



1st Junior Balkan Olympiad in Informatics
Belgrade 2007

Day 1 – Task 1

Map

After reading the input string and setting the initial position on the map to (0,0), we go through the string and look for **w**, **e**, **n** and **s**. After those letters there are digits, so we read them and convert it to a number. Depending on the letter we determine the direction and increase or decrease the corresponding coordinate, effectively storing the position after every move. Finally we use Pitagora's theorem and compute the distance from the origin.

Example solution in Pascal:



1st Junior Balkan Olympiad in Informatics Belgrade 2007

Day 1 – Task 1

```
program Map;
var
  s: String;
  c: Char;
  i, n, k, x, y: Integer;
begin
  ReadLn (s);
  N := Length (s);
  i := 0;
  x := 0;
  y := 0;
  while (i < n) do begin
    i := i + 1;
    if (s [i] = 'w') or (s [i] = 'e') or (s [i] = 'n') or (s [i] = 's') then begin
      c := s [i];
      k := 0;
      i := i + 1;
      while (Ord (s [i]) >= 48) and (Ord (s [i]) <= 57) do begin
        k := k * 10 + Ord (s [i]) - 48;
        i := i + 1;
      end;

      i := i - 1;
      if c = 'w' then x := x + k;
      if c = 'e' then x := x - k;
      if c = 'n' then y := y + k;
      if c = 's' then y := y - k;
    end;
  end;
  WriteLn (Sqrt (x * x + y * y):0:2);
end.
```

Example solution in C#:



1st Junior Balkan Olympiad in Informatics Belgrade 2007

Day 1 – Task 1

```
using System;

class Map
{
    static void Main(string[] args)
    {
        string s = Console.ReadLine();
        long x = 0;
        long y = 0;
        for (int i = 0; i < s.Length; i++)
            if ((s[i] == 'w') || (s[i] == 'e') || (s[i] == 'n') || (s[i] == 's'))
            {
                char c = s [i];
                i++;
                int n = 0;
                while ((i < s.Length) && (s [i] >= '0') && (s [i] <= '9'))
                {
                    n = n * 10 + (s [i] - '0');
                    i++;
                }
                i--;
                switch (c)
                {
                    case 'w':
                        x = x + n;
                        break;
                    case 'e':
                        x = x - n;
                        break;
                    case 'n':
                        y = y + n;
                        break;
                    case 's':
                        y = y - n;
                        break;
                }

                double dist = Math.Sqrt(x * x + y * y);
                decimal sol = decimal.Round((decimal) dist, 2);
                Console.WriteLine(sol);
            }
    }
}
```