**C PROGRAMMING PROBLEMS**

**1.Write a program in which declare all datatypes like integer,double,float,long int and print the values in specified format.**

**#include <stdio.h>**

**int main() {**

**int integer\_var = 10;**

**double double\_var = 3.14159;**

**float float\_var = 2.71828;**

**long int long\_int\_var = 9999999999;**

**printf("Integer: %d\n", integer\_var);**

**printf("Double: %.5f\n", double\_var);**

**printf("Float: %.5f\n", float\_var);**

**printf("Long Int: %ld\n", long\_int\_var);**

**return 0;**

**}**

**2. write a c program to find average of 3 numbers.**

**Sol:**

**#include <stdio.h>**

**int main() {**

**double num1, num2, num3, average;**

**printf("Enter the first number: ");**

**scanf("%lf", &num1);**

**printf("Enter the second number: ");**

**scanf("%lf", &num2);**

**printf("Enter the third number: ");**

**scanf("%lf", &num3);**

**average = (num1 + num2 + num3) / 3;**

**printf("The average of %.2lf, %.2lf, and %.2lf is: %.2lf\n", num1, num2, num3, average);**

**return 0;**

**}**

**3. Write a c program to convert temperature in Celsius to Fahrenheit.**

**Sol:**

**#include <stdio.h>**

**int main() {**

**double celsius, fahrenheit;**

**printf("Enter temperature in Celsius: ");**

**scanf("%f", &celsius);**

**fahrenheit = (celsius \* 9 / 5) + 32;**

**printf("%f degrees Celsius is equal to %.2lf degrees Fahrenheit.\n", celsius, fahrenheit);**

**return 0;**

**}**

**4. Write a c program to compare two numbers using ternery operator.**

**Sol:**

**#include <stdio.h>**

**int main() {**

**int num1, num2, max;**

**printf("Enter the first number: ");**

**scanf("%d", &num1);**

**printf("Enter the second number: ");**

**scanf("%d", &num2);**

**max = (num1 > num2) ? num1 : num2;**

**printf("The maximum number is: %d\n", max);**

**return 0;**

**}**

**5. Write a c program to find the largest number in given 3 numbers using ternery operator.**

**Sol:**

**#include <stdio.h>**

**int main() {**

**int num1, num2, num3, max;**

**printf("Enter the first number: ");**

**scanf("%d", &num1);**

**printf("Enter the second number: ");**

**scanf("%d", &num2);**

**printf("Enter the third number: ");**

**scanf("%d", &num3);**

**max = (num1 > num2) ? ((num1 > num3) ? num1 : num3) : ((num2 > num3) ? num2 : num3);**

**printf("The largest number is: %d\n", max);**

**return 0;**

**}**

**6. Write a C program to write the reverse of the numbers.**

**Sol:**

**#include <stdio.h>**

**int main() {**

**int number, reversedNumber = 0, remainder;**

**printf("Enter a number: ");**

**scanf("%d", &number);**

**while (number != 0) {**

**remainder = number % 10;**

**reversedNumber = reversedNumber \* 10 + remainder;**

**number /= 10;**

**}**

**printf("The reverse of the number is: %d\n", reversedNumber);**

**return 0;**

**}**

**7. Write a c program to write the multiples of a given number.**

**Sol:**

**#include <stdio.h>**

**int main() {**

**int number;**

**printf("Enter a number: ");**

**scanf("%d", &number);**

**printf("Multiples of %d: \n", number);**

**for (int i = 1; i <= 10; i++) {**

**int multiple = number \* i;**

**printf("%d ", multiple);**

**}**

**printf("\n");**

**return 0;**

**}**

**8. Write a c program to calculate simple interest when principle amount ,rate of interst and no of years is given.**

**Sol:**

**#include <stdio.h>**

**int main() {**

**double principal, rate, time, simpleInterest;**

**printf("Enter the principal amount: ");**

**scanf("%lf", &principal);**

**printf("Enter the interest rate: ");**

**scanf("%lf", &rate);**

**printf("Enter the number of years: ");**

**scanf("%lf", &time);**

**simpleInterest = (principal \* rate \* time) / 100;**

**printf("The simple interest is: %.2lf\n", simpleInterest);**

**return 0;**

**}**

**9.** **Write a C Code to check whether the number that is given by the user is a multiple of 8. Continue getting the numbers till -1 is pressed by the user. Display the count of both the cases.**

