


```

char[] firstSequence = new char[prizeToPay.Length];
char[] secondSequence = new char[prizeToPay.Length];

// Best programming principle (good practice): KISS
// Keep It Simply Stupid.
// If number at position is exactly 4 we can split it as 3 + 1
// otherwise leave number 'as is' and add 0 to second check.

for (int index = 0; index <= prizeToPay.Length - 1; index++)
{
    char currentNumber = prizeToPay[index];

    if (currentNumber == '4')
    {
        firstSequence[index] = '3';
        secondSequence[index] = '1';
    }
    else
    {
        firstSequence[index] = currentNumber;
        secondSequence[index] = '0';
    }
}

// Although solving mathematical operations on chars feels wrong
we need to do it.
string firstCheck = string.Join("",
firstSequence).TrimStart('0');
string secondCheck = string.Join("",
secondSequence).TrimStart('0');

Console.WriteLine($"Case #{i}: {firstCheck.ToString()}
{secondCheck}");

    }
}
}
}

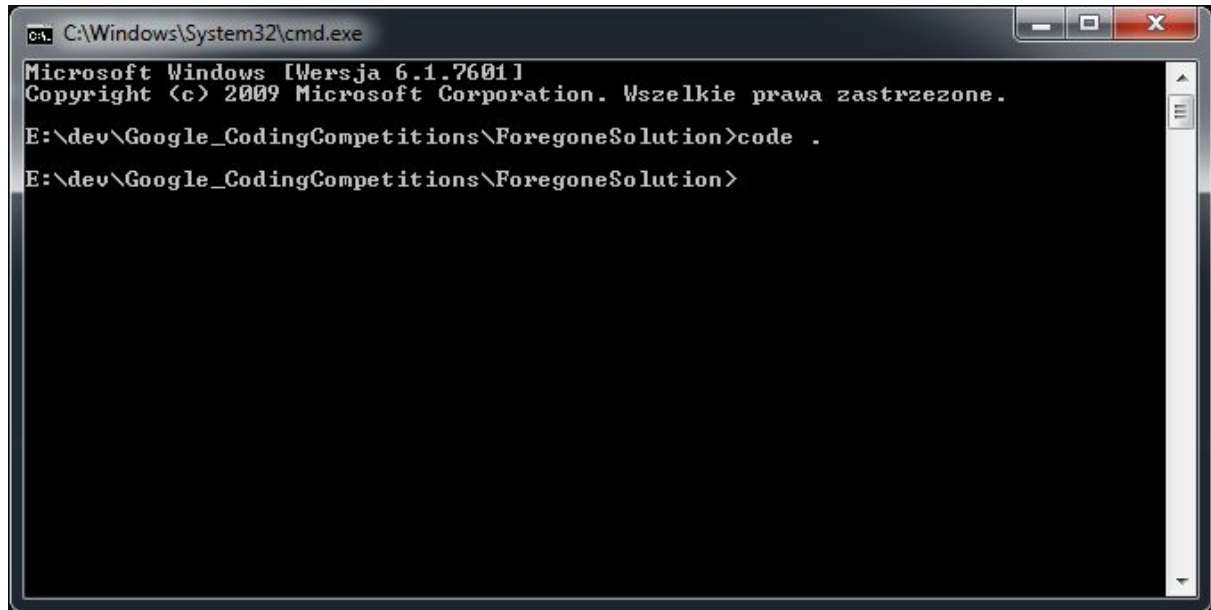
/*
 * Extra info:
 * string/char []
 * Unlike python using quotations marks and apostrophes is not
exchangable.
 * 'a' <= char

```


solution has MANY flaws.

```
*     }  
*   }  
*  
*   string.join(separator, values)  
*   Just example here:  
*  
*   string[] elementy = new string[] { "Wlazl kotek", "na plotek",  
"costam", "go mlotek."};  
*   string spacja = " ";  
*   string wierszyk = string.Join(spacja, elementy);  
*  
*   Console.WriteLine(wierszyk);  
*   Wlazl kotek na plotek costam go mlotek.  
*/
```

