C Programming Language II — Midterm

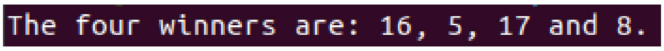
[Notes] Rules for the file names to be submitted :

- .cpp file : [Question number].cpp (for example: the first question is 1.cpp)
- Execution file : [Question number] (example : 1)
- Place all files (include your makefile) in the folder of your student ID (example : B1130400XX)
- Compress your folder to [Student ID].ZIP
- **Make sure that your makefile can successfully compiled, or you'll get zero point.**
- **You are only allowed to use Ubuntu and text editors in Ubuntu in the exam. Otherwise, 50% of deduction of your grade will be applied in any usage of other OS and its applications.**

1. (15%) You have four identical prizes to five away and a pool of 25 finalists. The finalists are assigned numbers from 1 to 25. Write a program to randomly select the numbers of 4 finalists to receiver a prize. Make sure not to pick the same number twice. For example, picking finalists 3, 15, 22 and 14 would be valid but picking 3, 3, 31, and 17 would be invalid, because finalist number 3 is listed twice and 31 is not a valid finalist number.

```
<Hint>
#include <ctime>
srand(time(NULL);
rand(......);
```

Sample :



```
The four winners are: 16, 5, 17 and 8.
```

2. (15%) You have a voucher for a bakery, which you redeem to buy baked goods of your choice. The value of the voucher is fixed and the total bill of the items selected should not exceed the voucher value. If the voucher has any value left after the purchase, then it cannot be utilized again. Write a program that displays 4 items and their prices as a menu and then reads the value of voucher.

The price of each item :

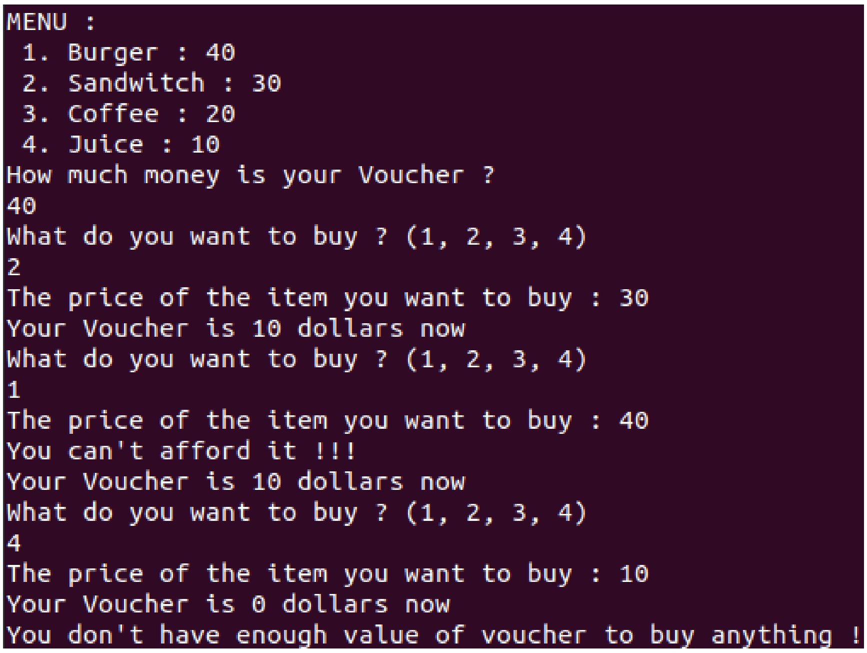
Items	Prices
Burger	40
Sandwich	30
Coffee	20
Juice	10

Write a function, with parameters *itemCode* and *voucherValue* passed by reference, that deducts the price of an item from the voucher's value only if the item's value is less that or equal to the voucher value. Use a loop to invoke the above function as many times as required until the value of the voucher is less than or equal to the cheapest item on the menu. Finally, display the calculated sum of the prices of the items selected, and the unutilized balance on the voucher after purchase.

