

מציאת מילים כפולות בטקסט (התוכנית המלאה - הצעה שלי)

```
using System;
namespace strings
{
    class MainClass
    {
        public static void Main(string[] args)
        {
            string current_word, next_word;
            int p=0; //our progress position starts at zero
            int l; //word's length
            int s_index;
            int pair_count = 0;
            Console.WriteLine("Please enter a sentence or a
                               story");
            string text = Console.ReadLine();
            string space = " ";
            s_index = text.IndexOf(space); //first space
            //length of word is the next space location minus
            //our position
            l = s_index - p;
            if (l > 0) { //more than one word in the text
                current_word = text.Substring(p, l);
                p = p + l + 1;
                while (s_index != -1) {
                    s_index = text.IndexOf(space, p);
                    if (s_index == -1) //last word
                        l = text.Length - p;
                    else
                        //length of word is the next space
                        //location minus our position
                        l = s_index - p;
                    next_word = text.Substring(p, l);
                    if (current_word.Equals(next_word)) {
                        pair_count += 1;
                        Console.WriteLine(p + " ");
                    }
                    p = p + l + 1;
                    current_word = next_word;
                }
            }
            if (pair_count == 0)
                Console.WriteLine("No doubles");
            else
                Console.WriteLine("There were: { 0} doubled
                                   words", pair_count);
        }
    }
}
```

```
    }  
  }  
}
```

שימו לב לשורה בה ישנה ההערה: `//last word`. אם הגענו למצב שאין את הרווח (הבא), סימן שהגענו לסוף המחרוזת, ולכן כדי להחשיב גם מילה זו, משתמשים באורך הכללי של כל המחרוזת `text.Length` ומזה מחסירים את מיקומנו האחרון (של `p`). `p` תמיד שומר את התחלת המילה הבאה שאליה הגענו.