UNIVERSITY OF DUBLIN

X-MA1261-1

TRINITY COLLEGE

Faculty of Engineering, Mathematics and Science

school of mathematics

JS/SS Maths/TSM **Trinity Term 2013**

MA1261/MA1262 — Mathematical Computation I/II

Tuesday, May 7 09.30 – 12.30

Dr. Colm Ó Dúnlaing

Students taking MA1261 attempt 3 questions from section 1.

Students taking MA1262 attempt 3 questions from section 2.

Students taking both modules attempt any 6 questions.

The time allowed for this paper is 3 hours for students registered for both MA1261 and MA1262. For students registered for just one of the modules the time allowed is 2 hours.

SECTION I — MA1261

1. (a) Convert 1234 and -456 to short integer, 2s complement.

   (b) Hence calculate the short integer sum of 1234 and -456.

   (c) Convert $32.0/7$ to single-precision floating point.

2. (a) Write a routine `int length(char *s)` to calculate the length of a string s, without using the `strlen()` function.

   (b) Write a routine `double average( double x[], int n)` to compute the average of the first n elements of the array x. You may assume $n > 0$.

3. (a) Carefully simulate

```
#include <stdio.h>
int xxx ( int n )
{ int i, x, y;
  x = 1;
  y = 2;
  while ( n > 0 )
  {
    if ( n%2 == 1 )
    { x = x * y; }

    y = y*y;
    n = n/2;
  }
  return x;
}
main()
{ printf("xxx(5) is %d\n", xxx(5)); }
```

   (b) What does `xxx(n)` compute, given $n \geq 0$?

(c) Carefully simulate

```
#include <stdio.h>
int yyy ( int n )
{
  int z;
  if ( n == 0 )
    return 1;
  else
  {
    z = yyy ( n/2 );
    if ( n%2 == 0 )
      return z * z;
    else
      return 2 * z * z;
  }
}
main()
{ printf("yyy(5) is %d\n", yyy(5)); }
```

(d) What does `yyy(n)` compute, given $n \geq 0$?

4. (a) What is the mistake in the following code? What happens? How do you correct it?

```
x = 1;
while ( x = 1 )
{ printf("hello\n");
  x = 0;
}
```

(b) What does the following code do?

```
main()
{ int a[3][3];
```

```
  int i, j;
  for ( i=0; i<3; ++i )
  { for ( j=0; j<3; ++j )
    { a[i][j] = (i==j); }
  }
}
```

(c) Evaluate

(i) `1 - 2/3 + 4.0`

(ii) `1 - 2/3.0 + 4`

(iii) `3*4/2`

(iv) `3/2*4`

SECTION II — MA1262

5. There is one mistake in the following program.

```
#include <iostream>
#include <iomanip>
#include <vector>
#include <algorithm>
using namespace std;
int main ()
{ double sum, val, b, c;
  vector <double> v;
  bool finished;
  finished = false;
  while ( ! finished )
  { cin >> val;
    if ( cin.eof () )
      finished = true;
    else
    { v . push_back ( val );
      sum = sum + val;
    }
  }
  a = a/v.size();
  sort ( v.begin(), v.end() );
  b = v[ 0 ]; c = v[ v.size()-1 ];
  cout << b << ' ' << a << ' ' << c << endl;
  return 0;
}
```

(a) What is the mistake and how is it corrected?

(b) With the mistake corrected, what does the program do?

6. The following code behaves differently on different machines.

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

int rand_int( int n )
{ return rand() % n;
}
main()
{ int i, p, q;
  for ( i=0; i<10; ++i)
    cout << rand_int ( 2 ) << ' ';
  cout << ";   ";
  for ( i=0; i<10; ++i )
  { p = rand_int ( 6 );
    q = rand_int ( 6 );
    cout << p+q << ' ';
  }
  cout << endl;
  return 0;
}
```

```
Machine A:
1 0 1 1 1 1 0 0 1 1 ;   3 3 9 0 8 7 6 4 3 2
Machine B:
0 1 0 1 0 1 0 1 0 1 ;   7 9 7 5 5 5 7 1 5 7
```

(a) Explain the different behaviour on the two machines.

(b) Write a function `rand_double()` which uses the function `rand()` to return random double-precision numbers between 0 and 1.

(c) Your answer to the above should use the number `1 + (double) RAND_MAX`. What would be wrong with `1 + RAND_MAX`, and with `(double)( 1 + RAND_MAX)`?

(d) The function `int rand_int ( int n )` was supposed to produce random integer numbers in the range $0 \ldots n-1$, but did not work well on 'Machine B' above. Give a better way of producing random integers between 0 and $n-1$.

7. (a) Write a program which takes a string from the command line, i.e., `argv[1]`, reads input (i.e., `cin`) line-by-line, and prints out those lines containing `argv[1]`.

Note: `cin.getline (buffer, 200)` reads up to 200 input characters from the input, `string::string ( char * s )` constructs a string object from the character string `s`, and `string::find ( char * word )` returns the starting position of the word in the string, default `string::npos`.

(b) Complete the member functions and operators for the following class. Note that $a/b + c/d = (ad+bc)/bd$, and $a/b = c/d \iff ad = bc$.

```
typedef class Fraction
{ public:
    Fraction ( int m, int n );
    int numerator ();
    int denominator ();

    Fraction sum ( Fraction other );
    bool operator == ( Fraction other );
  private:
    int num, denom;
} Fraction;
```

8. Show what gets printed by the following program, explaining why.

```
#include <iostream>
using namespace std;
char mess[] = "hello\n";
void a( int m )
{ cout << mess; }

void b( int mess )
{ cout << mess << endl; }

main()
{
  int c = 99, x = 5;
  a(9); b(9);
  cout << c << " " <<  x << endl;
  for ( int c = 0; c <= x; ++ c )
  { cout << c << " "; }

  cout << endl << c << endl;
  return 0;
}
```