Odpalenie vs code w folderze z programem (to jest default, nie dodawałem nic więcej po instalce)



```
C:\Windows\System32\cmd.exe

Microsoft Windows [Wersja 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Wszelkie prawa zastrzeżone.

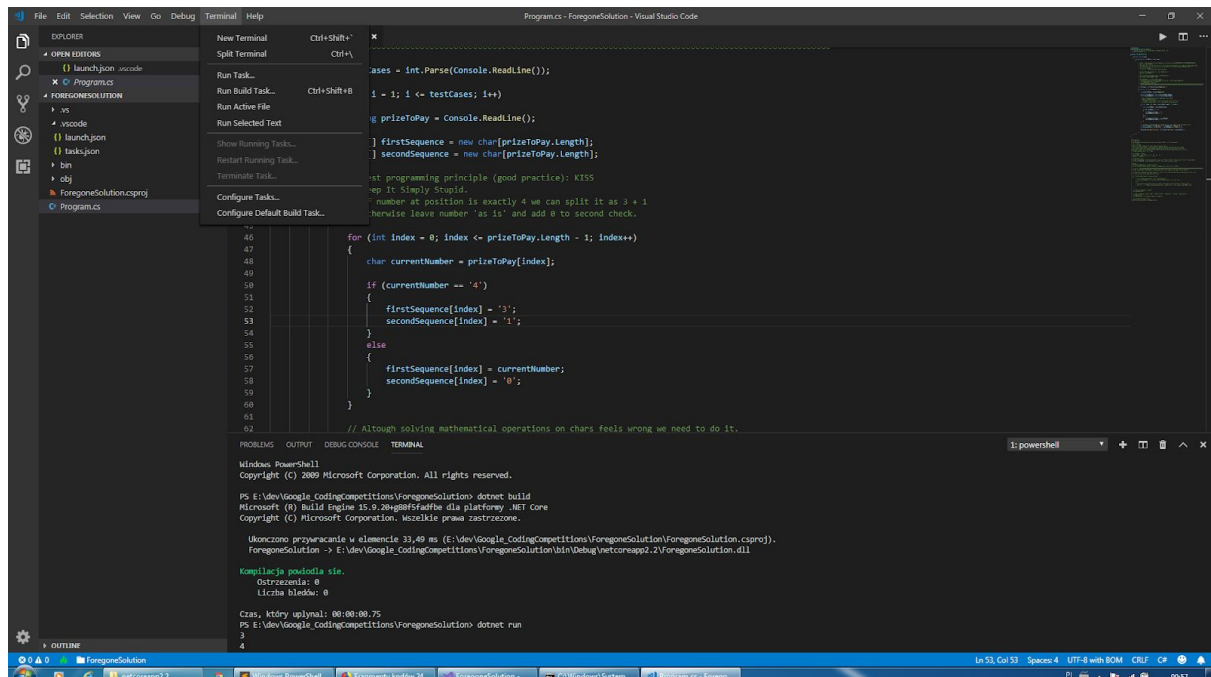
E:\dev\Google_CodingCompetitions\ForegoneSolution>code .

E:\dev\Google_CodingCompetitions\ForegoneSolution>
```

Terminal VS (można też użyć zwykłego cmd, VS powinien sam z siebie pociągnąć kilka pomocnych paczek)

U Alberta ściągano też .NET SDK (Software Developer Kit)

To już waży ~500mb. Po ściągnięciu trzeba będzie raz zresetować VS code (albo przynajmniej odpalić nowy terminal) - ja nie sprawdzałem :) aby zbudować programik.



```
Program.cs - ForegoneSolution - Visual Studio Code

// ...
45:         cases = int.Parse(Console.ReadLine());
46:         i = 1; i <= testCases; i++
47:         {
48:             prizeToPay = Console.ReadLine();
49:             firstSequence = new char[prizeToPay.Length];
50:             secondSequence = new char[prizeToPay.Length];
51:             // programming principle (good practice): KISS
52:             // It's Simply Stupid.
53:             // number at position is exactly 4 we can split it as 3 + 1
54:             // herwise leave number 'as is' and add 0 to second check.
55:             for (int index = 0; index <= prizeToPay.Length - 1; index++)
56:             {
57:                 char currentNumber = prizeToPay[index];
58:                 if (currentNumber == '4')
59:                 {
60:                     firstSequence[index] = '3';
61:                     secondSequence[index] = '1';
62:                 }
63:                 else
64:                 {
65:                     firstSequence[index] = currentNumber;
66:                     secondSequence[index] = '0';
67:                 }
68:             }
69:             // Although solving mathematical operations on chars feels wrong we need to do it.
70:         }
71:     }
72: }

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Windows PowerShell
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

PS E:\dev\Google_CodingCompetitions\ForegoneSolution> dotnet build
Microsoft (R) Build Engine 15.9.20-gbf9f7a1be dla platformy .NET Core
Copyright (c) Microsoft Corporation. Wszelkie prawa zastrzeżone.

Ukończono przywracanie w elemencie 33,49 ms (E:\dev\Google_CodingCompetitions\ForegoneSolution\ForegoneSolution.csproj).
ForegoneSolution -> E:\dev\Google_CodingCompetitions\ForegoneSolution\bin\Debug\netcoreapp2.2\ForegoneSolution.dll

Kompilacja powiodła się.
Ostrzeżenia: 0
Liczba błędów: 0

Czas, który upłynął: 00:00:00.75
PS E:\dev\Google_CodingCompetitions\ForegoneSolution> dotnet run
3
4
```

Dysk:\SciezkaDoFolderuZKodem> **Komenda która wprowadzamy**

Build, czyli stworzenie pliku wykonywalnego.

