TD ALGO1 exercice 8

```
def est_valide(mot, schema):
    for i in range(len(mot)):
        if schema[i] != "?":
            if schema[i] == mot[i]:
            else:
                return false
    return true
```

```
def est_valide(mot: str, schema: str) -> bool:
    if len(mot) != len(schema)
        return False
    i = 0
    while i < len(mot)
        if mot[i] != schema[i] and schema[i] != "?":
            return False
        i += 1
    return True
```

exercice 3

```
def est_prefixe(prefixe, mot):
    for i in range(len(prefixe)):
        if prefixe[i] != mot[i]:
            return False
    return True
```

exercice 10

```
def est_suffixe(suffixe, mot):
    for i in range(len(mot), 0, -1):
        if suffixe[i] != mot[i]:
            return False
    return True
```

exercice 11

```
def contient_tous_mots(mot, mot_2):
    for i in range(len(mot)):
        for j in range(len(mot_2)):
            if mot[i] in mot_2:
            else:
                return False
    return True
```

```
def contient_tous_mots(mot1, mot2):
    l = 0
    for i in range(len(mot2)):
        if mot1[l] == mot[i]
            l = l + 1
    while l < len(mot1):
        for i in range(len(mot2)):
            if mot1[l] == mot[i]
                l = l + 1
    return(l == len(mot1))
```

TD ALGO
TD NUM 8
2

exercice 12

```
def miroir(mot):
    mi = ''
    for i in range(len(mot)-1, -1, -1):
        mi += mot[i]
    return mi
```

exercice 13

```
def palindrome(mot):
    return mot == miroir(mot)
```

exercice 14

```
def trim(m):
    m_trim = ''
    for i in range(len(m)):
        if m[i-1] != ' ':
            m_trim += m[i]
            if m[i+1] == ' ':
                m_trim += m[i+1]
    return m_trim
```