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.

   (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

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

```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.

   (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 <u>NOT</u> have to follow any of the strategies we learned in class.

   (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.