```
enum ProductTypes { Sandwhich, Drink };
class Product
{
    private ProductTypes Ptype;
    private double cost;
    private double price;
    private string name;
    private double weight;
    private double caloricValue;

    public Product(ProductTypes Ptype, string name, int weight, int cost, double caloricValue)
    {
        this.Ptype = Ptype;
        this.name = name;
        this.weight = weight;
        this.cost = cost;
        this.price = cost + (30 / 100) * cost;
        this.caloricValue = caloricValue;
    }

    public void SetPrice(double price)
    {
        this.price = price;
    }

    public ProductTypes GetPtye()
    {
        return this.Ptype;
    }

    public double GetPrice()
    {
        return this.price;
    }

    public double GetCost()
    {
        return this.cost;
    }

    public double GetCaloricValue()
    {
        return this.caloricValue * 300;
    }

    public void SetSellPrice(double price)
    {
        this.price = price;
    }

    public override string ToString()
    {
        return this.Ptype + ", " + this.name + ", " + this.weight + ", " + this.cost + ", " +
this.price;
    }
}
```

המשך למטה

```
class DailySell
{

    private DateTime saleDay;
    private Product[] sandwiches;
    private Product[] drinks;
    private int totalSandwiches;
    private int totalDrinks;
    private double totalCost;
    private double income;

    public DailySell (DateTime saleDay)
    {
        this.saleDay = saleDay;
        this.totalCost = 0;
    }

    public void BuyAtMorning(int totalSandwiches, int totalDrinks)
    {
        this.sandwiches = new Product[totalSandwiches];
        this.totalSandwiches = totalSandwiches;

        for (int i = 0; i<sandwiches.Length; i++)
        {
            if (i % 2 == 0)
            {
                sandwiches[i] = new Product(ProductTypes.Sandwhich, "Tuna", 200, 20, 30);
                this.totalCost += sandwiches[i].GetCost();
            }
            else if (i % 3 == 0)
            {
                sandwiches[i] = new Product(ProductTypes.Sandwhich, "Egg", 200, 25, 40);
                this.totalCost += sandwiches[i].GetCost();
            }
            else
            {
                sandwiches[i] = new Product(ProductTypes.Sandwhich, "Salmon", 200, 30, 50);
                this.totalCost += sandwiches[i].GetCost();
            }
        }

        this.drinks = new Product[totalDrinks];
        this.totalDrinks = totalDrinks;

        for (int i = 0; i < drinks.Length; i++)
        {
            if (i % 2 == 0)
                drinks[i] = new Product(ProductTypes.Drink, "Sprite", 300, 7, 10);
            else if (i % 3 == 0)
                sandwiches[i] = new Product(ProductTypes.Drink, "Coke", 300, 10, 20);
            else
                sandwiches[i] = new Product(ProductTypes.Drink, "Juice", 300, 8, 20);
        }
    }

    public bool SellSandwich()
    {
        if (sandwiches[totalSandwiches].GetCaloricValue() > 200)
        {
            Console.WriteLine("WARNING! THIS PRODUCT CALORIC VALUE IS HIGH!");
        }

        if (totalSandwiches > 0)
        {
            income += this.sandwiches[totalSandwiches - 1].GetPrice();
            totalSandwiches--;
            return true;
        }
```

```
        return false;
    }

    public bool SellDrink()
    {
        if (totalDrinks > 0)
        {
            income += this.drinks[totalDrinks - 1].GetPrice();
            totalDrinks--;
            return true;
        }
        return false;
    }

    public double GetCurrentProfit()
    {
        return totalCost - income;
    }

    public double GetCurrentWeight()
    {
        return (sandwiches.Length * 200) + (drinks.Length * 300);
    }

    public void ShowAll()
    {
        Console.WriteLine("Sandwhiches:");
        for (int i = 0; i < sandwiches.Length; i++)
        {
            Console.WriteLine(sandwiches[i]);
        }
        Console.WriteLine("Drinks:");
        for (int i = 0; i < drinks.Length; i++)
        {
            Console.WriteLine(drinks[i]);
        }
    }

    public Product GetSandwich(int i)
    {
        return sandwiches[i];
    }

    public Product[] GetDrinks()
    {
        return this.drinks;
    }

    public int GetTotalDrinks()
    {
        return totalDrinks;
    }

    public override string ToString()
    {
        string s = saleDay + ": " + "Sandwiches: " + totalSandwiches + ", Drinks: " +
totalDrinks + ", Costs: " + totalCost + ", Income: " + this.income;
        return s;
    }



}
```

המשך למטה

```
internal class Program
{
    public static void AddXNIS(Product[] a, int length, int x)
    {
        for (int i = 0; i < length; i++)
        {
            a[i].SetPrice(a[i].GetPrice()+3);
        }
    }

    static void Main(string[] args)
    {

        DailySell ds = new DailySell(new DateTime(2021, 12, 28));

        ds.BuyAtMorning(10, 5, 20, 8);

        Console.WriteLine(ds);
        Console.WriteLine("Current Profit:" + ds.GetCurrentProfit());

        ds.SellDrink();
        ds.SellDrink();
        ds.SellDrink();
        ds.SellDrink();
        ds.SellSandwich();
        ds.SellSandwich();

        Console.WriteLine(ds);
        Console.WriteLine("Current Profit:" + ds.GetCurrentProfit());

        ds.GetSandwich(0).SetSellPrice(12);

        ds.ShowAll();

        Console.ReadLine();

        AddXNIS(ds.GetDrinks(), ds.GetTotalDrinks(), 3);

        ds.SellDrink();
        Console.WriteLine(ds);
        Console.WriteLine("Current Profit:" + ds.GetCurrentProfit());
        ds.ShowAll();

    }
}
```