Practice problems on recursive functions.

(1) Write a recursive function rectangle(int n) that prints a rectangle with
n rows and 10 columns. For example it could be applied as follows:

```
int main() {
  cout << rectangle(5) << endl;
  return 0;
}
```

This program should output:

```
**********
**********
**********
**********
**********
```

How is this picture related to the simpler picture drawn by rectangle(4)?
What value of n makes the task of the function as easy as possible?

(2) Write a function with title:

int triangle(int n)

that calculates the triangular number whose specification is:

triangle(n) = n + triangle(n - 1) if n is positive
and triangle(0) = 0.

Why are these numbers known as triangular numbers?

(3) Write a recursive function secondDigit that could be called as follows:

```
int main() {
  cout << secondDigit(7295) << endl;
  return 0;
}
```

This program should output 2.

(4) Write a recursive function printBinary, that prints
a positive integer n in binary. For example, the following
program would output 10111:

```
int main() {
  cout << printBinary(23) << endl;
  return 0;
```

```
}
```

Which number is easy to print?
How is printing 23 related to printing 11 = (23 - 1)/2?

(5) Write a recursive function with title:

```
int toBinary(int n)
```

It could be used in the following main program which should print 10111.

```
int main() {
  int x;
  x = toBinary(23);
  cout << x << endl;
  return 0;
}
```

(6) Write a recursive function with title:

```
 string baseChange(int n, int base)
```

which converts the decimal number n to the given base.
For example,

```
  baseChange(5,156)
```

would return 1111, because 156 is 125 + 25 + 5 + 1 which is written as 1111 in base 5.