

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;

namespace Exercise2
{
    class Program
    {
        const int SEARCH_NUMBER_BYTES = 5;
        static void Proces(
            byte[] buffer,
            byte[] search,
            byte[] searchData,
            ref byte[] currentData,
            ref int index
        )
        {
            int len = buffer.Length;
            int found = 0;
            string text = Encoding.ASCII.GetString(search);
            int n = SEARCH_NUMBER_BYTES - 1;
            for (int i=0;i<len;i++)
            {
                currentData[buffer[i]]++;
                index++;

                if (searchData[buffer[i]]>0)
                {
                    if(SEARCH_NUMBER_BYTES==index)
                    {
                        bool matched = true;
                        foreach (byte c in search)
                        {
                            if(searchData[c] != currentData[c])
                            {
                                matched = false;
                                break;
                            }
                        }
                        if(matched)
                        {
                            Console.WriteLine("{2}) {0}=={1}",
                                text,
                                Encoding.ASCII.GetString(buffer, i - n, SEARCH_NUMBER_BYTES),
                                found
                            );
                            found++;
                        }
                    }
                }

                if(i>= n && currentData[buffer[i - n]]>0)
                {
                    currentData[buffer[i - n]]--;
                    index--;
                }
            }

            if (found == 0)
            {
                Console.WriteLine("Nothing found");
            }
        }
        static void Find(string text, string fileName = "data.txt")
        {

```

```

if(File.Exists(fileName))
{
    byte[] search = Encoding.ASCII.GetBytes(text);
    byte[] searchData = new byte[256];
    byte[] currentData = new byte[256];
    foreach (byte c in search)
    {
        searchData[c] = (byte)(searchData[c]>0 ? searchData[c] + 1 : 1);
    }

    FileStream fs = new FileStream(fileName, FileMode.Open, FileAccess.Read);
    int bufferSize = 3;
    byte[] buffer;
    int index = 0;

    using (MemoryStream memoryStream = new MemoryStream())
    {
        using (Stream input = File.OpenRead(fileName))
        {
            byte[] buf = new byte[bufferSize]; // 32K buffer for example
            int bytesRead;
            while ((bytesRead = input.Read(buf, 0, buf.Length)) > 0)
            {
                memoryStream.Write(buf, 0, bytesRead);
            }
            memoryStream.Position = 0;
            buffer = memoryStream.GetBuffer();
            Procces(buffer, search, searchData, ref currentData, ref index);
        }
    }
}
else
{
    Console.WriteLine("File not exists");
}
}
static Boolean isAlphaNumeric(string strToCheck)
{
    Regex rg = new Regex(@"^[a-zA-Z0-9\s,]*$");
    return rg.IsMatch(strToCheck);
}
static void Main(string[] args)
{
    Console.Write("Search Data (5 characters required) : ");
    string input = Console.ReadLine();
    if(string.IsNullOrEmpty(input))
    {
        Console.WriteLine("Invalid Input - empty string");
    }
    else if(input.Length!=5)
    {
        Console.WriteLine("Invalid input - must be exactly 5 chars");
    }
    else if(!isAlphaNumeric(input))
    {
        Console.WriteLine("Invalid input - only alpha numeric allowed");
    }
    else
    {
        Find(input);
    }

    Console.Write("Press any key to finish");
    Console.ReadKey();
}
}
}

```