< Hints >

Function with parameters , and passed by reference : Void buy(int itemCode, int& voucherValus)

Sample :



```
MENU :
 1. Burger : 40
 2. Sandwitch : 30
 3. Coffee : 20
 4. Juice : 10
How much money is your Voucher ?
40
What do you want to buy ? (1, 2, 3, 4)
2
The price of the item you want to buy : 30
Your Voucher is 10 dollars now
What do you want to buy ? (1, 2, 3, 4)
1
The price of the item you want to buy : 40
You can't afford it !!!
Your Voucher is 10 dollars now
What do you want to buy ? (1, 2, 3, 4)
4
The price of the item you want to buy : 10
Your Voucher is 0 dollars now
You don't have enough value of voucher to buy anything !
```

The total price of the items selected is 40 dollars.

Your unutilized balance is 0 dollars.

3. (20%) 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:

Movies Reviewers	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.

< Hints >

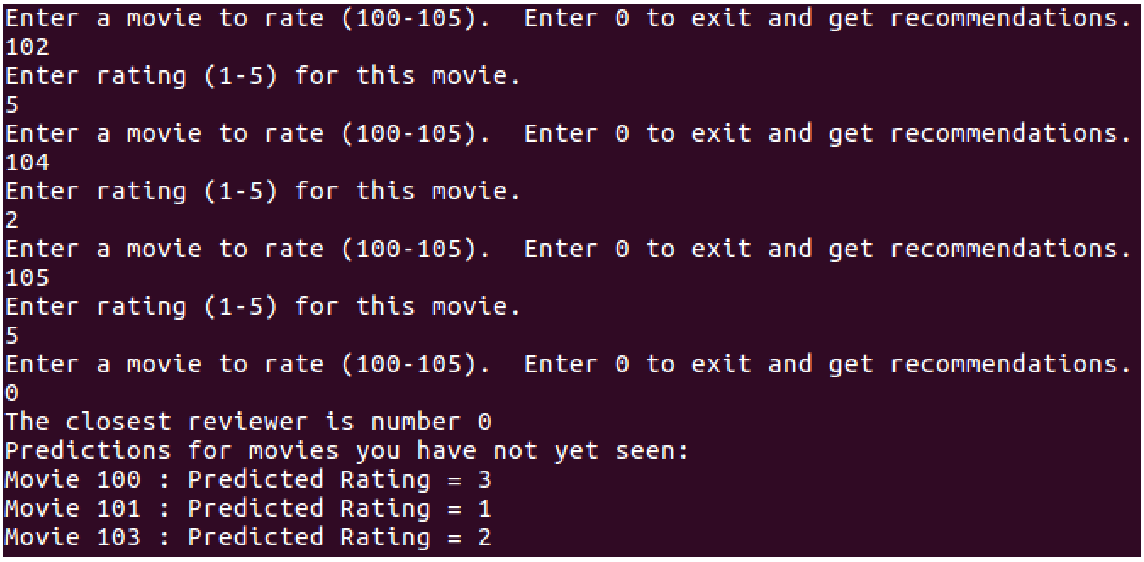
Example : if input movie102 is 5 , movie104 is 2 , movie105 is 5

The most closely match is reviewer0 , and Cartesian distance is : sqrt((5-5)^2+(2-1)^2+(5-5)^2) =1

Then, the ouput will show movie100,101,103 and their ratings.

