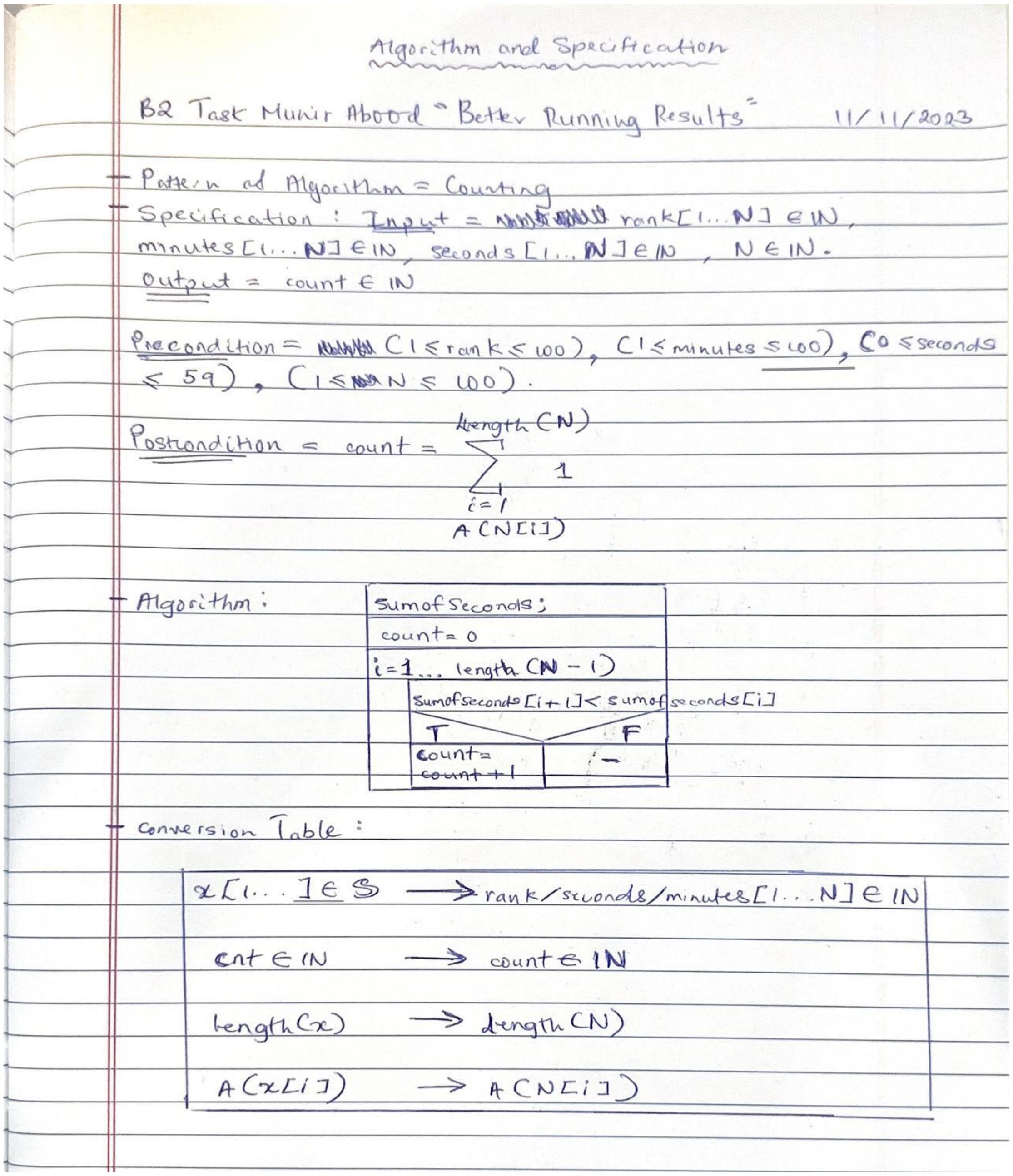
Code:

```
namespace b2_task_Munir
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int N = Convert.ToInt32(Console.ReadLine());
            int[] rank = new int[N];
            int[] minutes = new int[N];
            int[] seconds = new int[N];

            for (int i = 0; i < N; i++)
            {
                string input = Console.ReadLine();
                rank[i] = Convert.ToInt32(input.Split(" ")[0]);
                minutes[i] = Convert.ToInt32(input.Split(" ")[1]);
                seconds[i] = Convert.ToInt32(input.Split(" ")[2]);

            }

            int[] sumofseconds = new int[N];

            for (int i = 0;i < N; i++)
            {
                sumofseconds[i] = (minutes[i] * 60) + seconds[i];
            }

            int count = 0;

            for (int i = 0; i < N-1; i++)
            {
                if (sumofseconds[i+1] < sumofseconds[i])
                {
                    count++;
                }
            }

            Console.WriteLine(count);
        }
    }
}
```

Algorithm and Specification:



Algorithm and Specification

B2 Task Munir Abood "Better Running Results" 11/11/2023

- Pattern of Algorithm = Counting
- Specification : Input = rank[1...N] $\in \mathbb{N}$, minutes[1...N] $\in \mathbb{N}$, seconds[1...N] $\in \mathbb{N}$, N $\in \mathbb{N}$.

Output = count $\in \mathbb{N}$

Precondition = (1 $\leq$ rank $\leq$ 100), (1 $\leq$ minutes $\leq$ 100), (0 $\leq$ seconds $\leq$ 59), (1 $\leq$ N $\leq$ 100).

Postcondition = count = $\sum_{\substack{i=1 \\ A(N[i])}}^{length(N)} 1$

- Algorithm :

```
sumofSeconds;
count = 0
i=1... length(N - 1)
    sumofseconds[i+1] < sumofseconds[i]
        T: count = count + 1
        F: —
```

- Conversion Table :

x[1...] ∈ S	→	rank/seconds/minutes[1...N] ∈ IN
cnt ∈ IN	→	count ∈ IN
length(x)	→	length(N)
A(x[i])	→	A(N[i])