2.

A.

A person object has a social security number, a name, and a phone number. Give a complete C++ implementation of quicksort or mergesort that will sort an array of N pointers to person objects so that they appear with their social security numbers in ascending order.

put your answers to b and c in human-oriented terms.

B.

If it takes selection sort or bubble sort 0.1 S to sort an array of 1,000 items, approximately how long would it take ...

i. selection sort to sort an array of 1,000,000 items ?
ii. mergesort to sort the original array of 1,000 items ?
iii. mergesort to sort an array of 1,000,000 items ?

Explain very briefly how you reached your answers.

C.

If it takes mergesort 1 S to sort an array of 1,000,000 items, approximately how long would it take ...

i. selection sort to sort the same array ?
ii. mergesort to sort an array of 2,000,000 items ?
iii. mergesort to sort an array of 1,000 items ?

Explain very briefly how you reached your answers.

3.

A.

A binary (non-text) file contains a large number of groups of doubles. The groups are of various lengths, and the number of groups is unknown.

Immediately before each group of doubles is a 32 bit integer giving the number of doubles in the group. The file ends directly after the last group.

The input file is to be processed to create a second binary file. The output file should contain two doubles for each group in the input file, these doubles should be the minimum and maximum value found within the corresponding group. If a group is empty, then both should be zero.

Write a program using only C++ library functions and functions that you write yourself, that prompts the user for the name of the input and output files, and performs the processing described above. Speed of execution is very important.

B.

Write a function that counts the number of zeros in the binary representation of an integer, using bit-wise operations.

4.

I want a large two dimensional array of numbers, with 300 rows and 800 columns. It must be declared exactly like this:

```
double * * C;
```

Write a program that does whatever is required to allocate memory for the array, and fill it with data according to this engineering-related formula

$$C[x][y] = \frac{(x-150)^2 + 3x - y}{200y + 1}$$

Store all of those numbers in a disc file in a very efficient way. There must be no possibility of any loss of precision in the data, and there must be no wasted space (the resultant file must be as small as is reasonably possible). Make sure your solution would work under both unix and windows. No text files.

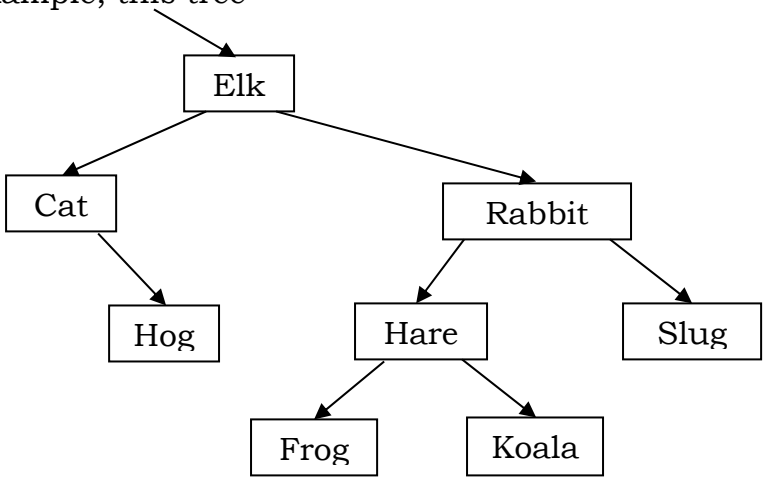
Before exitting, the program must fully de-allocate all memory used for the array.

Write a second program that scans the file you just created to find the average of all the numbers that appear in it. Your answer must be as fast as possible, with the restriction that it must not use more than one quarter of a megabyte of memory for data storage.

5.

A binary tree containing strings that contain only letters may be reduced to a single line of text by printing it with a post-order traversal. All the NULLs in the tree are printed as dashes, and all the dashes and strings are separated by spaces. A star is added to make the end of the line visible.

For example, this tree



is reduced to this text:

- - - Hog Cat **- -** Frog **- -** Koala Hare **- -** Slug Rabbit Elk *****

Write a function (or functions) that will read from `cin` the line of text representing the post-order traversal of a tree, then print out the pre-order traversal of that tree.

You will most probably find it easier if your function actually builds the tree as an intermediate step.

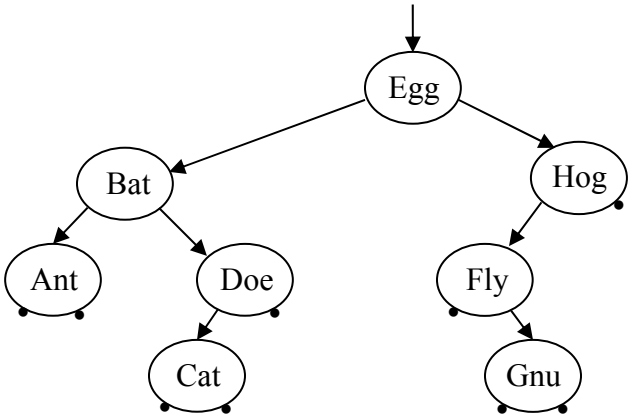
If it is impossible to construct a tree from the input provided, your program should not crash. It should print “error” and stop.

6.

A binary tree has been implemented with nodes defined thus:

```
struct node
{ string data;
  node * left, * right; };
```

Making use of your answer to question 3, write a function that will print all of the strings in a tree, in breadth-first order.



Given this tree, your function would print

Egg, Bat, Hog, Ant, Doe, Fly, Cat, Gnu

in exactly that order.

7.

A file contains a list of roads, one per line, in this format:

A R C

where

A is the name of a town
R is the name of a road
C is the name of another town, not the same as A.
A, R, and C are simple strings and they are separated by a space.
No two towns or roads will have the same name.

The only way you know that a town exists is that it appears in one of more of these lines. The same applies to roads.

Design and implement software that will read a file in this format, then find the optimum route between two towns whose names are provided by the user.

The optimum route is the one that has the smallest number of different roads: the vehicle has trouble with corners.

The output should be the list of roads to follow (in order), or “NO WAY” if no route can be found.

8.

Implement in C++ the Floyd-Warshall shortest path length algorithm (the one with the two dimensional array).

Explain how the actual shortest path (as a sequence of nodes to travel to) can be found once Floyd-Warshall has found the lengths of all shortest paths.

9.

A special kind of ordered tree, called a *three-tree*, is very similar to a binary tree but not identical. Every node in a three-tree has two data items (called `A` and `B`), and three pointers to other nodes (called `left`, `middle`, and `right`).

```
struct node
{ string A, B;
  node *left, *middle, *right; };
```

Just as with a binary tree, any of the pointers may be `NULL`, and there are strict rules for ordering the data:

- In any given node, `A` must be less than `B`;
- All data reachable through the `left` pointer must be `< A`;
- All data reachable through `middle` must be `> A` but `< B`;
- All data reachable through `right` must be `> B`.

This is the standard function for printing a whole three-tree in order:

```
void printall(node * p)
 { if (p == NULL)
      return;
   cout << "(";
   printall(p->left);
   cout << "-" << p->A;
   printall(p->middle);
   cout << "-" << p->B;
   printall(p->right);
   cout << ")"; }
```

Write a non-recursive function that has exactly the same effect as that function.

10.

Draw a picture of a cat sitting on a table. She does not like virtual methods, although she is not quite sure what they are.