Lab 7 Inheritance and Pointer (1)

Question-1. Inheritance

Design two base classes called "Shape" and "PaintCost". A new class "Rectangle" inherits from these two base classes. When width and height of a rectangle are provided, the area, perimeter and paint cost will be computed.
The detailed designs are as follows:

a) The class "Shape" has two protected members, namely width (type: int) and height (type int). The class "Shape" has a public member function "set" for setting the width and height.
b) The class "PaintCost" has a public function "getCost" which takes the area (type: int) as the input and output the paint cost (type: int). The paint cost is 231×area.
c) The class "Rectangle" inherits from "Shape" and "PaintCost". It has two public functions "getArea" and "getPerimeter". The outputs are area and perimeter respectively.

Hint-1. All calculations are within integer field.
Hint-2. Write codes in the main() function for implementation.

Expected Outcomes

Example

```
Enter width:
3
Enter height:
12
Area is: 36
Perimeter is: 30
Total paint cost is: 8316
```

Question-2. Pointer

[Will be marked. Your solution needs to pass ALL the E-QUIZ test cases of this question to get the mark for this lab; Otherwise, there is no mark for this lab.]

Design a function to called "stringCompare" to implement the comparison among strings. The rules for string comparison between string s1 and s2 are the followings:

1. Compare s1 and s2 bit by bit from the beginning of both strings.
2. If the i-1 bits of the two strings are identical, then:
   a. If ith bit of s1 is larger than s2, s1 is larger than s2 regardless the remaining part;
   b. If ith bit of s1 is smaller than s2, s1 is smaller than s2 regardless the remaining part;
   c. If ith bit of s1 is equal to s2, continue the operation on i+1th bit until one string ends;
3. When one string ends, and there is still no result for bitwise comparison, the string with longer length is larger. If the lengths of the two string are identical, the two strings are equal.

The function returns 1 if s1 is larger than s2, -1 if s1 is smaller than s2 and 0 if they are identical. Use the function you designed to compare the two input strings.

Expected Outcomes

Example 1
Enter the first string: *qwert* *Enter the second string:* *qwer* *The first string is larger.*
Example 2
Enter the first string: *Qwer* *Enter the second string:* *awer* *The second string is larger.*
Example 3
Enter the first string: *Qwer* *Enter the second string:* *Qwer* *The two strings are equal.*

Question-3. Pointer

After the implementation of the stringCompare, use it to perform bubble sort on the input strings. The input may first include an integer number(less than 15) to indicate how many strings there are for input. Then followed by all the input strings.

Hints:

For easier copy in bubble sort, you can try to use pointer to store all the strings.

Expected Outcomes

Example
Enter the number of the strings: *5* *Enter the contents of each string:* *Qwer* *qwer* *qwe* *qwexty* *asdf* *The sorted strings are:* *Qwer* *asdf* *qwe* *qwer* *qwexty*