MA U11601 Quiz 02 22/10/21 ANSWERS

Due after Reading Week.

Answer any 3 questions. Submit them through Blackboard as pdfs, either handwritten and scanned, or typeset. They should be submitted before midnight on Monday 1 November. If more than three answers are submitted, only the first three will be marked. All questions carry 20 marks.

Show all work. That is, where an answer requires some calculation, show the calculation.

Plagiarism. If copying is detected, all those involved will lose credit, irrespective of who copied from whom.

Question 1. Convert 1485 and -22116 to short int, giving the answers as 4 hex digits, big endian. Then add them as short ints giving the answer as 4 hex digits, big endian.

Answer.

```
m 1485: 05 cd
                16)1485=92 r 13, 16)92 = 5 r 12, 16)5 = 0 r 5
        0,5,12,13 = 05cd
n -22116: a9 9c
        16)22116 = 1382 r 4, 16)1382 = 86 r 6, 16)86 = 5 r 6,
        16)5 = 0 r 5  .... 0,5,6,6,4
        ffff - 5664 = a99b + 1 = a99c
m+n -20631: af 69
        05cd + a99c = af69
```

Question 2. Convert 7.0/11 to 32-bit floating point, showing all work, of course. Your answer should be 8 hex digits, little endian (this means that the four bytes are in reverse order, not that the hex digits in the bytes are reversed).

Answer.

```
 7/11 :
true exponent -1, biased  0 1 1 1 1 1 1 0
24 bits for mantissa
 0 1 0 0 0 1 0 1 1 1 0 1 0 0 0 1 0 1 1 1 0 1 0 0
round down
altogether
0: 0 1 1 1 1 1 1 0:  0 1 0 0 0 1 0 1 1 1 0 1 0 0 0 1 0 1 1 1 0 1 0
hexadecimal 3f22e8ba, and little endian ba e8 22 3f
```

Question 3. What does the following code do?

```
#include <stdio.h>
main()
{ int array[6] = {3,1,4,1,5,9};
  int i; int temp;

  for (i=0; i<6; ++i)
  { temp = array[i];
    array[i] = array[5-i];
    array[5-i] = temp;
  }

  for (i=0; i<6; ++i)
  { printf("%d ", array[i]);
  }
  printf("\n");
}
```

Answer. The code repeatedly swaps pairs of array elements. Since every pair gets swapped twice, it ends up same as before.

Question 4. Write a complete program which reads in and counts numbers, storing them in an array x[1000], and outputs the count and the total. You need not check for overflow (more than 1000 numbers). For example,

```
% a.out
3 1 4 1 5 9 2 6 5
(<Ctrl-D> is invisible)
total of 9 numbers is 36.000000
%
```

The sum can be calculated using a for-loop.

Answer.

```
#include <stdio.h>
int main()
{
  double x[1000], xx;
  double sum = 0;
  int count;
  for (count=0; scanf("%lf", &xx ) == 1 && count < 1000; ++count)
  {
    x[count] = xx; sum += xx;
  }

  printf("total of %d numbers is %f\n", count, sum);
}
%a.out
3 1 4 1 5 9
(<Ctrl-D> is invisible)
total of 6 numbers is 23.000000
```

Question 5. Extend the program in question 4 so that it produces the average, and also the variance. The average is

$$\overline{x} = \frac{1}{n}\sum_{i=0}^{n-1} x_i$$

and the variance is

$$\frac{1}{n-1}\sum_{i=0}^{n-1}(x_i - \overline{x})^2$$

Example:

```
% a.out
3 1 4 1 5 9 2 6 5
<Ctrl-D>
average of 9 numbers is 4.000000, variance 6.750000
%
```

Answer.

```
#include <stdio.h>
int main()
{ double x[1000], xx; double sum = 0;
  int count;
  for (count=0; scanf("%lf", &xx ) == 1 && count < 1000; ++count)
  { x[count] = xx; sum += xx;
  }
  double average = sum/count;

  int i; double sq = 0;
  for (i=0; i<count; ++i)
  { sq += (x[i]-average)*(x[i]-average);
  }

  double variance = sq/(count-1);
  printf("average of %d numbers is %f, variance %f\n", count,
        average,variance);
}
% a.out
3 1 4 1 5 9
average of 6 numbers is 3.833333, variance 8.966667
```