**Sol:**

**#include <stdio.h>**

**int main() {**

**int num, mult\_8\_count = 0, non\_mult\_8\_count = 0;**

**while (1) {**

**scanf("%d", &num);**

**if (num == -1) {**

**break;**

**}**

**if (num % 8 == 0) {**

**mult\_8\_count++;**

**} else {**

**non\_mult\_8\_count++;**

**}**

**}**

**printf("Multiples\_of\_8 is %d\n", mult\_8\_count);**

**printf("Non\_Multiples\_of\_8 is %d\n", non\_mult\_8\_count);**

**return 0;**

**}**

**10. Given an integer array nums of length n and an integer target, find three integers in nums such that the sum is closest to target.**

**Return the sum of the three integers.You may assume that each input would have exactly one solution write a c program**

**Sol:**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include<limits.h>**

**int compare(const void \*a, const void \*b) {**

**return (\*(int\*)a - \*(int\*)b);**

**}**

**int threeSumClosest(int\* nums, int numsSize, int target) {**

**int closestSum = INT\_MAX;**

**int diff = INT\_MAX;**

**qsort(nums, numsSize, sizeof(int), compare);**

**for (int i = 0; i < numsSize - 2; i++) {**

**int left = i + 1;**

**int right = numsSize - 1;**

**while (left < right) {**

**int sum = nums[i] + nums[left] + nums[right];**

**int currentDiff = abs(target - sum);**

**if (currentDiff < diff) {**

**diff = currentDiff;**

**closestSum = sum;**

**}**

**if (sum < target) {**

**left++;**

**} else {**

**right--;**

**}**

**}**

**}**

**return closestSum;**

**}**

**int main() {**

**int nums[] = {-1, 2, 1, -4};**

**int numsSize = 4;**

**int target = 1;**

**int closestSum = threeSumClosest(nums, numsSize, target);**

**printf("Closest sum = %d", closestSum);**

**return 0;**

**}**

**11. Mr. Richard was travelling in a train. The train started from Chennai at 8 hours, 30 minutes and 30 seconds and reached Madurai at 18 hrs, 50 minutes and 40 seconds. (i.e) it took 10 hours, 20 minutes and 10 seconds to travel. Write a program using default arguments in C++ and calculate the arrival time with the departure being provided by the user. (Answer should be in 24:60:60 formats).Write a c++ program**

**Sol:**

**#include <iostream>**

**using namespace std;**

**void calculateArrivalTime(int depHour, int depMin, int depSec, int travelTimeHour = 10, int travelTimeMin = 20, int travelTimeSec = 10) {**

**int depTime = depHour \* 3600 + depMin \* 60 + depSec;**

**int arrivalTime = depTime + travelTimeHour \* 3600 + travelTimeMin \* 60 + travelTimeSec;**

**int arrivalHour = arrivalTime / 3600;**

**int arrivalMin = (arrivalTime % 3600) / 60;**

**int arrivalSec = (arrivalTime % 3600) % 60;**

**cout << "The train will arrive at " << arrivalHour << ":" << arrivalMin << ":" << arrivalSec << endl;**

**}**

**int main() {**

**int depHour, depMin, depSec;**

**cout << "Enter departure time (hh mm ss): ";**

**cin >> depHour >> depMin >> depSec;**

**calculateArrivalTime(depHour, depMin, depSec);**

**return 0;**

**}**

**12. Write a function in C to print all the LEADERS in the array. An element is a leader if it is greater than all the elements to its right side. Special rule For example in the array {16, 17, 4, 3, 5, 2}, Leaders are 17, 5 and 2. Special rule says that the rightmost element in the array is always a leader. For example in the array {16, 17, 4, 3, 5, 2}, Leaders are 17, 5 and 2.**

