CPSC 3720 – S.E.	**Due Apr. 14** (in class)
Assignment 4	Total marks: 30

1) **(10 pts)** You are given the following specification of a C function. FYI, parameters $X1$ and $X2$ are pointers of double type. In C, the value `NULL` can be used with any pointer type.

```
int roots(int a, int b, int c, double* X1, double* X2);
```

This function computes the roots of the quadratic equation $ax^2 + bx + c = 0$. The roots are returned in parameters $X1$ and $X2$. The return value is *true* on success and *false* on failure, for example if the equation does not admit real solutions.

(a) Are there any problems with this specification? Explain.

Solution:

Failure is not clearly defined and the behaviour of the return parameters is not specified. Are the return parameters unchanged if failure ocurs? When both roots are equal, is this considered failure? What is returned when the roots are equal, both return parameters are updated, only one? What is the behaviour if the return parameters are passed NULL values? What is the behaviour when a==0? Is this failure? Are the roots computed? If yes, which of the two parameters will contain the root and what happens with the other parameter? Is a=b=c=0 considered failure? What about $a = b = 0$ and $c \neq 0$?

(b) Do a domain analysis on the parameters of the function, and write test cases. State any assumptions that you are making about the function. Use a table with the following header:

Valid range/combination	Invalid range/combination	Test values	Expected outcome

Solution:

From the previous answer, you can guess the answer to this question too. Valid/invalid ranges are determined by conditions such as $a \neq 0$, and $b^2 - 4ac \geq 0$. For test cases you should give some reasonable combination of valid/invalid ranges for a, b, c You should also provide test cases for the return parameters with NULL values and valid variable addresses. You don't have to cover all combinations exhaustively, but you should include the cases: two different real roots, two identical real roots, no real roots, linear equation, $a = b = 0$ and $c \neq 0$, $a = b = c = 0$, and perhaps two/three more such as: two different real roots and one NULL return parameter, and linear function and both (or only one) NULL return parameters.

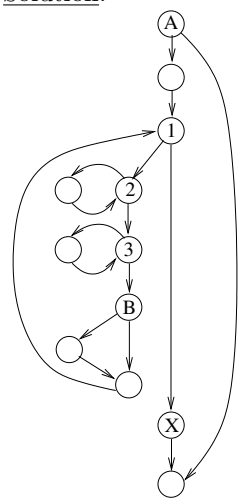
2) **(20 pts)** You are given the following quicksort routine written in C.

```
void q_sort(int array[], int left, int right)
{
  if (left >= right)
    return;

  int pivot = array[left];
  int min = left, max = right;

  while (min < max)
  {
    while (array[min] <= pivot && min < max)
      min++;

    while (array[max] > pivot && min < max)
      max--;

    if (min < max)
      swap(&(array[min]), &(array[max]));
  }
```

```
  q_sort(array, left, max-1);
  q_sort(array, max, right);
}


int quicksort(int array[], int dim)
{
  q_sort(array, 0, dim-1);
  return 1;
}
```

(a) How many test cases are there for the path covering strategy? Explain your answer.

Solution:



The control flow graph is to the left, where A, B are the two `if` statements, $1, 2, 3$ are the three `while` statements, and x is the pair of recursive calls.
In the *body* of the first `while` there are 2^3 different paths. If you include the first `while` with its condition testing, then you get 2^3+1 paths. Including the first `if` statement gives $2^3 + 2 = 10$ paths.

(b) Assume you are testing this code using an aray of 4 elements. List the test cases you would use. Explain your strategy in choosing the test cases.

NOTE: Rely on your intuition. You do NOT have to follow any of the strategies we learned in class.

Solution:

I would try values sorted increasingly, sorted decreasingly, different values unordered, $\{4, 2, 4, 1\}$, $\{4, 1, 1, 3\}$, and invalid values for parameter `dim`.

(c) Code inspections are very efficient in finding faults. Examine the code without compiling it. Can you find any faults? Explain and list them, if any.

Solution:

At least one fault occurs when one of the partitions is empty, for example if the pivot is the largest value.