Maths 1261 Quiz 3 answers Friday 1/11/13

(1: 6 marks) (a) What is wrong with the following piece of code? (b) What happens? (c) Write a correct version. Note: in C, an assignment statement has a *value*, namely, the value assigned to the variable.

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

```
(a) while ( finished = 0 )
(b) nothing is printed
(c) substitute while ( finished == 0 )
```

(2: 6 marks) As (1), with the following code.

```
finished = 0;
if ( ! finished )
{
    printf ("hello\n");
};
else
{
    printf ("goodbye\n");
};
```

(a) }; on fifth line(b) Won't compile: unmatched else(c) delete the semicolon on fifth line

(3: 6 marks) As (1), with the following code.

```
finished = 0;
while ( ! finished );
{
    printf("hello\n");
    finished = 1;
}
(a) while ( ! finished );
(b) infinite loop doing nothing
```

```
(c) while ( ! finished ) (delete semicolon)
```

(4: 12 marks) Identify the error in the following program, explain the resulting behaviour, and say how to correct it.

```
Given myprog.c:
#include <stdio.h>
int main()
ſ
 double x = 123;
 printf("x is %f. Enter new value for x.\n", x);
  scanf ("%f", &x );
 printf("x is %f\n", x);
 return 0;
}
Compile and run:
% gcc myprog.c
% a.out
x is 123.000000. Enter new value for x.
45678
x is 123.000017
```

Answer. The scanf() statement uses single-precision format instead of the required %lf. As a result, the single-precision floating-point representation for 45678 is stored in the leftmost 4 bytes of x, leaving the rightmost 4 bytes untouched.

Since the representation is little-endian, only some low-order bits of the mantissa are altered, so the effect on x is rather small.

Correct it by changing the scanf format to %lf.

(5: 20 marks) Given declarations

double x[10], a[30][6]; int c[10][9];

suppose that x begins at memory location 2000, and that a is stored immediately after x and c immediately after a.

(i) Give the number of bytes occupied by x, a, and c respectively.

Answer 80, 1440, 360.

(ii) Deduce the starting addresses of a, c.

Answer. 2080, 3520.

(iii) Calculate the addresses of x[5], a[2][3], and c[9][7].

Answer

x[5]: 2000 + 5 * 8 = 2040

a[2][3]: 2080 + 2 * 6 * 8 + 3 * 8 = 2200 c[9][7]: 3520 + 9 * 9 * 4 + 7 * 4 = 3872

(iv) x[100] is actually located within array a. Calculate i, j such that & (x[100]) == & (a[i][j])

Answer. x[100] = 2000 + 800 = 2800 2080 + i * 6 * 8 + j * 8 = 2800 i * 48 + j * 8 = 720 i = 720/48 = 15 j = 720 % 48 = 0 answer: a[15][0].

•