**Sri Sri University, Cuttack, Odisha.**

**Faculty of Science**

|  |  |
| --- | --- |
| **Program: B.Sc. – Computer Science, Data Science, & Environmental Science**  **(2020-23 Batch)**  **Subject Code/Subject Name: Data Structure Laboratory**  **Assignment –IV** | |
| **Full Name of the Student:** | VINAYAK SANJAY CHAVAN(CS) |
| **Full Roll Number:** | BCS-011 |
| **Program:** | B.Sc. (Computer Sc.) / B.Sc. (Data Sc.) / B.Sc. (Env. Sc.) |
| **Date:** | 8th March, 2021 (10.00 AM – 12.00 Noon) |
| **Signature** |  |

|  |  |
| --- | --- |
| **All Questions are compulsory** | **Total Marks: 40** |

|  |  |
| --- | --- |
| **Question (s)** | **Maximum Marks** |
| 1. Define structure that describes the set of books in a library. For each book, the members are name of author, publisher, rate, and branch information.   Write a program to print.   1. A list of books supplied by a publishers and, 2. A list of books in particular branch and their total cost.   **Your code:** #include<iostream>  #include <cstdlib>  using namespace std;  typedef struct book{  char book\_name[30];  char author\_name[10];  char publiser\_name[50];  int rate;  char branch;  }book\_info;  void publiser(book\_info[],int);  int sum\_of\_price(book\_info[],int,char);  int main(){  int cs\_price,ds\_price,es\_price,ms\_price;  int size;  cout<<"Enter the size of array: ";  cin>>size;  book\_info book\_arr[size];  for(int i=0;i<size;i++){  cout<<"Enter the book name "<<i+1<<": ";  cin>>(book\_arr[i].book\_name);  cout<<"Enter the author of the book "<<i+1<<": ";  cin>>(book\_arr[i].author\_name);  cout<<"Enter the publisher name "<<i+1<<": ";  cin>>(book\_arr[i].publiser\_name);  cout<<"Enter the branch for which the book is for "<<i+1<<": ";  cin>>(book\_arr[i].branch);  cout<<"enter the price "<<i+1<<": ";  cin>>(book\_arr[i].rate);  cout<<endl;  }  cs\_price=sum\_of\_price(book\_arr,size,'c');  ds\_price=sum\_of\_price(book\_arr,size,'d');  es\_price=sum\_of\_price(book\_arr,size,'e');  ms\_price=sum\_of\_price(book\_arr,size,'m');  cout<<"The total price of book in cs is: "<<cs\_price<<endl;  cout<<"The total price of book in ds is: "<<ds\_price<<endl;  cout<<"The total price of book in es is: "<<es\_price<<endl;  cout<<"The total price of book in ms is: "<<ms\_price<<endl;  cout<<"list of publisher: ";  publiser(book\_arr,size);  }  int sum\_of\_price(book\_info b1[],int n,char branch\_code){  int total=0;  for(int i=0;i<n;i++){  if(b1[i].branch==branch\_code){  total=total+b1[i].rate;  }  }  return total;  }  void publiser(book\_info b1[],int n){  for(int i=0;i<n;i++){  cout<<(b1[i].publiser\_name)<<endl;  }  }  **Screenshot of output:** | 10 |
| 1. Design a structure type Student that contains the name, roll number, gender of a student and pointers to the student’s house leader (consider there are two houses in a school). Write statements that define a structure value for 10 students, correctly establishing the pointer links for leader.   **Your code:** **/\* 2. Design a structure type Student that contains the**  **name, roll number, gender of a student and pointers to the student's house leader**  **(consider there are two houses in a school). Write statements that define a**  **structure value for 10 students, correctly establishing the pointer links for leader \*/**    **#include <iostream>**  **#include <cstdlib>**  **#include <cmath>**  **using namespace std;**  **typedef struct details\_of\_stud{**  **char name[50];**  **int rollNum;**  **char gender[10];**  **struct details\_of\_stud \* leader = NULL;**  **}STUDENT ;**    **int main(void){**  **int size;**  **cout<<"enter the array size: ";**  **cin>>size;**  **STUDENT arr[size];**  **for(int i=0; i<size; i++){**  **cout<<"write a name of student: "<<i+1<<": ";**  **cin>>(arr[i].name);**  **cout<<"Enter roll no "<<i+1<<": ";**  **cin>>(arr[i].rollNum);**  **cout<<"write gender "<<i+1<<": ";**  **cin>>(arr[i].gender);**    **/\* House Leader of All even students**  **are first students, and for all odd students are second.