UNIVERSITY OF DUBLIN
TRINITY COLLEGE

Faculty of Science

school of mathematics

JF Maths/TP/TSM **Trinity Term 2017**

Mathematics 1234: C, C++ programming

Dr. Colm Ó Dúnlaing

Attempt 3 questions

1. (a) Convert 2017 to short int (4 hex digits, big-endian).

   (b) Convert −18284 to short int.

   (c) Add the above two numbers as short ints.

   (d) Given `int a[10]; double b[10][10];`
   (Recall that ints are 4 bytes and doubles are 8 bytes.) Suppose that `a` begins at location 4159, and `b` begins immediately after `a`.

   i. Calculate the address where `b` starts.

   ii. Calculate `b[3]`.

   iii. Calculate the address (outside the range of `a`) of `a[25]`.

   iv. Does this address coincide with that of any entry `b[i][j]`?

Answer

```
      126 r 1           7 r 14   answer 0,7,14,1
16 ) 2017        16 ) 126               07 e1
     16               112
      41               14
      32
       97
       96
        1

      1142 r 12         71 r 6           4 r 7
16 ) 18284       16 ) 1142        16 ) 71
```

```
 16                112                64
  22                 22                 7
  16                 16
   68                 6      4 7 6 12
   64
    44   18284 -> 47 6c      ffff
    32                     - 476c
    12                       b893
                           +    1
                             b894

Adding    b894
          07e1
          c075

(a) 4159 + 10 * 4 = 4199
(b) 4199 + 10 * 8 * 3 = 4199 + 240 = 4439
(c) 4159 * 4 * 25 = 4259
(d) 4259 % 8 = 3,   4199 % 8 = 7.  No.
```

---

2. (a) Consider the following progam.

```
#include <stdio.h>
#include <stdlib.h>
int xxx ( int m, int n )
{ if ( m == 0 )
    return 0;
  else
  { int r = m % 3;
    int s = xxx ( m/3, n );
    if ( r == 0 )
      return 3 * s;
    else if ( r == 1 )
      return 3 * s + n;
    else
      return 3 * s + n + n;
  }
}
```

```
main ( )
{ int p = xxx ( 5, 10 );
  printf("xxx ( 5, 10 ) == %d\n", p);
}
```

i. Carefully simulate the program, showing what it prints.

ii. What does `xxx (m,n)` return, for general $m \geq 0$?

Answer

```
(i) prints xxx ( 5, 10 ) = 50
(ii) returns m * n
```

(b) Identify 3 errors in the following program (there are more than 3).

```
#include <stdio.h>
main ()
{ int m = atoi ( argv [1] );
  while ( x = m );
  { ++ x; }
  if ( x == 0 )
  { printf("zero\n") };
  else
  { printf("nonzero\n") };
}
```

3. (a) Given the type definition below, write code for the functions `make_zero_matrix()` and `matrix_sum` (which returns `NULL` if the matrices have different sizes).

```
#include <stdlib.h>
typedef struct MATRIX
{ int height, width;
  double ** entry;
} MATRIX;
MATRIX * make_zero_matrix ( int ht, int wdth )
{ ... complete the code here ...  }
MATRIX * matrix_sum ( MATRIX * a, MATRIX * b )
{ ... complete the code here ...  }
```

Answer

```
MATRIX * make_zero_matrix ( int ht, int wdth )
{
```

```
  double * mat = (double *) calloc (1, sizeof ( MATRIX ));
  double *pool = (double *)calloc(ht * wdth, double );
  mat -> height = ht;  mat->width = wdth;
  mat -> entry = ( double ** ) calloc ( ht, sizeof ( double * ) );

  int i;
  for (i=0; i<ht; ++i)
    mat->entry[i] = & ( pool [ i  * wdth ] );

  return mat;
}

MATRIX * matrix_sum ( MATRIX * a, MATRIX * b )
{
  if ( a->height != b->height || a->width != b->width )
    return NULL;

  MATRIX * sum = make_zero_matrix ( a->height, a->width );

  int i,j;
  for (i=0; i<a->height; ++i)
  for (j=0; j<a->width; ++j)
    sum->entry[i][j] = a->entry[i][j] + b->entry[i][j];

  return sum;
}
```

(b) Given the type definition below, write code for (i) the constructor and (ii) the `[ ]` subscripting operator.

```
typedef class Matrix
{ public:
    Matrix ( int the_ht, int the_wdth );
    double * operator [] ( int i );
  private:
    int ht, wdth;
    double * workspace;
} Matrix;
```

Answer

```
Matrix::Matrix ( int m, int n )
```

```
{
  height = m; width = n;
  workspace = new double [ m * n ];
}

double * Matrix:: operator [] ( int i )
{
  return workspace + width * i;
}
```

---

4. (a) Write a complete C program which reads in a list of numbers (doubles) from standard input, stores them in an array and counts them, and prints the count, the sample mean, and the sample variance.

   (As for storing in an array, you may assume that at most 1000 numbers will be input.)

Answer

```
#include <stdio.h>

main ()
{
  double store [1000], x;
  int n = 0;
  double sum, mean, variance;
  sum = 0;
  while ( scanf ( "%lf", & x ) == 1 )
  {
    store[n++] = x;
    sum += x;
  }
  mean = sum/n;

  int i;
  sum = 0;
  for (i=0; i<n; ++i)
  {
    double x = store[i] - mean;
    sum += x*x;
  }
  variance = sum / (n-1);
```

```
  printf("count %d mean %lf variance %f\n",
        n, mean, variance);
}
```

(b) Write a complete C++ program which reads in a list of numbers (doubles) from standard input, storing them in `vector <double>` object, sorts them, and prints the count, the sample mean, the sample variance, and the median. (For this question, by 'median' we mean the item indexed $(n-1)/2$ in sorted order, where n is the count).

Answer

```
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

int main ()
{
  vector <double> vec;
  double x, sum;

  sum = 0;
  while ( cin >> x )
  {
    vec.push_back ( x );
    sum += x;
  }

  int n = vec.size ();

  double mean = sum / n;

  sum = 0;

  for ( int i = 0; i<n; ++i )
  {
    double x = vec[i] - mean;
    sum += x*x;
  }
```

```
  double variance = sum / (n-1);

  sort ( vec.begin(), vec.end() );

  double median = vec[ (n-1) / 2 ];

  cout << "count " << n << " mean " << mean <<
        " variance " << variance << " median " <<
        median << endl;
}
```