**Sol:**

**#include <stdio.h>**

**void printLeaders(int arr[], int n) {**

**int max\_so\_far = arr[n-1];**

**printf("Leaders: %d ", max\_so\_far);**

**for (int i = n-2; i >= 0; i--) {**

**if (arr[i] > max\_so\_far) {**

**printf("%d ", arr[i]);**

**max\_so\_far = arr[i];**

**}**

**}**

**}**

**int main() {**

**int arr[] = {16, 17, 4, 3, 5, 2};**

**int n = sizeof(arr) / sizeof(arr[0]);**

**printLeaders(arr, n);**

**return 0;**

**}**

**13. Write a C program to check whether the same numbers appear in the given set of numbers in the first and last position.**

**If so, find the sqaure of the number and store it in a variable that should not be returned but should be accessed in main.**

**Else, find the cube of the number and return it to main.**

**Use multiple functions if needed.**

**Sample Input:**

**Number of elements : 5**

**Element 1**

**Element 2**

**Element 3**

**Element 4**

**Element 5**

**Output:**

**Square of the number if both element 1 and element 5 are same**

**Cube of numbers if both are different IN C PROGRAM.**

**Sol:**

**#include <stdio.h>**

**int square\_or\_cube(int arr[], int n);**

**int main()**

**{**

**int n, i;**

**printf("Enter the number of elements: ");**

**scanf("%d", &n);**

**int arr[n];**

**printf("Enter the elements: ");**

**for (i = 0; i < n; i++) {**

**scanf("%d", &arr[i]);**

**}**

**int result = square\_or\_cube(arr, n);**

**if (result < 0) {**

**printf("The first and last elements are different. Cube of the number is %d", -result);**

**} else {**

**printf("The first and last elements are same. Square of the number is %d", result);**

**}**

**return 0;**

**}**

**int square\_or\_cube(int arr[], int n)**

**{**

**int result;**

**if (arr[0] == arr[n-1]) {**

**result = arr[0] \* arr[0];**

**} else {**

**result = -(arr[0] \* arr[0] \* arr[0]);**

**}**

**return result;**

**}**

**14.**

**Write a C code using functions to find the lcm of two numbers using functions.**

**Input format:**

**number 1**

**number 2**

**Output format:**

**LCM of the two numbers is to be printed**

**Sol:**

**#include <stdio.h>**

**int gcd(int num1, int num2) {**

**if (num2 == 0)**

**return num1;**

**else**

**return gcd(num2, num1 % num2);**

**}**

**int lcm(int num1, int num2) {**

**int gcd\_value = gcd(num1, num2);**

**return (num1 \* num2) / gcd\_value;**

**}**

**int main() {**

**int num1, num2, lcm\_value;**

**printf("Enter two numbers: ");**

**scanf("%d %d", &num1, &num2);**

**lcm\_value = lcm(num1, num2);**

**printf("LCM of %d and %d is %d\n", num1, num2, lcm\_value);**

**return 0;**

**}**

**15. Write a C program to generate odd and the even position th sum of Fibonacci series**

**Value of n is given as input.**

**Output Format:**

**Series in the same line separated by comma**

**Odd position total**

**Even position total using functions**

**Sol:**

**#include <stdio.h>**

**void fibonacci(int n, int \*odd\_sum, int \*even\_sum) {**

**int prev = 0, curr = 1, temp, i;**

**for (i = 1; i <= n; i++) {**

**printf("%d", curr);**

**if (i != n) {**

**printf(", ");**

**}**

**if (i % 2 == 0) {**

**\*even\_sum += curr;**

**} else {**

**\*odd\_sum += curr;**

**}**

**temp = curr;**

**curr = prev + curr;**

**prev = temp;**

**}**

**}**

**int main() {**

**int n, odd\_sum = 0, even\_sum = 0;**

**printf("Enter the value of n: ");**

**scanf("%d", &n);**

**fibonacci(n, &odd\_sum, &even\_sum);**

**printf("\nSum of odd positions: %d\n", odd\_sum);**

**printf("Sum of even positions: %d\n", even\_sum);**

**return 0;**

**}**

**16. Get the size "N" and the "N" elements of a array.