Q1.

Define a class SUPPLY in Java with the following descriptions :

* Members are : Code of int type, FoodName of type String, Sticker of type String, FoodType of type String.
* Member Functions : A member function GetType() to assign the following values for FoodType as per the given sticker

|  |  |
| --- | --- |
| **Sticker** | **FoodType** |
| Green | Vegetarian |
| Yellow | Contains Egg |
| Red | Non Vegetarian |
|  |  |

* A function FoodIn() to allow user to enter values for Code, FoodName, Sticker and call function GetType() to assign respective FoodType.
* A function FoodOut() to allow user to view the content of all the data members.

Q2.

Define a class Resort with the following description:

* Members are : RNo to store Room Number, Name store customer name, Charges to store per day charges, Days to store number of days of stay.
* Member functions :
* Compute() to calculate and return Amount as Days \* Charges and if the values of Days\*Charges is more than 11000 then as 1.02\*Days\*Charges
* Getinfo() A function to enter the content Rno, Name, Charges and Days.
* DispInfo() A function to enter the content Rno, Name, Charges, Days and Amount by calling function Compute().

Q3.

Define a class Candidate with the following description

* Members are : RNo of int type, Name of type String, Score of type float, Remarks of type String.
* Member functions : AssignRem() to assign Remarks as per the Score obtained by a candidate. Score range are given below:

|  |  |
| --- | --- |
| **Score** | **Remarks** |
| >=50 | Selected |
| <50 | Not Selected |

* A function ENTER() to allow user to enter values for RNo, Name, Score and call function AssignRem() to assign the remarks.
* A function DISPLAY() to allow user to view the content of all the data members.

Q4.

Define a class CARRENTAL with the following details :

* Class Members are : CarId of int type, CarType of string type and Rent of float type.
* Define GetCar() method which accepts CarId and CarType.
* GetRent() method which return rent of the car on the basis of car type, i.e. Small Car = 1000, Van = 800, SUV = 2500
* ShowCar() method which allow user to view the contents of cars i.e. id, type and rent.

Q5.

Given a string, compute recursively a new string where all the adjacent chars are now separated by a "\*".

allStar("hello") → "h\*e\*l\*l\*o"  
allStar("abc") → "a\*b\*c"  
allStar("ab") → "a\*b"

Q6.

Given a string, count the number of words ending in 'y' or 'z' -- so the 'y' in "heavy" and the 'z' in "fez" count, but not the 'y' in "yellow" (not case sensitive). We'll say that a y or z is at the end of a word if there is not an alphabetic letter immediately following it. (Note: Character.isLetter(char) tests if a char is an alphabetic letter.)

countYZ("fez day") → 2  
countYZ("day fez") → 2  
countYZ("day fyyyz") → 2