SORU 1

```
/// <summary>
/// The Subject abstract class
/// </summary>
abstract class Veggies
{
    private double _pricePerPound;
    private List<IRestaurant> _restaurants = new List<IRestaurant>();

    public Veggies(double pricePerPound)
    {
        _pricePerPound = pricePerPound;
    }

    public void Attach(IRestaurant restaurant)
    {
        _restaurants.Add(restaurant);
    }

    public void Detach(IRestaurant restaurant)
    {
        _restaurants.Remove(restaurant);
    }

    public void Notify()
    {
        foreach (IRestaurant restaurant in _restaurants)
        {
            restaurant.Update(this);
        }
        Console.WriteLine("");
    }

    public double PricePerPound
    {
        get { return _pricePerPound; }
        set
        {
            if (_pricePerPound != value)
            {
                _pricePerPound = value;
                Notify(); //Automatically notify price changes
            }
        }
    }
}
```

SLOC = 37, LSLOC = 14, PSLOC = 40

SORU 2

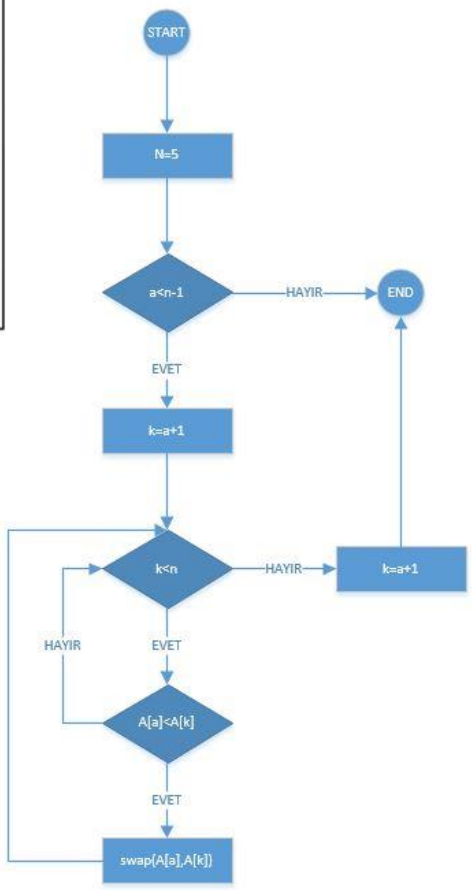
```
class Program
{
    static void Main(string[] args)
    {
        SandwichMenu sandwichMenu = new SandwichMenu();

        // Initialize with default sandwiches
        sandwichMenu["BLT"] = new Sandwich("Wheat", "Bacon", "", "Lettuce, Tomato");
        sandwichMenu["PB&J"] = new Sandwich("White", "", "", "Peanut Butter, Jelly");
        sandwichMenu["Turkey"] = new Sandwich("Rye", "Turkey", "Swiss", "Lettuce");

        // Deli manager adds custom sandwiches
        sandwichMenu["LoadedBLT"] = new Sandwich("Wheat", "Turkey, Bacon","American");
        sandwichMenu["ThreeMeatCombo"] = new Sandwich("Rye", "Turkey, Ham, Salami");
        sandwichMenu["Vegetarian"] = new Sandwich("Wheat", "", "", "Lettuce, Onion");

        // Now we can clone these sandwiches
        Sandwich sandwich1 = sandwichMenu["BLT"].Clone() as Sandwich;
        Sandwich sandwich2 = sandwichMenu["ThreeMeatCombo"].Clone() as Sandwich;
        Sandwich sandwich3 = sandwichMenu["Vegetarian"].Clone() as Sandwich;

        // Wait for user
        Console.ReadKey();
    }
}
```

LOC =17 , PLOC = 21 , LLOC =11

SORU 3

```
a = 0;
n=5;

while (a<n-1) do
k = a + 1;

while (k<n) do

if A[a]<A[k] then
swap(A[a], A[k]);

end do;
a=a+1;

end do;
```

DK = Y – N + 2*P
Y = Yol , N = Node , P = Bağlı Olmayan Yol
Y = 11 , N = 9 , P = 1
DK = 11 – 9 + 2 = 4



SORU 4

```
int eşitlik;
while(t<150)
   if(x[t] % 2 == 0){
        esitlik = 0;
}
   else {
        esitlik = 1;
}
   switch(eşitlik) {
        case 0:
           Console.WriteLine("Sayı çifttir");
        case 1:
           Console.WriteLine("Sayı tektir");
        default:
           Console.WriteLine("Beklenmeyen hata
oluştu");
   t++;
}
bool c = true;
```

DK = Y – N + 2*P
Y = Yol , N = Node , P = Bağlı Olmayan Yol
Y = 17 , N = 14 , P = 1
DK = 17 – 14 + 2 = 5