**

**Find the largest number from the array.**

**input will contain:**

**i) value of n**

**ii) n elements of the array (one below the other)**

**Output:**

**i)n elements of the array (one below the other)**

**ii) The largest number:**

**the answer for the largest number in c program**

**Sol:**

**#include <stdio.h>**

**int main() {**

**int n, arr[100], i, max;**

**printf("Enter the size of the array: ");**

**scanf("%d", &n);**

**printf("Enter the array elements:\n");**

**for(i=0; i<n; i++) {**

**scanf("%d", &arr[i]);**

**}**

**max = arr[0];**

**for(i=1; i<n; i++) {**

**if(arr[i] > max) {**

**max = arr[i];**

**}**

**}**

**printf("Array elements:\n");**

**for(i=0; i<n; i++) {**

**printf("%d\n", arr[i]);**

**}**

**printf("The largest number is: %d\n", max);**

**return 0;**

**}**

**17. Define a Structure to store name, account number and balance of customers in a**

**bank.**

**Get the number of customers of the bank.**

**Get the name, account number and balanace of customers from the user.**

**Write a function to print the names of customers having balance less than**

**Rs.2000.Write a function to add an interest of 5% to the balance of customers**

**having more than Rs.10000 in their balance. Print the details of customers who got**

**interest added.**

**Input**

**3**

**Raju**

**9999**

**3000**

**Ramu**

**8888**

**11000**

**Rama**

**7777**

**1200**

**Expected output**

**Ramu**

**8888**

**11550**

**Rama**

**Sol:**

**#include<stdio.h>**

**#include<string.h>**

**struct Customer{**

**char name[50];**

**int acc;**

**int bal;**

**};**

**int getCustomerDetails(struct Customer \*customer) {**

**scanf("%s",&customer->name);**

**scanf("%d",&customer->acc);**

**scanf("%d",&customer->bal);**

**return 0;**

**}**

**int main() {**

**int c;**

**scanf("%d",&c);**

**struct Customer customers[c];**

**for(int i = 0; i<c; i++) {**

**getCustomerDetails(&customers[i]);**

**}**

**for(int i = 0; i<c; i++){**

**if (customers[i].bal<2000) {**

**printf("%s ",customers[i].name);**

**}**

**else if(customers[i].bal>10000){**

**float interest = customers[i].bal\* 0.05;**

**customers[i].bal+=interest;**

**printf("%s\n",customers[i].name);**

**printf("%d\n",customers[i].acc);**

**printf("%d\n",customers[i].bal);**

**}**

**}**

**return 0;**

**}**

**18. Given an integer, n, perform the following conditional actions:**

**If n is odd, print Weird**

**If n is even and in the inclusive range of 2 to 5, print Not Weird (2,3,4,5)**

**If n is even and in the inclusive range of 6 to 20, print Weird (6,7,…,19,20)**

**If n is even and greater than 20, print Not Weird (21, 22..)\*/**

**Sol:**

**#include<stdio.h>**

**int main(){**

**int n;**

**scanf("%d",&n);**

**if(n%2!=0){**

**printf("Weird");**

**}**

**else{**

**if(1<n<5){**

**printf("Not Weird");**

**}**

**else if(5<n<21){**

**printf("Weird");**

**}**

**else{**

**printf("Not Weird");**

**}**

**}**

**return 0;**

**}**

**19. Bob has "n" bags full of playing coins.**

**Get the count of the coins in each bag and store it on an array.**

**From the array, sum all the items and display the total coins with him.**

**input will contain:**

**i) value of n**

**ii) count of coins in each bag (one below the other)**

**output should contain the following:**

**i) count of the coins separated by comma in each bag**

**ii) sum of all the coins**

**Sol:**

**#include <stdio.h>**

**int main() {**

**int n, coins[100], i, sum = 0;**

**printf("Enter the number of bags: ");**

**scanf("%d", &n);**

**printf("Enter the count of coins in each bag:\n");**

**for(i = 0; i < n; i++) {**

**scanf("%d", &coins[i]);**

**}**

**printf("Count of coins in each bag:\n");**

**for(i = 0; i < n; i++) {**

**printf("%d", coins[i]);**

**if(i != n-1) {**

**printf(", ");**

**}**

**sum += coins[i];**

**}**

**printf("\nTotal number of coins: %d\n", sum);**

**return 0;**

**}**

**20. Get an array of integers display them.