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

1. a. Design a C++ function called `longest_word` that takes a string and its length and returns the length of the longest word in the string. You may assume that a word is defined as any sequence of characters (excluding the special character `'\0'`) not containing a space. For example,

```
cout << longest_word("the longest word is longest", 27);
```

should print out a 7, as the word “longest” has 7 characters and is longer than “the”, “word” or “is”. You may *not* assume that any other functions, such as `strlen` *etc*, are available. Your answer should include a full design, and properly commented C++.

[18 marks]

b. Explain how you would test the function designed in the first part of the question.

[10 marks]

c. The function you have designed can be used to find the length of the longest word in a string. Explain briefly how you would modify it, so that it can be used to find the number of words in the string that have a length lying between two specified numbers.

[5 marks]

[Total 33 marks]

2. a. The following class, called `Int_box`, can be used for storing the individual integers in a linked list of integers. For simplicity, all members of `Int_box` have been made public.

```
class Int_box
{
public:
  int stored_int;
  Int_box * next;
  Int_box(int = 0, Int_box * = 0);
  ~Int_box();
};

Int_box::Int_box(int new_int, Int_box * new_next)
  : stored_int(new_int), next(new_next)
{
}

Int_box::~Int_box()
{
  delete next;
}
```

The class `Int_box` is used in conjunction with the following class, called `Store`.

```
class Store
{
private:
  Int_box ** array;
  int size;
public:
  Store(int = 1);
  ~Store();
  void add(int, int);
```

CONTINUED

```
};

Store::Store(int new_size)
  : size(new_size)
{
  array = new Int_box * [size];
  for (int i = 0; i < size; i++)
    array[i] = 0;
}

Store::~Store()
{
  for (int i = 0; i < size; i++)
    delete array[i];
  delete [] array;
}

void Store::add(int position, int value)
{
  array[position] = new Int_box (value, array[position]);
}
```

The constructor for `Store` has a single parameter, which denotes the size of an array. The constructor operates as follows. It makes the private member `size` equal to the parameter passed, and it makes the private member `array` point to an array of the specified size. Each of the elements in this array is a pointer to an `Int_box`, and is initially set by the constructor to be a nil pointer. The purpose of each of the pointers in the array is to be the head for a linked list of integers. The member function `add` adds a specified integer to a specified linked list.

Explain, in as much detail as you can, how the member function `add` works.

[5 marks]

b. Explain, in as much detail as you can, how the destructor for `Store` works.

[4 marks]

c. Design a further member function for the class `Store` called `remove`. This member function should have a single integer parameter specifying a particular linked list, and should return the integer most recently stored in that list. It should also completely remove this integer from the list. (If the same integer appears in the list several times, only the most recent needs to be removed.) Your answer should include fully commented C++.

[10 marks]

d. Design an overload of `operator<<` for the class `Store`. Your answer should include fully commented C++.

[10 marks]

e. Comment on how the initial classes `Int_box` and `Store` could be improved.

[4 marks]

[Total 33 marks]

3. a. Explain what it means to say that a function is “recursive”.

[1 marks]

b. Describe the operation of the quicksort algorithm, and compare its performance with that of the bubblesort algorithm.

[10 marks]

c. Consider the following C++ function.

```
int function (char * string, int length)
{
  if (length > 1)
    {
      if (string[0] == string[length-1])
        return function(++string, (length - 2));
      else
        return 0;
    }
  else return 1;
}
```

Explain what this function does, and how it works. Give a detailed explanation of what happens when the function is called using `function("abcdeba", 7);`

[16 marks]

d. Explain, stating any assumptions that you make, how the function could be modified so that:

1. where at present it returns 1, it instead returns -1.
2. where at present it returns 0, it instead returns the number of times that the function called itself.

[6 marks]

[Total 33 marks]

4. a. Describe the various kinds of documentation that are generally produced as part of the software development process.

[16 marks]

b. Explain what is meant by the term *abstract data type*, and discuss the way in which C++ supports abstract data types.

[8 marks]

c. Explain the difference between passing a parameter to a function by value, and passing a parameter to a function by reference. Explain in detail how passing by reference can sometimes be used to speed up a program. Give an example of when passing by reference is essential and explain why it is essential in this case.

[9 marks]

[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: dynamic allocation of memory, default values for function parameters, overloading, constant member functions, friends.

[25 marks]

b. Explain in detail why it is sometimes necessary, when designing a C++ class, to include a copy constructor and an overload of `operator=`.

[8 marks]

[Total 33 marks]

END OF PAPER