Dla .NET core będzie to <Nazwa_Projektu>.dll

```
E:\dev\Google_CodingCompetitions\ForegoneSolution> dotnet build
Microsoft (R) Build Engine 15.9.20+g88f5fadbfe dla platformy .NET Core
Copyright (C) Microsoft Corporation. Wszelkie prawa zastrzeżone.

    Ukonczono przywracanie w elemencie 33,49 ms
(E:\dev\Google_CodingCompetitions\ForegoneSolution\ForegoneSolution.csproj).
    ForegoneSolution ->
E:\dev\Google_CodingCompetitions\ForegoneSolution\bin\Debug\netcoreapp2.2\ForegoneSolution.dll

Kompilacja powiodła się.
    Ostrzeżenia: 0
    Liczba błędów: 0

Czas, który upłynął: 00:00:00.75
```

Start programu i wprowadzenie inputu 'z palca'

```
E:\dev\Google_CodingCompetitions\ForegoneSolution> dotnet run
3
4
Case #1: 3 1
940
Case #2: 930 10
123123123123123123123213213517972147012347032740832740723409273403287403
2877510237401283470125601740740734017404
Case #3:
123123123123123123123213213517972137012337032730832730723309273303287303
2877510237301283370125601730730733017303
100000100000100000100001000001000001000000000000100000100000000010010001
000101
```

Jezeli komus podejdzie to w mozna w nowym folderze zrobic:

```
E:\dev\Google_CodingCompetitions\NewProject>dotnet new console
Pomyslne utworzono szablon "Console Application".
```

```
Trwa przetwarzanie akcji po utworzeniu...
Trwa uruchamianie polecenia "dotnet restore" dla
E:\dev\Google_CodingCompetition
s\NewProject\NewProject.csproj...
    Trwa przywracanie pakietów dla elementu
E:\dev\Google_CodingCompetitions\NewPr
object\NewProject.csproj...
    Generowanie pliku programu MSBuild
E:\dev\Google_CodingCompetitions\NewProject
\obj\NewProject.csproj.nuget.g.props.
    Generowanie pliku programu MSBuild
E:\dev\Google_CodingCompetitions\NewProject
\obj\NewProject.csproj.nuget.g.targets.
    Ukonczono przywracanie w elemencie 160,01 ms
(E:\dev\Google_CodingCompetitions
\NewProject\NewProject.csproj).
```

Przywracanie powiodlo sie.

```
E:\dev\Google_CodingCompetitions\NewProject>dotnet run
Hello World!
```

I eksperymentujemy :) Nie wybuchnie.

Zeby bylo latwiej w VS code odpalamy View -> Problems:

Edytor podkresli nam bledy jak Word literowki i jeszcze napisze co mu sie nie podoba.

U mnie informacje sa po Polsku bo mam polskiego Windowsa. Da sie to niestety zmienic tylko zmieniajac wersje jezykowa systemu.

Klikajac na problemie, a potem na 'zarowke' po lewej odpalamy menu quickfix.

Np: Debug.WriteLine wymaga 'zaimportowania' System.Diagnostics. Klikamy w podpowiedz z listy, zeby edytor dodal je za nas.

```
Program.cs
1  using System;
2
3  namespace NewProject
4  {
5      0 references
6      class Program
7      {
8          0 references
9          static void Main(string[] args)
10         {
11             OjGlupotyPopisalem();
12             Console.WriteLine("Hello World!");
13             Debug.WriteLine("DUPA");
14         }
15     }
16
```

using System.Diagnostics;

System.Diagnostics.Debug

Generuj zmienną „Debug” -> Generuj właściwość „Program.Debug”

Generuj zmienną „Debug” -> Generuj pole „Program.Debug”

Generuj zmienną „Debug” -> Generuj pole tylko do odczytu „Program.Debug”

Generuj zmienną „Debug” -> Generuj lokalny element „Debug”

Generuj typ „Debug” -> Generuj element class „Debug” w nowym pliku

Generuj typ „Debug” -> Generuj element class „Debug”

Generuj typ „Debug” -> Generuj zagnieżdżony element class „Debug”

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL

Program.cs 3

- Nazwa „OjGlupotyPopisalem” nie istnieje w bieżącym kontekście (CS0103) [NewProject] [10, 13]
- Oczekiwano znaku) (CS1026) [NewProject] [10, 32]
- Nazwa „Debug” nie istnieje w bieżącym kontekście (CS0103) [NewProject] [12, 13]

Dygresja:

Printy z Debuga są widoczne w wersji developerskiej (debug), ale jeżeli kiedyś zrobimy wersję dla klientów (release) nie będą nawet skompilowane. Program nie będzie nimi ‘zasmiecony’.