ES 112 First Midterm Examination
-B-
Fall 2003-2004

1 A Game of Dice

Two friends have decided to play a game of dice as follows: Each is going to roll a single, six-sided die once. If the difference between the two values is two or greater, the one that rolls the higher value wins, and the game ends. If the difference is less than two, the game continues, and each player rolls again.

Your task in this problem is to write a C program that will simulate this game, and print how the game proceeds and how it ends. Output from a sample run should look as follows:

```
Player A rolls: 3
Player B rolls: 4

The game continues...
Player A rolls: 5
Player B rolls: 4

The game continues...
Player A rolls: 2
Player B rolls: 6

Player B wins!
```

You should try to match the output of your program as closely as possible to the samples here, including newlines and blank lines.

2 Product of Terms

A product P_n is defined as follows:

$$P_n = \prod_{k=2}^{n} \left(1 - \frac{1}{k^2}\right)$$

So, for example:

$$\begin{aligned} P_2 = (1 - 1/2^2) &= 0.7500 \\ P_3 = (1 - 1/2^2)(1 - 1/3^2) &= 0.6666 \\ P_4 = (1 - 1/2^2)(1 - 1/3^2)(1 - 1/4^2) &= 0.6250 \\ P_5 = (1 - 1/2^2)(1 - 1/3^2)(1 - 1/4^2)(1 - 1/25) &= 0.6000 \end{aligned}$$

a. Write a C *function* that takes one `integer` argument `n`, and returns the `n`th product as a `double`. In other words, the prototype of the function should be:

```
double product(int n);
```

b. Write a C program that uses the above function (assume the function is typed at the bottom of the file; do not write it again!) to do the following: Ask the user for an integer. If the user enters an integer greater than or equal to two, display the above product with the number the user has typed and go back and ask for another integer. If the user enters any integer less than two, the program should terminate.

A sample output of the program should be as follows:

```
Enter an integer greater than one (0 to end): 2
P2 = 0.7500

Enter an integer greater than one (0 to end): 4
P4 = 0.6250

Enter an integer greater than one (0 to end): 0
```

The output should print only four decimal places, as shown in the sample output. Again, you need to match the output of your program as closely as possible to the sample output, including newlines and blanks.