recall that the character codes that represent the digits are not equal to the digits. Also, you may make use of the fact that digits are arranged sequentially starting with ’0’ in the character set.)

3 File Checksum *(25 points)*

Write a C **function** that takes a filename as an argument, and if that file exists, calculates and returns an integer “checksum” for that file. The exact method to be used to calculate the checksum is up to you. A checksum is usually used to determine whether a file has been altered or damaged (perhaps during download). So, the checksum you calculate should change when one byte of the file has changed. Make sure the checksum changes, for instance, when bytes with the value zero are appended to the file. The prototype is:

```
int fileChecksum(const char *filename);
```

4 What am I Doing? *(25 points)*

```
int zomtrak(int a, int b)
{
  int z=0;
  int k;

  for (k = 0; k < 32; k++) {
    if (a & (1 << k)) {
      z += b << k;
    }
  }

  return p;
}
```

Explain what this function does. Write another (much shorter and simpler!) function that produces the same return value. (*Hint: Try a few small values for the arguments...*)