

EEN118
Very Tame Test
9th October 2014
Puppies Strictly Forbidden

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: 3 (int)

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

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

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

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

  show("b", 836482817 % 2);

  show("c", 3 + 4 + 5 * 6 + 7 * 2);

  show("d", 37924 / 10 / 10 / 10);

  show("e", 1*2*3*4 / 1*2*3*4);

  show("f", 2+3 * 4+5);

  show("g", 3726 - 3726 / 100 * 100);

  show("h", 3726 - 3726 / 100 * 100.0);

  show("i", 3726 - 3726 / 100.0 * 100);

  show("j", 3726.0 - 3726 / 100 * 100.0);

  show("k", 1 / 2 > 0);

  show("l", 1 / 2 > 0.0);

  show("m", 123 / 10 / 10 * 10 * 10);

  show("n", 37924 / 10 / 10 / 10.0);

  show("o", 37924 / 10 / 10.0 / 10);

  show("p", 654321 / 100 % 100); }
```

2. For this question, only consider positive numbers.

a. Examine these functions:

```
void twoA_worker(const int a)
{ cout << "x";
  if (a > 0)
    twoA_worker(a - 1); }

void twoA(const int N)
{ twoA_worker(N);
  cout << "\n"; }
```

State in plain English exactly what the function `twoA` does, and show (or explain) how you worked that out.

b. Examine these functions:

```
void twoB_worker(const int a)
{ if (a > 0)
  twoB_worker(a - 1);
  cout << "x"; }

void twoB(const int N)
{ twoB_worker(N);
  cout << "\n"; }
```

State in plain English exactly what the function `twoB` does.

c. Examine this function:

```
void twoCee(const int W)
{ if (W > 0)
  { twoA(W);
    twoCee(W - 1);
    twoA(W); } }
```

State in plain English exactly what the function `twoCee` does, and show (or explain) how you worked that out.

3. Do not make use of any variables in your answers to this question.
Use recursive functions to cause repetition when needed.

All numbers will be ints.

- a. Write a function `addup(x, y)` that adds up all the numbers between `x` and `y` (inclusive), and returns their sum as its result.

e.g. `const int a = addup(2, 5);`
defines `a` to be 14 because $2+3+4+5 = 14$.

- b. Write a function `addup_arr(R, x, y)`, where `R` is an array of ints. It should add up all the numbers from position `x` to position `y` in the array, and return their sum as its result.

e.g. `const int b[] = { 7, 3, 1, 6, 14, 2, 9, 5 };`
`cout << addup_arr(b, 2, 5);`
prints 23 because $1+6+14+2 = 23$.

- c. Write another function `sum_arr(R, n)`, where `R` is an array of ints, and `n` is the size of that array. It should add up all the numbers in the array and return the result.

e.g. `cout << sum_arr(b, 8)`
prints 47 because $7+3+1+6+14+2+9+5 = 47$

- d. Write a function `max_arr(R, n)`, where `R` and `n` are as described in part c. It should find the largest number in the array, and return that maximum as its result.

Hint – use an extra parameter to remember the biggest number encountered *so far*.

e.g. `max_arr(b, 8)` is 14

4.

Draw a picture of a happy student who has just passed a test.

It doesn't have to be a good picture.

Look over your answers to make sure that happy student is you.