Naloga:

Napišite funkcijo, ki kot argument prejme en string. Funkcija naj preveri ali je ta string pangram.Naj vrne True v primeru , da je string pangram, v nasprotnem False.

Pangram je beseda v kateri se pojavijo vse črke abecede (vzemimo angleško abecedo).

Primeri:

```
Input:
Checking: The quick brown fox jumps over the lazy dog.

Output:
It IS pangram
```

In [16]:
```
def is_pangram(word):
    alphabet = {'a': 0, 'b': 0, 'c': 0, 'd': 0, 'e': 0, 'f': 0, 'g': 0, 'h': 0, '
    # alphabet = {key:0 for key in "abcdefghijklmnopqrstuwvxyz"}

    word = word.lower()
    for ch in word:
        if ch in alphabet.keys():
            alphabet[ch] += 1
    print("Character count:")
    print(alphabet)

    for key, value in alphabet.items():
        if value == 0:
            return False
    return True


str_ = "The quick brown fox jumps over the lazy dog."
print("Checking:", str_)
if is_pangram(str_):
    print("It IS pangram")
else:
    print("It ISN'T pangram")
```

```
Checking: The quick brown fox jumps over the lazy dog.
Character count:
{'a': 1, 'b': 1, 'c': 1, 'd': 1, 'e': 3, 'f': 1, 'g': 1, 'h': 2, 'i': 1, 'j':
1, 'k': 1, 'l': 1, 'm': 1, 'n': 1, 'o': 4, 'p': 1, 'q': 1, 'r': 2, 's': 1, 't':
2, 'u': 2, 'w': 1, 'v': 1, 'x': 1, 'y': 1, 'z': 1}
It IS pangram
```

Naloga:

Napišite funkcijo, ki kot argument prejme string besed, ki so med seboj povezane z **-**.

Funkcija naj vrne string, ki je sestavljen iz teh besed, povezanih med seboj z **-**, razvrščenih po abecedi.

Primeri:

```
Input:
Before sort:  brown-orange-red-gray-yellow

Output:
After sort:  brown-gray-orange-red-yellow
```

In [32]:
```
def my_sort(str_):
    words = []
    beseda = ""
    for ch in str_:
        if ch == "-":
            words.append(beseda)
            beseda = ""
        else:
            beseda += ch
    words.append(beseda)

    words = sorted(words)
    final_word = ""
    for word in words:
        final_word += word
        final_word += "-"

    return final_word[:-1]

str_ = "brown-orange-red-gray-yellow"
print("Before sort: ", str_)
str_ = my_sort(str_)
print("After sort: ", str_)
```

```
Before sort:  brown-orange-red-gray-yellow
After sort:  brown-gray-orange-red-yellow
```

In [24]:
```
def my_sort(str_):
    l_ = str_.split("-")
    l_ = sorted(l_)
    l_ = "-".join(l_)
    return l_


str_ = "brown-orange-red-gray-yellow"
print("Before sort: ", str_)
str_ = my_sort(str_)
print("After sort: ", str_)
```

```
Before sort:  brown-orange-red-gray-yellow
After sort:  brown-gray-orange-red-yellow
```

In [ ]:
```

```

Naloga:

Napišite funkcijo, kjer lahko igramo **vislice**.

Funkcija **vislice()** naj ima 2 parametra. Prvi je besedo katero se ugiba in drugi število možnih ugibov. Če števila ugibov ne podamo naj bo default vrednost 10.

Uporabnika konstantno sprašujte naj vnese črko. Nato izpišite iskano besedo. Črke katere je uporabnik uganil izpišite normalno, črke katere še ni uganil pa nadomestite z _.

Dodatno zraven prikazujte katere vse črke je uporabnik že preizkusil.

Če uporabnik besedo uspešno ugani v danih poizkusih naj funkcija vrne vrednost True. V nasprotnem primeru naj vrne vrednost False.

Primeri:

```
Input:
vislice("jabolko")

Output:
Guesses so far [].
What is your guess? a
_ a_ _ _ _ _

Guesses so far ['a'].
What is your guess? e
_ a_ _ _ _ _

Guesses so far ['a', 'e'].
What is your guess? o
_ a_ o_ _ o

Guesses so far ['a', 'e', 'o'].
What is your guess? p
_ a_ o_ _ o

Guesses so far ['a', 'e', 'o', 'p'].
What is your guess? r
_ a_ o_ _ o

Guesses so far ['a', 'e', 'o', 'p', 'r'].
What is your guess? l
_ a_ ol_ o

Guesses so far ['a', 'e', 'o', 'p', 'r', 'l'].
What is your guess? k
_ a_ olko

Guesses so far ['a', 'e', 'o', 'p', 'r', 'l', 'k'].
What is your guess? j
ja_ olko

Guesses so far ['a', 'e', 'o', 'p', 'r', 'l', 'k', 'j'].
What is your guess? b
jabolko
KONEC
True
```

https://pythontutor.com/visualize.html#mode=edit
(https://pythontutor.com/visualize.html#mode=edit)

In [207]:
```
# Rešitev
def vislice(beseda, n=10):
    correct_guesses = []
    all_guesses = []

    try_ = 0
    while try_ < n:
        print()
        # Print vseh ugibanj
        print(f"Guesses so far: {all_guesses}")

        # Zahtevanje inputa
        guess = input(f"What is your guess? ")
        all_guesses.append(guess)

        # Test, če se črka nahaja v iskani besedi
        if guess in beseda:
            correct_guesses.append(guess)

        # Print beseda _ _ _ _ _ _
        beseda_print = ""
        for ch in beseda:
            if ch in correct_guesses:
                beseda_print += ch
            else:
                beseda_print += "_ "
        print(beseda_print)

        # Testiranje ali je igralec zmagal
        if len(set(correct_guesses)) == len(set(beseda)):
            print("KONEC")
            return True

        try_ += 1

    return False


print(vislice("jabolko"))
```

```
Guesses so far [].
What is your guess? a
_ a_ _ _ _ _

Guesses so far ['a'].
What is your guess? e
_ a_ _ _ _ _

Guesses so far ['a', 'e'].
What is your guess? o
_ a_ o_ _ o

Guesses so far ['a', 'e', 'o'].
What is your guess? p
_ a_ o_ _ o
```

```
Guesses so far ['a', 'e', 'o', 'p'].
What is your guess? r
_ a_ o_ _ o

Guesses so far ['a', 'e', 'o', 'p', 'r'].
What is your guess? l
_ a_ ol_ o

Guesses so far ['a', 'e', 'o', 'p', 'r', 'l'].
What is your guess? k
_ a_ olko

Guesses so far ['a', 'e', 'o', 'p', 'r', 'l', 'k'].
What is your guess? j
ja_ olko

Guesses so far ['a', 'e', 'o', 'p', 'r', 'l', 'k', 'j'].
What is your guess? b
jabolko
KONEC
True
```

In [ ]: