

Testing:

While going through the code I ran into a few things which slowed me down such as making sure that the error checking work against abnormal and extradentary code to make sure the programs robust. At first the error checking didn't work and looked like this

```
Console.Write("Enter the races gender : ");
gender = Console.ReadLine();

if (gender == "male" || gender == "Male")
{
    gezer = 2;
    MR = 5;
}

else if (gender == "female" || gender == "Female")
{
    gezer = 2;
    FR = 5;
}

else
{
    Console.Write("What you've entered is invalid. Click enter to continue ");
    Console.ReadLine();
}
```

It was similar to this yet with a few key differences. The code would run the first 2 lines yet had no error checking at all. This was easily solved while a while loop which was able to break the while loop plus was able to produce a feature which would specify weather someone had beaten the male or female record based off what the gender equals. This is the finished version looks like.

```

string gender;
int gezer = 0;
int MR = 0;
int FR = 0;

while (gezer < 1)
{
    Console.WriteLine("Enter the races gender : ");
    gender = Console.ReadLine();

    if (gender == "male" || gender == "Male")
    {
        gezer = 2;
        MR = 5;
    }

    else if (gender == "female" || gender == "Female")
    {
        gezer = 2;
        FR = 5;
    }

    else
    {
        Console.WriteLine("What you've entered is invalid. Click enter to continue ");
        Console.ReadLine();
    }
}

```

This is what the finished version looks like which protects against people abnormal and extradentary data making it robust against certain attacks which would keep data safe.

```

int num_lanes = 0;

List<int> lannes = new List<int>();

while (num_lanes < 1)
{
    Console.WriteLine("Enter the number of lanes ");
    lane = Int32.Parse(Console.ReadLine());

    for (int i = 1; i <= lane; i++)
    {
        lannes.Add(lane);
        num_lanes = 2;
    }
}

```

The code above is the next section which leads allows the user to enter the lanes and when entering the lanes, it will count to the number added which means it doesn't need then user to enter every

lane individually. It also contains a loop which isn't optimised fully so it will break the code if any abnormal or extreme code is typed. This code is followed on by the code below

```
int lanes = 0;

double[] time = new double[0];

for (int lane = 0; lane < lanes; lane++)
{
    Console.WriteLine("Enter time of lanes : ");
    time[lanes] = Convert.ToDouble(Console.ReadLine());
}
```

Which takes the values of the number of lanes and will then loop to the number of lanes. Once all of the times have been entered it will be sorted using this line of code.

```
Array.Sort(time);
```

From there we lead onto the last part of the code. This is the part where we inform the user if any of the times have broken a record. This part could be better yet it fits its purpose.

```

for (int i = 0; i < time.Length; i++)
{
    if (MR == 5)
    {
        if (time[i] < 9.59)
        {
            Console.WriteLine("You have beaten the males world record");
            Console.ReadLine();
        }

        if (time[i] < 9.86)
        {
            Console.WriteLine("You have beaten the males european record");
            Console.ReadLine();
        }

        if (time[i] < 9.87)
        {
            Console.WriteLine("You have beaten the males british record");
            Console.ReadLine();
        }
    }

    if (FR == 5)
    {
        if (time[i] < 10.49)
        {
            Console.WriteLine("You have beaten the males world record");
            Console.ReadLine();
        }

        if (time[i] < 10.73)
        {
            Console.WriteLine("You have beaten the males european record");
            Console.ReadLine();
        }

        if (time[i] < 10.99)
        {
            Console.WriteLine("You have beaten the males british record");
            Console.ReadLine();
        }
    }
}
}

```

Depending on what was put at the beginning will dictate what outcome is met here. Right now it will display if a record has beaten a previous one but I would have liked it to display by how much and which one beat it. These are things which im going to remember for future projects as it will improve the quality and professionalism of the code.