**Structures and Unions**

Department of Computer Science and Engineering, Rajalakshmi Engineering College

|  |  |
| --- | --- |
| **Ex. No. :** | **Date :** |
|  | **Boxes through a Tunnel** |
| **Problem Statement:**  You are transporting some boxes through a tunnel, where each box is a parallelepiped, and is characterized by its length, width and height.  The height of the tunnel 41 feet and the width can be assumed to be infinite. A box can be carried through the tunnel only if its height is strictly less than the tunnel's height. Find the volume of each box that can be successfully transported to the other end of the tunnel.  Note: Boxes cannot be rotated.  Input Format  The first line contains a single integer n, denoting the number of boxes.  n lines follow with three integers on each separated by single spaces - lengthi, widthi and heighti which are length, width and height in feet of the i-th box.  Constraints  1 ≤ n ≤ 100  1 ≤ lengthi, widthi, heighti ≤ 100  Output Format  For every box from the input which has a height lesser than 41 feet, print its volume in a separate line.  Sample Input 4  5 5 5  1 2 40  10 5 41  7 2 42  Sample Output 125  80  Explanation  The first box is really low, only 5 feet tall, so it can pass through the tunnel and its volume is 5 x 5 x 5 = 125.  The second box is sufficiently low, its volume is 1 x 2 x 4= = 80.  The third box is exactly 41 feet tall, so it cannot pass. The same can be said about the fourth box. | |

Department of Computer Science and Engineering, Rajalakshmi Engineering College

151

|  |
| --- |
| **Program:**  **#include<stdio.h>**  **Int main()**  **{**  **Int n;**  **Scanf(“%d”,&n);**  **Int length, width, height;**  **Const int tunnelHeight = 41;**  **For(int i=0; i<n; i++)**  **{**  **Scanf(“%d %d %d”, &length, &width, &height);**  **If (height < tunnelHeight)**  **{**  **Int volume = length\*width\*height;**  **Printf(“%d\n”, volume);**  **}**  **}**  **Return 0;**  **}** |

Department of Computer Science and Engineering, Rajalakshmi Engineering College

|  |  |
| --- | --- |
| **Ex. No. :** | **Date :** |
| **Small Triangles, Large Triangles** | |
| **Problem Statement:**  You are given n triangles, specifically, their sides ai, bi and ci. Print them in the same style but sorted by their areas from the smallest one to the largest one. It is guaranteed that all the areas are different.  The best way to calculate a volume of the triangle with sides a, b and c is Heron's formula: S =  p \* (p – a) \* (p - b) \* (p – c) where p = (a + b + c) / 2.  Input Format  First line of each test file contains single integer n. n lines follow with ai, bi and ci on each separated by single spaces.  Constraints 1 ≤ n ≤ 100  1 ≤ ai, bi, ci ≤ 70  ai + bi > ci, ai + ci > bi and bi + ci > ai  Output Format  Print exactly n lines. On each line print 3 integers separated by single spaces, which are ai, bi and ci of the corresponding triangle.  Sample Input 3  7 24 25  5 12 13  3 4 5  Sample Output 3 4 5  5 12 13  7 24 25  Explanation  The square of the first triangle is 84. The square of the second triangle is 30. The square of the third triangle is 6. So, the sorted order is the reverse one. | |

Department of Computer Science and Engineering, Rajalakshmi Engineering College

|  |
| --- |
| **Program:**  **#include <stdio.h>**  **#include <math.h>**  **Double calArea(int a,int b,int c){**  **Double p = (a + b + c) / 2.0;**  **Double area = sqrt(p\*(p-a)\*(p-b)\*(p-c));**  **Return area;**  **}**  **Int main(){**  **Int n;**  **Scanf(“%d”,&n);**  **Int triangles[n][3];**  **Double areas[n];**  **For(int I = 0;I < n;i++){**  **Scanf(“%d %d %d”,&triangles[i][0],&triangles[i][1],&triangles[i][2]);**  **Areas[i] = calArea(triangles[i][0],triangles[i][1],triangles[i][2]);**  **}**  **For(int I = 0; I < n – 1; i++){**  **For(int j = I + 1;j < n;j++){**  **If(areas[i] > areas[j]){**  **Double tempArea = areas[i];**  **Areas[i] = areas[j];**  **Areas[j] = tempArea;**  **Int temp[3];**  **For(int k = 0; k < 3;k++){**  **Temp[k] = triangles[i][k];**  **Triangles[i][k] = triangles[j][k];**  **Triangles[j][k] = temp[k];**  **}**  **}**  **}**  **}**  **For(int I = 0;I < n;i++){**  **Printf(“%d %d %d\n”,triangles[i][0],triangles[i][1],triangles[i][2]);**  **}**  **Return 0;**  **}** |

Department of Computer Science and Engineering, Rajalakshmi Engineering College

Department of Computer Science and Engineering, Rajalakshmi Engineering College