**

**Then, remove duplicates from the array and display:**

**i).the removed elements that comes after the first occurance (one below the other).**

**ii). the revised array after removal**

**Sol:**

**#include<stdio.h>**

**int main(){**

**int n;**

**scanf("%d",&n);**

**int arr[n];**

**int count=0;**

**for(int i=0;i<n;i=i+1){**

**scanf("%d",&arr[i]);**

**}**

**for(int i=0;i<n-1;i=i+1){**

**printf("%d, ",arr[i]);**

**}**

**printf("%d\n",arr[n-1]);**

**for(int i=0;i<n;i=i+1){**

**for(int u=1;u<n;u=u+1){**

**if(arr[i]==arr[u+i]){**

**printf("%d\n",arr[i]);**

**count+=1;**

**}**

**else{**

**continue;**

**}**

**}**

**}**

**int d=n-count;**

**int arrn[d];**

**for(int i=0;i<n;i=i+1){**

**for(int u=1;u<n;u=u+1){**

**if(arr[i]==arr[u+i]){**

**arrn[i]=arr[i];**

**}**

**else{**

**arrn[i]=arr[i];**

**}**

**}**

**}**

**for(int i=0;i<d-1;i=i+1){**

**printf("%d, ",arrn[i]);**

**}**

**printf("%d",arrn[d-1]);**

**return 0;**

**}**

**21. Write a c program to print count of Angstrom numbers until given numbers.**

**Sol:**

**#include<stdio.h>**

**#include<math.h>**

**int main(){**

**int n;**

**printf("Enter the no till which u want armstrong no's:");**

**scanf("%d",&n);**

**int i=0;**

**int count=0;**

**for(i;i<=n;i+=1){**

**int r=1;**

**int c=i;**

**int s=0;**

**int sum=0;**

**while(r!=0){**

**r=c/10;**

**c=r;**

**s+=1;**

**}**

**int g=s;**

**int k=i;**

**while(s>0){**

**int d;**

**d=k%10;**

**k=k/10;**

**sum+=pow(d,g);**

**s-=1;**

**}**

**if(sum==i){**

**count+=1;**

**printf("\nArmstrong no: %d",i);**

**}**

**else{**

**continue;**

**}**

**}**

**printf("\nTotal counts of armstrong no's till number %d is :%d",n,count);**

**return 0;**

**}**

**22. EMI and Education expenses of a family is to be calculated for two months (May, June) using operator overloading concept in C++ language. Create two objects, one for each month. Create two functions for EMI and Expenditure, an operator overloading function to add up the expenses of same month and return the total EMI expense and total education expense via separate functions. Print the total EMI expenses of both the months in the main function. You should also print the total education expenses of both the months in the main function.**

**Sol:**

**#include <iostream>**

**#include<string>**

**using namespace std;**

**class expense{**

**public:**

**int emi;**

**int expenditure;**

**void set()**

**{**

**cin>>emi;**

**cin>>expenditure;**

**}**

**expense operator+(expense operator1){**

**expense x;**

**x.emi=emi+operator1.emi;**

**x.expenditure=expenditure+operator1.expenditure;**

**return x;**

**}**

**};**

**int main()**

**{**

**expense may,june,total;**

**may.set();**

**june.set();**

**total=may+june;**

**cout<<"Total expenses for EMI: "<<total.emi<<endl;**

**}**

**23. A magic person wants to put some data of the same type into his magic bag. He wants to repeat the**

**same magic with different data such as int, string, float etc. could be put into his bag.**

**The process of putting the data inside the bag is said to be as “Push”. The bag contains some restriction in its size. Ie. The size must be specified in the initial part of every iteration.Once when the input crosses the maximum limit it displays the message "Bag is full"**

