Q1. Write a function

```
string cipher(string s, int n) { . . . }
```

which takes a C++ string and encrypts it by shifting each character value by the number n, for example "abcd" shifted by 2 becomes "cdef". So your function would return the string "cdef" if called with ciper("abcd", 2). Test your method by encrypting a string (shift by n) and then decrypting it (shift by −n).

What do you think the following message says?

```
[]qr|)r|)j)wx})|x)|nl{n})vn||jpn*
```

Q2. Write a C++ function:

```
string prefixMatch(const string & a, const string & b) { . . . }
```

which returns only the matching characters at the beginning of the C++ strings a and b. For example, the strings "program" and "progress" have a prefix match "progr".

Note: Use only the string method length() and the string indexing operator [], no other string library methods.

Q3. Write a C++ function

```
char frequentChar(const string & s) { . . . }
```

which takes a C++ string and returns the most frequently occuring character in the string, but not counting spaces. For example, the most frequently occurring character in the string "This string is a test" is 's'.

Note: Use only the string method length() and the string indexing operator [], no other string library methods.

Q4. Write a C++ class called Circle. A circle has an x and y coordinate and a radius as private data members. It has a constructor which initializes a new circle with x, y and radius values. Add a method to your Circle class called contains. The contains method takes a given Circle object as a parameter and returns *true* if the circle completely contains the given circle, and returns *false* otherwise.

Q5. Write a C++ class called Song, which has the following features:

- An artist and title data member, stored as C++ strings
- A playCounter data member that keeps count of the number of times the song has been played
- A play() method, which simply increments the play counter of the song
- A constructor which allows a Song to be initialized with a song title and an artist name and also sets the play counter of the new object to 0.

Q6. Add a copy constructor to your Song class of Question 5. Your copy constructor should copy the song title and artist name from the source object, but should set the play counter in the new object to 0. Write your method outside of the class body. Submit only your copy constructor method code.

Q7. Add a static member function, called mostPopular, to your Song class. Your method should take an array of Song objects and return the song in the array that has been played the most times. Return the song from your method as a constant reference to Song. So, your method should not copy any Song objects when it is called. You can check this by putting a printout line in your Song copy constructor.

Submit your complete Song class code below.

Q8. Write a C++ class called SalesItem. A sales item has a description (a string) and a price. Your class should also have the following features:

- A constructor which takes a description string and a price value, to initialize a new SalesItem object with.
- A static data member VATRate, which should be initially set to the value 0.23 (23%).
- A static member function called setVATRate which can be called to change the current VATRate value.
- A print method which prints out the item description and its price.
- A getPriceWithVAT method which returns the price of the item after VAT has been applied. (The price data member stores the net price - price before VAT).

Q9. Write a simple class of your own design, which demonstrates one use of the this pointer.

Q10. Complete the following code, as per the instructions:

```
class Test { private: int i; public: Test(int _i) : i(_i) {} int get() { return i; } };

int main() {
    Test* arr[10];
    for (int i=0; i<10; i++)
       // add a Test object for each array element
    for (int i=0; i<10; i++)
       // call get() on each object and print out the value
}
```

In the first for-loop, write a statement that adds a Test object at each array element, creating each object using the new keyword.

In the second for-loop, write a statement that calls the get() method on each Test object and prints out the value.

Does this programme "leak memory"? Briefly explain your answer.