Name	Shubhan Singh
UID no.	2022300118
Experiment No.	4-C

PROBLEM STATEMENT :	Shopping Cart: Create a 3D array named as cart which stores the cost of items purchased. Each conveyor belt holds 3 carts at a time. Each cart should contain Perishable and Non-perishable category items. Find out: a) Total cost of each cart b) Find out all perishable items sold (on 3 carts) c) Find out costliest non-perishable item sold
THEORY:	Multidimensional arrays in Java: Multidimensional arrays in Java are arrays that contain other arrays as elements, forming a matrix or a table of values. They are useful when dealing with complex data that has multiple dimensions and when a single array is not enough to represent the data efficiently. In Java, multidimensional arrays can be created in different ways, such as using the new keyword or by initializing them with values. Multidimensional arrays in Java can be of different dimensions, such as two-dimensional, three-dimensional, or higher-dimensional arrays. Two-dimensional arrays are the most common type of multidimensional array, and they are often used to represent data in the form of a matrix or a table. For example, a two-dimensional array can be used to store the grades of students in different subjects. Multidimensional arrays in Java are accessed using nested loops. Each dimension of the array requires a separate loop, and the number of loops depends on the number of dimensions of the array. Accessing elements in a multidimensional array can be more complicated than accessing elements in a one-dimensional array, but it provides a more flexible way to store and manipulate data. One advantage of using multidimensional arrays in Java is that they allow for the efficient use of memory. Since the elements in a multidimensional array are stored in contiguous memory locations, accessing them is faster than accessing non-contiguous memory locations. Additionally, using multidimensional arrays can lead to more readable code, especially when

	dealing with complex data structures. Multidimensional arrays in Java can also be used in various applications, such as image processing, scientific computing, and data analysis. For example, a three-dimensional array can be used to represent a volume of data, while a four-dimensional array can be used to represent a video sequence.
PROGRAM:	

```
import java.util.Scanner;

public class ShoppingCart {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);

        // Create a 3D array to store shopping cart data
        int[][][] cart =new int[3][2][];

        // Ask for input from the user for each cart
        int item_count;
        for(int i=0;i<3;i++){
            System.out.printf("Enter number of perishable
items in cart %d: ",i+1);
            item_count=sc.nextInt();

            // Create a 1D array to store perishable item
costs
            cart[i][0]=new int[item_count];

            // Ask for input from the user for each
perishable item
            System.out.println("Enter cost of these items,
one by one (Enter 0 if the item was not sold):");
            for(int x=0;x<item_count;x++){
                cart[i][0][x]=sc.nextInt();
            }

            System.out.printf("Enter number of non-perishable
items in cart %d: ",i+1);
            item_count=sc.nextInt();

            // Create a 1D array to store non-perishable item
costs
            cart[i][1]=new int[item_count];

            // Ask for input from the user for each non-
perishable item
            System.out.println("Enter cost of these items,
one by one (Enter 0 if the item was not sold):");
            for(int x=0;x<item_count;x++){
                cart[i][1][x]=sc.nextInt();
            }
        }

        int sum,perishable_sold=0,costliestnp=0;
```

```
        for(int i=0;i<3;i++){
            sum=0;

            // Iterate through each item in the cart
            for(int j=0;j<2;j++){
                for(int k=0;k<cart[i][j].length;k++){
                    if(j==0){ // Check if the item is
perishable
                        if(cart[i][j][k]!=0) {
                            perishable_sold++;
                        }
                    }
                    sum+=cart[i][j][k];
                    if(j==1){ // Check if the item is non-
perishable
                        if(cart[i][j][k]>costliestnp){
                            costliestnp=cart[i][j][k];
                        }
                    }
                }
            }

            // Print the total cost of the cart
            System.out.printf("The total cost of cart %d is
%d\n",i+1,sum);
        }

        // Print the number of perishable items sold and the
costliest non-perishable item sold
        System.out.println("The number of perishable items
sold is: "+perishable_sold);
        System.out.println("The costliest non perishable item
sold is: " + costliestnp);
    }
}
```

RESULT:

```
Enter number of perishable items in cart 1: 3
Enter cost of these items, one by one (Enter 0 if the item was not sold):
45 0 235
Enter number of non-perishable items in cart 1: 5
Enter cost of these items, one by one (Enter 0 if the item was not sold):
234 54 23 0 98
Enter number of perishable items in cart 2: 5
Enter cost of these items, one by one (Enter 0 if the item was not sold):
45 65 23 654 2321
Enter number of non-perishable items in cart 2: 2
Enter cost of these items, one by one (Enter 0 if the item was not sold):
34 542
Enter number of perishable items in cart 3: 1
Enter cost of these items, one by one (Enter 0 if the item was not sold):
555
Enter number of non-perishable items in cart 3: 3
Enter cost of these items, one by one (Enter 0 if the item was not sold):
342 324 54
The total cost of cart 1 is 689
The total cost of cart 2 is 3684
The total cost of cart 3 is 1275
The number of perishable items sold is: 8
The costliest non perishable item sold is: 542

Process finished with exit code 0
```