

1.

The area of a rectangle can be computed using the formula

$$\text{Area} = b \times l$$

where b and l are the breadth and length of the sides of the rectangle.

Perimeter s is given by $S = 2(b + l)$

Write a void function that uses four parameters.

Two value parameters that provide the length and breadth of the rectangle, and two reference parameters that compute the area and perimeter of the rectangle.

```
Enter length & Breadth of the rectangle :5
2
Rectangle Length :5 Breadth :2 Area:10 Perimeter :14
```

2.

You would like to know how fast you can run in Kilometers Per Hour.

Your treadmill will tell you your speed in terms of a pace

(Minutes and Seconds per mile, e.g. a "5:30 mile").

Write an overloaded function called "ConvertToKPH".

The first definition should take as input two integers that represent the pace in Minutes and Seconds per mile and return the speed in KPH as a double. The second definition should take as input one double that represents the speed in MPH and return the speed in KPH as a double.

One mile is approximately 1.61 kilometers. Write a driver program to test your function.

```
5:30 pace is 17.5636 KPH.
7:30 pace is 12.88 KPH.
8:00 pace is 12.075 KPH.
10 mph is 16.1 KPH.
20 mph is 32.2 KPH.
120 mph is 193.2 KPH.
```

3.

Consider a text file named `scores.txt` that contains player scores for a game. A possible sample is shown below where Ronaldo's best score is 10400, Didier's best score is 9800, etc.

Write a function named `getHighScore` that takes a string reference parameter and an integer reference parameter. The function should scan through the file and set the reference parameters to the name of the player with the highest score and the corresponding score.

```
The high score is 12300 by Pele
```

4.

Problem: to write (and test) a function that has a partially filled array of characters as a formal parameter. A partially filled array requires two parameters, the array itself and a size parameter. The function deletes all vowels from the array, and close up the 'empty' positions, then decrease the size parameter. Test this function.

The main program generates test data.

****TEST STRING:

(1):Mary had a little lamb. its fleece was white as snow.

(2):Now is the time for all good men to come to the aid of the country.

```
Mary had a little lamb. its fleece was white as snow.
size = 52
reduced array = Mry hd  lttl lmb. ts flc ws wht s snw.....
reduced array size = 37
Now is the time for all good men to come to the aid of the country.
size = 66
reduced array = Nw s th tm fr ll gd mn t cm t th d f th cntry.....
...
reduced array size = 45
```

5.

You need to implement a program to calculate the sum of two big number. The number may be 20 digits or less.

Input and output as follow the picture.

```
Large integer summing program
Please enter an integer, 20 digits or less
503333
you entered
          503333
Enter another integer, 20 digits or less
6213
you entered
          6213

The sum is:
          503333
           6213
-----
          509546
```

```
Large integer summing program
Please enter an integer, 20 digits or less
123456789987654321123
Input number size too large. Aborting.
```

```
Large integer summing program
Please enter an integer, 20 digits or less
99999999999999999999
you entered
 99999999999999999999
Enter another integer, 20 digits or less
99999999999999999999
you entered
 99999999999999999999

The sum is:
 99999999999999999999
 99999999999999999999
-----
An overflow has occurred. Result is set to 0
0
```

6.

You have collected reviews from four movie reviewers where the reviewers are numbered 0-3. Each reviewer has rated six movies where the movies are numbered 100-105. The ratings range from 1 (terrible) to 5 (excellent).

The reviews are shown in the following table:

	100	101	102	103	104	105
0	3	1	5	2	1	5
1	4	2	1	4	2	4
2	3	1	2	4	4	1
3	5	1	4	2	4	2

Write a program that stores this data using a 2D array. Based on this information your program should allow the user to enter ratings for any three movies. The program should then find the reviewer whose ratings most closely match the ratings input by the user. It should then predict the user's interest in the other movies by outputting the ratings by the reviewer for the movies that were not rated by the user. Use the Cartesian distance as the metric to determine how close the reviewer's movie ratings are to the ratings input by the user. This technique is a simple version of the **nearest neighbor** classification algorithm.

For example, if the user inputs a rating of 5 for movie 102, 2 for movie 104, and 5 for movie 105, then the closest match is reviewer 0 with a distance of $\sqrt{(5-5)^2 + (2-1)^2 + (5-5)^2} = 1$. The program would then predict a rating of 3 for movie 100, a rating of 1 for movie 101, and a rating of 2 for movie 103.

Note: To store the data in a 2D array the movie ID's must be mapped to 0-5. The solution below stores -1 to represent unrated movies.

```
Enter a movie to rate (100-105). Enter 0 to exit and get recommendations.
102
Enter rating (1-5) for this movie.
5
Enter a movie to rate (100-105). Enter 0 to exit and get recommendations.
104
Enter rating (1-5) for this movie.
2
Enter a movie to rate (100-105). Enter 0 to exit and get recommendations.
105
Enter rating (1-5) for this movie.
5
Enter a movie to rate (100-105). Enter 0 to exit and get recommendations.
0
The closest reviewer is number 0
Predictions for movies you have not yet seen:
Movie 100 : Predicted Rating = 3
Movie 101 : Predicted Rating = 1
Movie 103 : Predicted Rating = 2
```

注意:OUTPUT僅供參考