Maths 1262 quizzes
Answers Wednesday 19/2/14   Your answers should show all work.

(1: 10 marks) Suppose that you took remainder mod 2 of a sequence of values emitted by a random number generator, which may or may not be of linear congruential type. Suppose that it produced the following:

---

0 0 1 1 1 1 1 1 1 1 1 1 1

Is this surprising? What would you say of the chances of the next value also being 1?

Answer. It is surprising. It fits none of the patterns expected for linear congruential generators. The chances of the next value being 1 are evens.

(2: 24 marks) Design a simple 2-vector class, call it `Vec2`.

- Its functions and routines should all be public.

- Its data consists of two `doubles`: `x,y`: the data is also public.

- It has two routines, `void set ( double a, double b )`, which sets x to a and y to b, and `void print()` which prints x and y.

- There is no constructor: `set ()` is used to set the values of x and y.

- It has one function, `double dotprod ( Vec2 other )`, so that `a.dotprod(b)` returns the dot product of two vectors.

- Your main program should initialise two vectors (using set()) and compute and print their dot product.

```cpp
#include <iostream>
using namespace std;

typedef class Vec2
{
public:
  void set ( double a, double b );
  void print();
  double dotprod ( Vec2 other );
private:
  double x, y;
} Vec2;

void Vec2::set ( double a, double b )
{
  x = a; y = b;
}

void Vec2::print ()
{
  cout << "x " << x << " y "<< y << endl;
}

double Vec2::dotprod ( Vec2 other )
{
  return x * other.x + y * other.y;
```

```
}

int main ()
{
  Vec2 a,b;
  a.set (1,2);
  b.set (3,4);
  a.print();
  b.print();
  cout << a.dotprod ( b ) << endl;
}
```

(3: 2 marks each) Carefully evaluate the following expressions, being careful to produce the answer in the correct type (`int, double, char`). Indicate doubles with decimal points and character constants with single quotes.

```
1/2/3 == 0
1/2/3+0.5  == 0.5
1/2.0/3 + 0.5 == 0.666667
1-2-3-4 == -8
5/3*2/2 == 1
5*2/3/2 == 1
(3.2 + 4)/3+3 == 5.4
(char) ('h'-'a'+'A') == 'H'
```