# include <cmath>

Sample :



```
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
```

4. (25%) Define a class called Pizza that has member variables to track the type of pizza (either deep dish, hand tossed, or pan) along with the size (either small, medium, or large) and the number of pepperoni or cheese toppings. You can use constants to represent the type and size. Include mutator and accessor functions for your class. Create a void function, outputDescription( ), that outputs a textual description of the pizza object. Also include a function, computePrice( ), that computes the cost of the pizza and returns it as a double according to the rules:

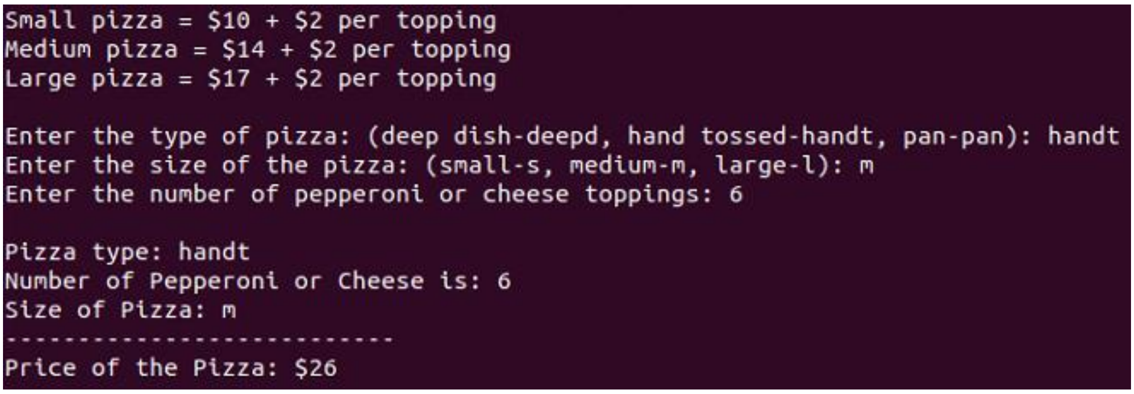
   Small pizza = $10 + $2 per topping

   Medium pizza = $14 + $2 per topping

   Large pizza = $17 + $2 per topping

   Write a suitable test program that creates and outputs a description and price of various pizza objects

Sample :



```
Small pizza = $10 + $2 per topping
Medium pizza = $14 + $2 per topping
Large pizza = $17 + $2 per topping

Enter the type of pizza: (deep dish-deepd, hand tossed-handt, pan-pan): handt
Enter the size of the pizza: (small-s, medium-m, large-l): m
Enter the number of pepperoni or cheese toppings: 6

Pizza type: handt
Number of Pepperoni or Cheese is: 6
Size of Pizza: m
--------------------------
Price of the Pizza: $26
```

5. (25%) In an ancient land, the beautiful princess Eve had many suitors. She decided on the following procedure to determine which suitor she would marry. First, all of the suitors would be lined up one after the other and assigned numbers. The first suitor would be number 1, the second number 2, and soon up to the last suitor , number n.. Starting at the first suitor she would then count three suitors down the line (because of the three letters in her name) and the third suitor would be eliminated from Winning her hand an removed from the line. Eve would then continue counting three more suitors, eliminating every third suitor. When she reached the end of the line she would continue counting from the beginning. For example, if. there were six suitors then the elimination process would proceed as follows:

123456 , initial list of suitors, start counting from 1
12456 , suitor 3 eliminated, continue counting from 4
1245 , suitor 6 eliminated, continue counting from 1
125 , suitor 4 eliminated, continue counting from 5
15 , suitor 2 eliminated, continue counting from 5
1 , suitor 5 eliminated, 1 is the lucky winner .

Write a program that uses a vector to determine which position you should stand in to marry the princess if there are n suitors.

< Hints >

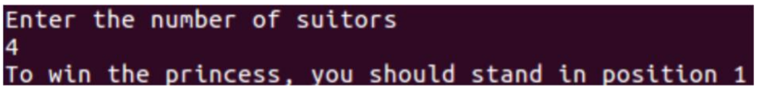
# include <vector>

vector<int> v // Size=0

v.size()   v.begin()   v.erase()// Removes element at position iter

EX : theVector.erase(theVector.begin( ) + 3); // The number 3 is used because the first element in the vector is at index position 0

Sample :



```
Enter the number of suitors
4
To win the princess, you should stand in position 1
```