

Answer *THREE* questions, including question 1 or question 2 (or both).

1. a. Design a C++ function called `reverse_words` that takes a string and reverses the individual words in it. You may assume that a word is defined as any sequence of characters (excluding the special character `'\0'`) not containing a space. For example, if `string` is a string containing "This is a special string", then after using the function as follows:

```
reverse_words(string);
```

`string` should contain `sihT si a laiceps gnirts`". You may make use of the usual functions `strlen` and `strcpy` but you may not assume that any other functions are available. Your answer should include a full design, and properly commented C++.

[28 marks]

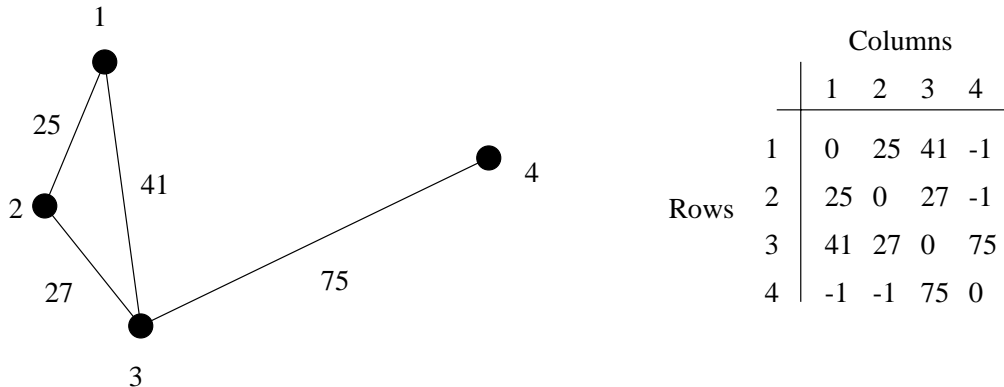
- b. Briefly explain, giving examples where necessary, how you would test the function designed in the first part of the question.

[5 marks]

[Total 33 marks]

TURN OVER

2. This question involves writing a C++ class to represent simple maps. A map is a finite collection of cities identified by consecutive numbers, starting with 1, along with the distances between cities that are connected. A map of this kind can be represented using a grid of numbers as follows:



On the left are four cities, labelled from 1 to 4, with the distances between them marked. On the right is a grid representing the same map. The grid has numbers of rows and columns equal to the number of cities, and the distance between any pair of cities a and b appears in row a , column b . All cities are distance 0 from themselves, and if cities are not connected then their distance is -1 .

The following is a basic class called `Map` which can store a map for any number of cities using this idea. The `grid` pointer points to the grid of numbers, which is stored using a separate array for each row. The variable `cities` stores the number of cities.

```
class Map
{
private:
    int ** grid;
    int cities;
public:
    Map(int = 1);
    ~Map();
};
```

CONTINUED

a. The constructor for the Map class is as follows.

```
Map::Map(int number)
{
    cities = number;
    if (cities < 2)
    {
        cout << "There must be at least 2 cities" << endl;
        cities = 2;
    }
    grid = new int * [cities];
    for (int i = 0; i < cities; i++)
    {
        grid[i] = new int [cities];
        for (int j = 0; j < cities; j++)
            if (i == j)
                *(grid[i] + j) = 0;
            else
                *(grid[i] + j) = -1;
    }
}
```

Explain in detail how this constructor works. Draw a diagram similar to that given above, showing the map produced by the single line of C++,

```
Map a_map (4);
```

[8 marks]

TURN OVER

- b. The destructor for the Map class is as follows.

```
Map::~~Map()  
{  
    for (int i = 0; i < cities; i++)  
        delete [] grid[i];  
    delete [] grid;  
}
```

Explain in detail how this destructor works.

[3 marks]

- c. Write a new member function called `set_distance` for the class Map. The function should take two numbers specifying two cities, and a number specifying the distance between them, and update the grid so that the distance is stored correctly. The function should return 1 if the update is successful and 0 otherwise.

[15 marks]

- d. Design an overload of `operator<<` for the Map class which allows us to print out the distance between each pair of cities that is connected. The overload should work in such a way that the distance between two cities is only printed once. For example, if the grid in the diagram above is printed we might obtain:

```
Cities 1 and 2: distance = 25  
Cities 1 and 3: distance = 41  
Cities 2 and 3: distance = 27  
Cities 3 and 4: distance = 75
```

[7 marks]

[Total 33 marks]

CONTINUED

3. a. Explains what it means to say that a function is “recursive”. Give two properties of a recursive function that are required in general for such a function to operate correctly.

[3 marks]

- b. Explain what the following recursive function does, and how it works.

```
int recursive (char * string, char character)
{
    if (string[0] == '\0')
        return 0;
    else
    {
        if (string[0] == character)
            return 1;
        else
        {
            string = string + 1;
            return recursive(string, character);
        }
    }
}
```

[7 marks]

- c. Explain in detail how the function `recursive` operates when called using,

```
letter_search("Hippo", 'p');
```

[3 marks]

- d. Describe in detail, including a pseudo-code description, the operation of the binary search algorithm.

[10 marks]

TURN OVER

- e. Describe in detail, illustrating your answer with suitable examples, the operation of the mergesort algorithm.

[10 marks]

[Total 33 marks]

4. Write an essay on the subject of pointers and their use in C++ programming, included their use in the implementation of data structures. Your essay should be illustrated with examples of C++ code.

[Total 33 marks]

5. a. Briefly describe each of the following features of C++ and explain how they are used, including examples of C++ code where necessary: operator overloading, references, classes, destructors, virtual functions.

[25 marks]

- b. Explain, including examples, the distinction between the `public` and `private` members of a class. Explain how, in general, it is possible to decide whether a member should be `public` or `private`.

[8 marks]

[Total 33 marks]

END OF PAPER