**Implement this magic person’s game for his specification of any datatype being used inside his bag for each iteration.**

**Note that he should not give clue of doing this operation to anyone. So have the same name for the function that does the push operation for every datatype.**

**Sol:**

**#include <iostream>**

**#include <vector>**

**#include <typeinfo>**

**template <typename T>**

**class MagicBag {**

**private:**

**std::vector<T> bag;**

**int maxSize;**

**public:**

**MagicBag(int size) : maxSize(size) {}**

**void Push(T item) {**

**if (bag.size() < maxSize) {**

**bag.push\_back(item);**

**std::cout << "Pushed item into the bag: " << item << std::endl;**

**} else {**

**std::cout << "Bag is full." << std::endl;**

**}**

**}**

**};**

**int main() {**

**MagicBag<int> intBag(3);**

**MagicBag<std::string> stringBag(5);**

**MagicBag<float> floatBag(2);**

**intBag.Push(10);**

**intBag.Push(20);**

**intBag.Push(30);**

**intBag.Push(40);**

**stringBag.Push("Apple");**

**stringBag.Push("Banana");**

**stringBag.Push("Cherry");**

**stringBag.Push("Durian");**

**stringBag.Push("Elderberry");**

**stringBag.Push("Fig");**

**floatBag.Push(3.14);**

**floatBag.Push(2.71);**

**floatBag.Push(1.23);**

**return 0;**

**}**

**24. Create a program to model a banking system. You have a BankAccount class that stores information about each account, including the account holder’s name, account number, and balance. You want to add a function to the class that allows account holders to transfer money to another account, but you need to access the private member variables of two BankAccount objects to do so. Write a C++ program that implements a transferMoney() function as a friend function of the BankAccount class. The transferMoney() function should take two BankAccount objects as arguments and transfer a specified amount of money from the first account to the second account. Make sure to include appropriate member functions in the BankAccount class to store and retrieve account information. Also, make sure to handle cases where the transfer amount exceeds the balance of the first account or where the account numbers are invalid.**

**Sol:**

**#include <iostream>**

**#include <string>**

**class BankAccount {**

**private:**

**std::string accountHolderName;**

**int accountNumber;**

**double balance;**

**public:**

**BankAccount(const std::string& name, int number, double initialBalance)**

**: accountHolderName(name), accountNumber(number), balance(initialBalance) {}**

**void displayAccountDetails() {**

**std::cout << "Account Holder Name: " << accountHolderName << std::endl;**

**std::cout << "Account Number: " << accountNumber << std::endl;**

**std::cout << "Balance: " << balance << std::endl;**

**}**

**friend void transferMoney(BankAccount& sender, BankAccount& receiver, double amount);**

**};**

**void transferMoney(BankAccount& sender, BankAccount& receiver, double amount) {**

**if (amount > sender.balance) {**

**std::cout << "Insufficient balance in the sender's account." << std::endl;**

**return;**

**}**

**sender.balance -= amount;**

**receiver.balance += amount;**

**std::cout << "Transfer successful." << std::endl;**

**std::cout << "Transferred amount: " << amount << std::endl;**

**std::cout << "\nSender's Account Details:" << std::endl;**

**sender.displayAccountDetails();**

**std::cout << "------------------------" << std::endl;**

**std::cout << "Receiver's Account Details:" << std::endl;**

**receiver.displayAccountDetails();**

**}**

**int main() {**

**BankAccount sender("John Doe", 123456, 3000);**

**BankAccount receiver("Alice Smith", 987654, 3000);**

**std::cout << "Sender's Account Details:" << std::endl;**

**sender.displayAccountDetails();**

**std::cout << "------------------------" << std::endl;**

**std::cout << "Receiver's Account Details:" << std::endl;**

**receiver.displayAccountDetails();**

**std::cout << "\nEnter the amount to transfer: ";**

**double amount;**

**std::cin >> amount;**

**transferMoney(sender, receiver, amount);**

**return 0;**

**}**

**25. Design a C++ program to model a supermarket that maintains three different product sections: grocery, vegetables, and snacks. The program should utilize inheritance to create derived classes for each section, inheriting methods from a base class called "Product." Each product should have a name and a price.**