\*/**    **if(i%2==0){**  **arr[i].leader = &arr[0];**  **}else{**  **arr[i].leader = &arr[1];**  **}**  **}**  **cout<<"printing the data"<<endl;**  **for(int i=0;i<size;i++){**  **cout<<"\n STUDENT "<<i<<endl;**  **cout<<" name "<<arr[i].name<<endl;**  **cout<<" gender "<<arr[i].gender<<endl;**  **cout<<" roll number "<<arr[i].rollNum<<endl;**  **cout<<"house leader"<<arr[i].leader->name<<endl;**  **}**  **}**  **Screenshot of output:** | 10 |
| 1. A program that reads coordinates of 3 points in a plane and finds out the area of the triangle formed by these three points. *If the triangle is valid else return warning message with 0 (zero) return.*   **Your code:** **/\* A program that reads coordinates of 3 points in a plane and finds out the area of the**  **triangle formed by these three points. If the triangle is valid**  **else return warning message with 0 (zero) return.**  **\*/**  **#include<iostream>**  **#include<cmath>**  **using namespace std;**  **typedef struct{**  **int x;**  **int y;**  **}POINT;**  **/\***  **or...**  **typedef struct point{**  **int x\_coord;**  **int y\_coord;**  **}POINT;**  **POINT P1,P2,P3;**    **P1.x\_coord=2;**  **P1.y\_coord=4;**  **\*/**    **double triarea(int,int,int,int,int,int);**  **int main(){**  **POINT p1,p2,p3;**  **double my\_area;**  **cout<<"value of x coordinate of 1st triangle: ";**  **cin>>p1.x;**  **cout<<"value of y coordinate of 1st triangle: ";**  **cin>>p1.y;**  **cout<<"value of x coordinate of second triangle: ";**  **cin>>p2.x;**  **cout<<"value of y coordinate of second triangle: ";**  **cin>>p2.y;**  **cout<<"value of x coordinate of third triangle: ";**  **cin>>p3.x;**  **cout<<"value of y coordinate of third triangle: ";**  **cin>>p3.y;**  **my\_area=triarea(p1.x,p1.y,p2.x,p2.y,p3.x,p3.y);**  **cout<<"area of triangle is: "<<my\_area;**  **}**  **double triarea(int x1,int y1,int x2,int y2,int x3,int y3){**  **double s=0;**  **double area1,area2;**  **double a,b,c;**  **a=sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2));**  **b=sqrt(pow(x3 - x2, 2) + pow(y3 - y2, 2));**  **c=sqrt(pow(x3 - x1, 2) + pow(y3 - y1, 2));**  **s=(a+b+c)/2;**  **area2=(s-a)\*(s-b)\*(s-c);**  **area1=sqrt(area2);**  **return area1;**  **if((a+b>c)||(a+c>b)||(b+c>a)){**  **return area1;**  **}else{**  **return area1=0;**  **}**  **}**  **Screenshot of output:** | 10 |
| 1. Write a program that reads coordinates of 3 points in a plane and check whether a triangle is equilateral, scalene or isosceles triangle.   **Your code:** **#include <iostream>**  **#include <cstdlib>**  **#include <cmath>**  **using namespace std;**  **typedef struct tri{**  **int x;**  **int y;**  **}point;**  **int main(){**  **point p1,p2,p3;**    **cout<<"1st point: ";**  **cin>>p1.x>>p1.y;**  **cout<<"2nd point: ";**  **cin>>p2.x>>p2.y;**  **cout<<"3rd point: ";**  **cin>>p3.x>>p3.y;**    **double a,b,c;**  **a=sqrt(pow(p1.x - p2.x, 2) + pow(p1.y - p2.y, 2));**  **b=sqrt(pow(p2.x - p3.x, 2) + pow(p2.y - p3.y, 2));**  **c=sqrt(pow(p3.x - p1.x, 2) + pow(p3.y - p1.y, 2));**    **if(a==b && b==c){**  **cout<<"equilateral triangle";**  **}else if(a==b || b==c || c==a){**  **cout<<"isosceles triangle";**  **}else{**  **cout<<"scalene triangle";**  **}**  **}**  **Screenshot of output:** | 10 |