Genspark-training - HACKERRANK

1)https://www.hackerrank.com/challenges/plus-minus/problem?isFullScreen=true

```
class Result
{
    public static void Main(string[] args)
    {
        int n = int.Parse(Console.ReadLine().Trim());
        string input = Console.ReadLine();
        string[] parts = input.Split(' ',
StringSplitOptions.RemoveEmptyEntries);
        List<int> calculation = new List<int>();
        foreach (string part in parts)
        {
            calculation.Add(int.Parse(part));
        }
        int pos=0,neg=0,zer=0;
        decimal total = calculation.Count;
        foreach (int i in calculation)
        {
            if(i > 0)
            {
                pos++;
            }
            else if(i<0)
            {
                neg++;
            }
            else
            {
                zer++;
            }
        }
        Console.WriteLine($"{(decimal)pos / total:F6}");
        Console.WriteLine($"{(decimal)neg / total:F6}");
        Console.WriteLine($"{(decimal)zer / total:F6}");
    }
}
```

2)https://www.hackerrank.com/challenges/staircase/problem?isFullScreen=true

```
public static void staircase(int n)
    {
        for (int i = 1; i <= n; i++)
        {
            for (int j = 0; j < n - i; j++)
            {
                Console.Write(' ');
            }

            for (int k = 0; k < i; k++)
            {
                Console.Write('#');
            }
            Console.WriteLine();
        }
    }
```

3)https://www.hackerrank.com/challenges/mini-max-sum/problem?isFullScreen=true

```
public static void miniMaxSum(List<int> arr)
    {
        arr.Sort();
        long l=0,r=0;
        for(int i=0;i<4;i++)
        {
            l=l+arr[i];
        }
        for(int i=1;i<5;i++)
        {
            r=r+arr[i];
        }
        Console.WriteLine(l+" "+r);
    }
```

4)https://www.hackerrank.com/challenges/birthday-cake-candles/problem?isFullScreen=true

```
public static int birthdayCakeCandles(List<int> candles)
    {
        var frequency = candles.GroupBy(x => x).ToDictionary(x =>
x.Key, x => x.Count());
        int maxheight = frequency.Keys.Max();
        return frequency[maxheight];
    }
```

5)https://www.hackerrank.com/challenges/time-conversion/problem?isFullScreen=true

```
public static string timeConversion(string s)
    {
        string period = s.Substring(s.Length - 2);
        String[] timeparts = s.Substring(0,8).Split(':');
        int hour = int.Parse(timeparts[0]);
        int minute = int.Parse(timeparts[1]);
        int second = int.Parse(timeparts[2]);
        if(period == "AM")
        {
                if(hour == 12)
                {
                    hour = 0;
                }
        }
        else
        {
            if(hour != 12 )
            {
                hour += 12;
            }
        }
        return  $"{hour:D2}:{minute:D2}:{second:D2}";
    }
```

6)https://www.hackerrank.com/challenges/grading/problem?isFullScreen=true

```
public static List<int> gradingStudents(List<int> grades)
    {
        List<int> result = new List<int>();

        foreach (int grade in grades)
        {
        if (grade < 38)
        {
            result.Add(grade);
        }
        else
        {
            int nextMultipleOf5 = ((grade + 4) / 5) * 5;
            if (nextMultipleOf5 - grade < 3)
            {
                result.Add(nextMultipleOf5);
            }
            else
            {
                result.Add(grade);
            }
        }
    }
    return result;
    }
```

7)https://www.hackerrank.com/challenges/apple-and-orange/problem?isFullScreen=true

```
public static void countApplesAndOranges(int s, int t, int a, int b, List<int> apples, List<int> oranges)
    {
        int m = apples.Count();
        int n = oranges.Count();
        int count_app=0;
        int count_orng=0;
        foreach(int apple in apples)
        {
            if(a + apple >= s && a+apple <= t)
            {
                count_app++;
            }
        }
        foreach(int orange in oranges)
        {
            if(b + orange >= s && b+orange <= t)
            {
                count_orng++;
            }
        }
        Console.WriteLine(count_app);
        Console.WriteLine(count_orng);

    }
```

8)https://www.hackerrank.com/challenges/kangaroo/problem?isFullScreen=true

```
public static string kangaroo(int x1, int v1, int x2, int v2)
        {
            if(x2 > x1 && v2 > v1 )
            {
                return "NO";
            }
            else
            {
                if((v1>v2) && (x1-x2)%(v1-v2)==0)
        {
          return "YES";
        }
        else
        {
          return "NO";
        }
      }
    }
```

9)https://www.hackerrank.com/challenges/between-two-sets/problem?isFullScreen=true

```
public static int getTotalX(List<int> a, List<int> b)
  {
    int lcm = a[0];
    foreach(int i in a.Skip(1))
    {
      lcm = LCM(lcm,i);
    }
    int gcd = b[0];
    foreach(int i in b.Skip(1))
    {
      gcd = GCD(gcd,i);
    }
    int count=0;
    for(int i=lcm;i<=gcd;i+=lcm)
    {
      if(gcd%i==0)
      {
        count++;
      }
    }
    return count;
  }
  public static int GCD(int x,int y)
  {
    while(y!=0)
    {
      int temp = y;
      y = x%y;
      x = temp;
    }
    return x;
  }
```

```
    public static int LCM(int x,int y)
    {
        return (x*y)/GCD(x,y);
    }
```

10)https://www.hackerrank.com/challenges/breaking-best-and-worst-records/problem?isFullScreen=true

```
public static List<int> breakingRecords(List<int> scores)
    {
        int n=scores.Count();
        int min=scores[0];
        int max=scores[0];
        int h=0,l=0;
        for(int i=0;i<n;i++)
        {
            if(scores[i]<min)
            {
                l++;
                min=scores[i];
            }
            if(scores[i]>max)
            {
                h++;
                max=scores[i];
            }
        }
        List<int> result = new List<int> { h, l };
    return result;
    }
```