EEN118
Happy Friendly Test
20th October 2009

Name:

Student number:

Sign here if you did not give or receive aid in any form during this test, and did not consult any written or printed material apart from this test:

Don't write in these boxes.

Question	Out of	Grade
1	33	
2	33	
3	33	
4	1	

1.

Exactly what would be printed by this program?
Here’s a clue: the first line would be

```
                a:    int 3
```

```
void output(const string question, const int n)
{ cout << question << ":    int " << n << "\n"; }

void output(const string question, const float n)
{ cout << question << ":  float " << n << "\n"; }

void output(const string question, const double n)
{ cout << question << ": double " << n << "\n"; }

void output(const string question, const bool n)
{ cout << question << ":   bool " << n << "\n"; }

void main()
{
  output("a", 1+2);

  output("b", 13%5);

  output("c", 13%5==0);

  output("d", 120/6-2/6);

  const double e = 52/5+3/5;

  output("e", e);

  output("f", 4567/1000+1/2);

  output("g", 4567/1000+0.5);

  output("h", 4567.0/1000+1/2);

  output("i", 4567/1000.0+1/2.0);

  output("j", 4567/e);

  output("k", 4567-4567/100*100);

}
```

2.

Consider this function:

```
void speckled(const int a, const int b)
{ if (a==b)
    cout << a;
  else if (a<b)
  { cout << b;
    speckled(a+1, b-1);
    cout << a; } }
```

a. What, in plain English, does it do?

b. And what does this one calculate?

```
int hippopotamus(const int m)
{ if (m<=O)
    return O;
  else
  { const int k = hippopotamus(m-1);
    return m*m+k; } }
```

3.

a. `[DO NOT USE ANY VARIABLES]`

Write a C++ function that takes two integer parameters (`a` and `b`) and prints a list of all the integers between `a` and `b-1` (inclusive).

Example:

`one(2, 12)` should print `2 3 4 5 6 7 8 9 10 11`.
`one(2, 13)` should print `2 3 4 5 6 7 8 9 10 11 12`.

b. `[DO NOT USE ANY VARIABLES]`

Write a C++ function that is the same as the first one, except that it only prints out the numbers between `a` and `b-1` that b is exactly divisible by.

Examples:

`two(2, 12)` should print `2 3 4 6`.
`two(2, 13)` should print nothing.

c. `[DO NOT USE ANY VARIABLES]`

Write a C++ function that is very similar to the second one, except that it doesn't print anything, it just returns as its result the number of things that `two` *would* print.

Example:

`three(2, 12)` returns 4, because `two(2, 12)` prints four things.
`three(2, 13)` returns 0, because `two(2, 13)` prints nothing.

Hint:

If this seems a little tricky, first write a function that would return `true` if `two` would print anything at all, and `false` if `two` would print nothing.

d. `[DO NOT USE ANY VARIABLES]`

You are probably aware that a number that has no divisors except for 1 and itself is called a prime number.

Using your previous answers, write a function that takes one integer parameter `n`, and if `n` is prime prints out the word "`prime`", and if `n` is not prime prints out all the divisors of `n`.

Examples:

`four(11)` should print `prime`
`four(12)` should print `2 3 4 6`
`four(13)` should print `prime`
`four(14)` should print `2 7`

4.

Draw a picture of a cat sitting on a table.

Do not write a program that draws the picture, just draw the picture yourself.