```
using System;

namespace prtb6
{
    public class Car
    {
        private int currSpeed;
        private string petName;

        private Car() { }

        public Car(string name, int speed)
        {
            petName = name;
            currSpeed = speed;
        }

        public override string ToString()
        {
            return string.Format("{0} is going {1}MPH", petName,
currSpeed);
        }
    }

    internal class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("A, Number of Generations:");
            Console.WriteLine("This OS has {0} object generation \n",
(GC.MaxGeneration + 1));

            Car refToMyCar = new Car("Zippy", 100);
            Console.WriteLine(refToMyCar.ToString());

            Console.WriteLine("\nB, Generation Number of target object");
            Console.WriteLine("Generation of refToMyCar is: {0}",
GC.GetGeneration(refToMyCar));

            object[] tonsOfObjects1 = new object[50000];

            for (int i = 0; i < 50000; i++)
            {
                tonsOfObjects1[i] = new object();
            }

            GC.Collect(0);
            GC.WaitForPendingFinalizers();
            Console.WriteLine("Generation of refToMyCar is: {0}",
GC.GetGeneration(refToMyCar));

            object[] tonsOfObjects2 = new object[50000];

            for (int i = 0; i < 50000; i++)
```

```
            {
                tonsOfObjects2[i] = new object();
            }

            GC.Collect(0);
            GC.WaitForPendingFinalizers();
            Console.WriteLine("Generation of refToMyCar is: {0}\n",
GC.GetGeneration(refToMyCar));

            Console.WriteLine("C, Number of bytes allocated:");
            Console.WriteLine("Estimated bytes on heap memory: {0}",
GC.GetTotalMemory(false));
            Console.ReadLine();
        }
    }
}
```