**Additionally, implement a "Customer" class that allows customers to add products to their cart and calculates their total bill. The program should prompt the user to input customer details and the products they want to purchase. If the total bill exceeds a certain amount of 5$, provide a discount of 10% to the customer.**

**Sol:**

**#include <iostream>**

**#include <string>**

**#include <vector>**

**using namespace std;**

**class Product {**

**public:**

**Product(string name, double price) : name(name), price(price) {}**

**virtual ~Product() {}**

**string getName() const { return name; }**

**double getPrice() const { return price; }**

**private:**

**string name;**

**double price;**

**};**

**class Grocery : public Product {**

**public:**

**Grocery(string name, double price) : Product(name, price) {}**

**};**

**class Vegetable : public Product {**

**public:**

**Vegetable(string name, double price) : Product(name, price) {}**

**};**

**class Snack : public Product {**

**public:**

**Snack(string name, double price) : Product(name, price) {}**

**};**

**class Customer {**

**public:**

**Customer(string name) : name(name), totalBill(0.0) {}**

**void addItem(Product\* item) {**

**cart.push\_back(item);**

**totalBill += item->getPrice();**

**}**

**double getTotalBill() const {**

**return totalBill;**

**}**

**void applyDiscount() {**

**if (totalBill > 5.0) {**

**totalBill \*= 0.9;**

**}**

**}**

**private:**

**string name;**

**vector<Product\*> cart;**

**double totalBill;**

**};**

**int main() {**

**Grocery rice("Rice", 1.0);**

**Vegetable carrot("Carrot", 1.99);**

**Snack chips("Chips", 2.0);**

**Customer john("John");**

**john.addItem(&rice);**

**john.addItem(&chips);**

**john.applyDiscount();**

**cout << "Customer: " << john.getName() << endl;**

**cout << "Items Purchased:" << endl;**

**for (Product\* item : john.cart) {**

**cout << "Product: " << item->getName() << endl;**

**cout << "Price: $" << item->getPrice() << endl << endl;**

**}**

**cout << "Total Bill: $" << john.getTotalBill() << endl << endl;**

**Customer jane("Jane");**

**jane.addItem(&carrot);**

**jane.addItem(&chips);**

**jane.applyDiscount();**

**cout << "Customer: " << jane.getName() << endl;**

**cout << "Items Purchased:" << endl;**

**for (Product\* item : jane.cart) {**

**cout << "Product: " << item->getName() << endl;**

**cout << "Price: $" << item->getPrice() << endl << endl;**

**}**

**cout << "Total Bill: $" << jane.getTotalBill() << endl;**

**return 0;**

**}**

**26. Write a c++ program to search a word in given string and count the number of times the word occurs.**

**Sol:**

**#include <stdio.h>**

**#include <string.h>**

**int main()**

**{**

**char s[1000],w[1000];**

**int n,a[1000],i,j,k=0,l,found=0,t=0;**

**printf("Enter the string : ");**

**gets(s);**

**printf("Enter word to be searched: ");**

**gets(w);**

**for(i=0;s[i];i++)**

**{**

**if(s[i]==' ')**

**{**

**a[k++]=i;**

**}**

**}**

**a[k++]=i;**

**j=0;**

**for(i=0;i<k;i++)**

**{**

**n=a[i]-j;**

**if(n==strlen(w))**

**{**

**t=0;**

**for(l=0;w[l];l++)**

**{**

**if(s[l+j]==w[l])**

**{**

**t++;**

**}**

**}**

**if(t==strlen(w))**

**{**

**found++;**

**}**

**}**

**j=a[i]+1;**

**}**

**printf("word '%s' is occurred count=%d ",w,found);**

**return 0;**

**}**

**27.**