**Practical 9**

1. Consider FruityBar Shop sells various types of fruits. Some of the pricing information is given in the table below. Answer the following questions.

|  |  |  |
| --- | --- | --- |
| Types of fruits | Product Code | Cost Price per KG (MYR) |
| Apple | PAP103 | 10.00 |
| Banana | PBA706 | 5.00 |
| Cherry | PCH989 | 35.00 |
| Durian | PDR450 | 8.00 |

* 1. Rename the string type as Fruits using *typedef* statement. Define the constants of APPLE, BANANA, CHERRY and DURIAN. Declare a variable named FruityFruits as Fruits type.
  2. Create a void function called PrintFruitPrice that will accept the Fruits type variable as a parameter. Print the respective name of the fruit and its cost price using IF-ELSE statement based on the value of the parameter.
  3. Create a *enum* type called FruitTypes. Use the Product Code as its literals. Declare a variable named FruitCodes as FruitTypes type.
  4. Write a function called SearchPrice that will accept the FruitTypes variable as a parameter. Return the price of the respective fruit using *Switch* statement based on the value of the parameter.
  5. Write a function called SearchCode that will accept the Fruits type variable as a parameter. Return the product code as FruitTypes value.
  6. In the main function,
     1. prompt user for the fruits that he wants to purchase. Assign the user input to FruityFruits variable. Then pass FruityFruits as the parameter of the PrintFruitPrice function to display the price of the fruits. Example of the dialog:

Enter fruit name: -> *Apple (input)*

Cost Price of Apple: RM10.00 *(output)*

* + 1. prompt user for the quantity of the fruits that he wants to purchase. Then ask for confirmation. If the user confirms to purchase, use SearchCode function to obtain the product code (assign it to the FruitCodes variable), and SearchPrice function to return the price.
    2. calculate and display the total price of the fruits that the customer purchases.

1. Rename the unsigned int as PositiveInt using *typedef* statement. Declare x as a PositiveInt array that can store 10 values and initialize all of them with zero. Use For loop to generate 10 random numbers and store them into the x array. Display the random numbers on the screen.
2. Define an *enum* type called Emotion that contains of five literals: HAPPY, SATISFIED, BORED, ANGRY, and SAD. Then perform the following:
   1. Create an Emotion type variable named Mood.
   2. Create a function named GetMood, which will prompt user to enter a number that represents his mood, i.e. 0 for happy, 1 for satisfied, and so on. Based on the given index, returns the respective literal of the Emotion type. For instance, if the user enters 0, then HAPPY should be returned by the function.
   3. Create a function named DisplayMessage based on mood. Display the message based on the table below.

|  |  |
| --- | --- |
| HAPPY | A happy heart is a good medicine |
| SATISFIED | Satisfaction of one’s curiosity is one of the greatest happiness in life. |
| BORED | The cure for boredom is curiosity. |
| ANGRY | Anger blows out the lamp of the mind |
| SAD | You can choose to be happy or sad and whichever you choose that is what you get. |

1. Given the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| Product Code | Year Made | Quantity | Price |
| AB567 | 2010 | 200 | 99.50 |
| NZ205 | 2008 | 58 | 138.40 |
| DR043 | 2009 | 124 | 36.90 |

Write a C++ program by performing the following:

1. Define a record structure *PRODUCT*.
2. Declare the structure variable *ProdRec* that includes the above fields.
3. Prompt user to input a product record into variable *ProdRec*.
4. Display the product record input in part (iii) above.
5. Declare an array variable, *product*, to hold 5 product records. At the same time, initialize it with the three records given in the table above.
6. Copy the record input by user in part (iii) into the array variable *product* to be the 4th record in the array.
7. Prompt user to change the 2nd letter of the product code of the 3rd product record, which is from “D**R**043” to “D**G**043”.
8. Prompt user to input a product record to be stored as the last record in the array variable *product*.
9. Calculate the product amount of each product stored in the record structure. Use the formula below:

Product amount = Quantity \* Price

1. Then, display the following table with the format given:

Product Code Product Amount

----------------- ---------------------

XX999 999.99

: :

: :

: :

1. A graphic representation of the record structure is shown below (the structs members are labeled in the diagram):

StudentName

Sex

ICNo

BirthDate

day

month

year

Note that the student name and IC No should be of *character* type, whereas month, day and year are of *integer* type.

Based on the diagram above, write a C++ program for the following tasks:

1. Define a DATE and STUDENT record structure.
2. Declare record structure variable called **studrec** having the above structure. This variable is capable of storing 10 student records.
3. Prompt user to input 3 sets of student records into the *studrec* variable by using a *for* loop. Enter a wrong sex ‘Z’ into 3rd student record.
4. Use another *for* loop to display the student details of *studrec* variable.
5. Use *for* loop to validate all the sex stored in the *studrec* variable. If a particular student’s sex is not ‘M’ or ‘F’, display error “Wrong sex!” and prompt user to enter a new sex until the correct sex is entered (use *while* loop). The validation should continue for the rest of the student records.
6. Assign “Steven Ken” to the 2nd student name.
7. Display the last four digits of the IC number of the 3rd student
8. Prompt user to input new birth date for the 1st student.
9. Display the updated student details of *studrec* variable after the above changes are made.

1. Write a program based on the following requirements:-

(i) Declare a structure named *Employee* that holds an employee’s *ID* (4 characters), name (20 characters) and *department* (5 characters).

1. Declare an array of *Employee* type that can hold 4 employees’ information.

(iii) Assign the following employee information into the array declared in Q4 (ii) during compilation time:-

|  |  |  |
| --- | --- | --- |
| Employee ID | Name | Department |
| E01  E02  E03  E04 | Alice Chin  John  Vivian  Alice Chin | R&D  IT  HR  IT |

Write a program that allows a user to search an employee by employee’s name.

**Extra Exercise**

1. A catalog listing for a textbook consists of the *author’s names*, the *title*, the *publisher*, and the *year of publication*. Declare a structure type *catalog\_entry\_t* and a variable *book.* Then write a program to store and display the relevant data for a textbook in *book.*
2. Declare a structure named TIME that holds two members, hours and minutes. Obtain the input from the user and stores it into the structure. Calculate and display the time one-minute later.
3. Define a structure type to represent a common fraction. Write a program that gets a fraction and displays both the fraction and the fraction reduced to lowest terms using the following code fragment:

frac = get\_fraction();

print\_fraction(frac);

cout << “ = ”;

print\_fraction**(**reduce\_fraction(frac)**)**;

Note:

Normal fraction

Fraction in lowest reduced- form

Denominator

Numerator