National University of Computer & Emerging Sciences, Karachi
FINAL –Programming Fundamentals (Lab) CS-Department

Course Code : CL118	Course Name : PROGRAMMING FUNDAMENTALS LAB
Instructor Name / Names:	AQSA ZAHID/ ABDULLAH ZARSHAID/ KARIZ KAMAL
Student Roll No:	Section :

Instructions:

- SUBMIT .c files and pdf containing (outputs of screenshots + code).
- Read each question completely before answering it. In case of any ambiguity, you may make assumption. But your assumption should not contradict any statement in the question paper.
- All the answers must be solved according to the sequence given in the question paper.
- Each student should perform tasks by themselves in case of plagiarism found in submission, case should be forwarded to examination committee.

Time: 3 hours.

Max Marks 60 points
EACH QUESTION CARRY 10 POINTS

Question # 01 :(Structure, Array of Structure)

customers generally come in with an idea of how much money they want to spend. So at the shoes of developer the owner has asked you to build up a functionality where you going to think of items as either:

- CHEAP (under $20),
- MODERATE (between $20 and $100)
- EXPENSIVE (over $100)

Using structure First, fill in the appropriate variable inventory record which will contain name, price, and quantity so that all of the data for the eight items is inside inventory. Make sure that you maintain the notion of CHEAP, MODERATE and EXPENSIVE. Then, implement the function get_info that takes a cheapness and returns a list of information about each item that falls under that category, as the function's information says. Generate bill to your customer at the end.

Question # 02 :(2D Array)

In this pandemic time the demand of supermarket supporting software has been increase. With respect to current situation you are hired to build a solution for real time application.
Survey is conducted for Imtiaz supermarket recently the average number of customer on daily basis per branch per district, results are report as per district as shown below.
Each district have at least 3 branches with an average number of customers.

District No:	branch1	branch2	branch3
1	33	3	159
2	234	122	351
3	100	140	19
4	53	310	76
5	111	2	4

To get some insight you are asked to build program which do following work:

- Which district is getting most profit?
- Which district is making less profit?
- What the total number are of customers as per district?

Suppose the capacity to provide entrance of branch1 is 300, branch2 is 500 and branch3 is 200

- Which branch has maximum capacity utilization per district?
- Which branch has maximum capacity utilization per district?

Question # 03 :(Strings, string function)

Competition is held for primary class students to implement encryption algorithm. Encryption is method to change the formation of message. You are one of the participant to implement given customized encryption algorithm. The method works as of alterations played in English words. To form encrypted version of English word, the conversion happens following the rules such as:

- Moving initial string of consonants (A consonant is a speech sound that is not a vowel. to the end and appending "ay". Eg: trend -> endtray
- Words beginning with a vowel are translated by appending "way". Eg: end --> endway
- The letter 'y' is a treated as a consonant if it appears as the first letter of a word (yard -> ardyay)
- As a vowel if it appears at an interior location (at mid or somewhere) take consonants before the first vowel shift to end. Eg: crybaby-> abycrybay
- Also, "sch" at the beginning should become "sk" at the end. Eg: schooner -> oonerskay
- Capitalized words should be capitalized after transformation. Eg: Sam -> Amsay

Write a c program to implement mentioned encryption algorithm.

Question # 04:(Arithmetic Operations, loops)

A roundnumber is an integer number with the property that any rotation of that number's digits. In this case, rotation refers to cycling the digits of a number; for example, The possible combination of numbers in terms of circular for digit 98321 can be 981,321,98,21 etc. (make your own logic to form the possible combination of two or more digits). Write c program including functions:

1. roundNumber

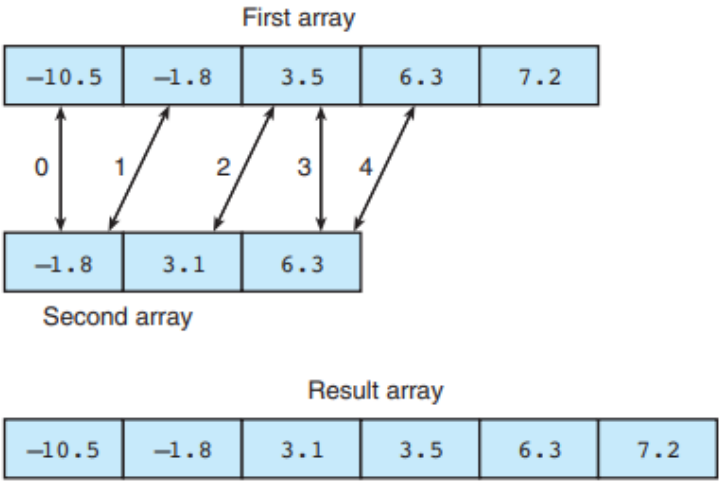
   This function takes a non-negative integer number, x, and rotates that number's digits to form possible collection of combinations.

2. isXNumber
   Determine if number is XNumber. XNumber is number if sum of the squares of digits of number till 1 takes less than 10 iteration.

Eg 23 is xNumber because 2^2+3^2 = 13, 1^2+3^2=10, 1^2+0^2 = 1 (3 steps)

Question # 05:

Write a program containing function use array to pointer that will merge the contents of two sorted (ascending order) arrays of type double values, storing the result in an array output parameter (still in ascending order). The function should not assume that both its input parameter arrays are the same length but should check that one array does not contain two copies of the same value. The result array should also contain no duplicate values.
Example:



Test your function with cases in which:

(1) the first array is shorter than array2 in length
(2) the second array shorter than array1 in length
(3) the two arrays are equal in length

Remember that the arrays input to this function must already be sorted

Question # 06:

(Call By Reference)

Write c program in which define four "chromosomes":

A = "gtggcaacgtgc"

B = "gtagcagcgcgc"

C = "gcggcacagggt"

D = "gtgacaacgtgc"

There are two methods to find the "distance" of two chromosomes. That is, how much the two have varied through mutations. One method is to go letter by letter and count the number of discrepancies (difference). Another method is to sum the discrepancies of a's (e.g. chromosome x has 5 a's, chromosome y has 7 a's), c's, g's and t's.

Write a call by reference function that uses method 1.
It will take two chromosomes, and go letter by letter in each, comparing them. It then returns the total number of discrepancies it finds.

Write another call by reference function that uses method 2. It will take two chromosomes, and return the sum of 'a' discrepancies, 'c' ones, and so on.

Then call each function on each of the combinations (A, B), (A, C), (A, D), (B, C), (B, D) and (C, D) and see which methods tell you which pair of chromosomes is "furthest" (i.e. most varied) and which pair is "closest" (i.e. least varied).