**Title**

-------------------------------------------------------------------------------------------------------------

**Pangrams**

-------------------------------------------------------------------------------------------------------------

**Description**

-------------------------------------------------------------------------------------------------------------

A pangram is a string that contains every letter of the alphabet. Given a sentence determine whether it is a pangram in the English alphabet. Ignore case. Return either pangram or not pangram as appropriate.

**Example**

s = "The quick brown fox jumps over the lazy dog"

The string contains all letters in the English alphabet, so return pangram.

**Function Description**

Complete the function pangrams in the editor below. It should return the string pangram if the input string is a pangram. Otherwise, it should return not pangram.

**pangrams has the following parameter(s):**

string s: a string to test

**Returns**

string: either pangram or not pangram

**Input Format**

A single line with string s.

**Constraints**

0 < length of s <= 103

Each character of s, s[i] ϵ {A – Z, a –z, space}

**Sample Input**

"We promptly judged antique ivory buckles for the next prize"

**Sample Output**

"pangram"

**Sample Input**

"We promptly judged antique ivory buckles for the prize"

**Sample Output**

"not pangram" (Missing x)

-------------------------------------------------------------------------------------------------------------

**Code**

-------------------------------------------------------------------------------------------------------------

package main

import(

    "fmt"

    "strings"

    "regexp"

)

func main(){

    str := "We promptly judged antique ivory buckles for the next prize"

    fmt.Println(pangrams(str))

}

func pangrams(s string) string {

    var result string

    str := strings.ToUpper(s)

    str = regexp.MustCompile(`[^A-Z]`).ReplaceAllString(str,"")

    rns := []rune(str)

    if len(rns)<26{

        result = "not pangram"

    } else{

        m := make(map[rune]bool)

        for \_, v:= range rns{

            m[v] = true

        }

        if len(m)==26{

            result = "pangram"

        } else{

            result = "not pangram"

        }

    }

return result

}

-------------------------------------------------------------------------------------------------------------