
Combinacions en Esperanto (Versió Sencera)**X45308_ca**

L'Esperanto és una llengua *artificial* és a dir, creada de manera expressa per humans, en contrast amb les altres llengües que s'anomenen *naturals*, com per exemple, el català. Va ser creada pel oftalmòleg jueu polonès **Ludwik Lejzer Zamenhof**. La seva intenció era crear una llengua que no fos de ningú i que, alhora, pogués ser de tothom qui ho volgués, facilitant així l'entesa entre parlants de llengües diferents, de manera que no fos un dels dos parlants qui hagués de cedir cap a l'altra llengua, sinó que tots dos haguessin de cedir i fer servir aquesta llengua *auxiliar*.

L'Esperanto té 16 regles gramaticals, i en aquesta pràctica en farem servir dues:

1. Els *noms* (independentment del gènere) acaben en la lletra **-o**. Per exemple, **viro** (home), o **domo** (casa).
2. Els *adjectius* (independentment del gènere de la paraula que determinen) acaben en la lletra **-a**. Per exemple, **alta** (alt), o **bona** (bo).

La pràctica consisteix en **tres exercicis** que cal fer de manera **acumulativa**. Això vol dir que per a fer la segona part, caldrà fer servir el que hagueu fet (si funciona bé) a la primera part, i per a fer la tercera caldrà fer servir el que haureu fet a la primera i a la segona part.

1 Primera Part

La **primera part** de la pràctica consistirà en, donat una seqüència d'entrada amb aquest format:

```
paraula_1
paraula_2
paraula_3
...
paraula_n
FINO
```

on cada paraula `paraula_i` acabarà en la lletra **o** o en la lletra **a**, fer la funció `entraSeq ()`

PRE Cert.

POST Torna dues llistes: **NOMS** i **ADJECTIUS** tals que **NOMS** conté totes les paraules que apareixen al canal d'entrada i que acaben en la lletra **o** en el mateix ordre en què apareixen al canal d'entrada. La llista **ADJECTIUS** contindrà totes les que acaben en **a**.

Com podeu veure, aquesta funció no rep cap paràmetre, i simplement torna dues llistes, una amb les paraules que acaben en **o** i una altra amb les paraules que acaben en **a**.

2 Segona Part

La **segona part** consisteix a fer una funció que, donats dos vectors `NOMS` i `ADJECTIUS` formi totes les possibles combinacions de *adjectiu* i *nom* (separats per un espai). Concretament, la funció serà aquesta:

```
calculaParelles(NOMS, ADJECTIUS)
```

PRE `NOMS` és un vector que conté paraules que acaben en la lletra **o**. `ADJECTIUS` és un vector que conté paraules que acaben en la lletra **a**.

POST Torna una llista amb totes les combinacions que es pot fer `adjectiu nom`, separats per un espai.

Per exemple, si `ADJECTIUS = ['bona', 'granda']` i `NOMS = ['viro', 'domo']`, el resultat hauria de ser la llista

```
R = ['bona viro', 'granda viro', 'bona domo', 'granda domo'].
```

Com es pot observar, l'ordre de les combinacions ha de respectar l'ordre que hi ha en els vectors `NOMS` i `ADJECTIUS`.

Recordeu que podeu concatenar `strings` fent servir l'operador de suma. Per exemple, `"bona" + " " + "domo"` avalua a `"bona domo"`.

3 Tercera Part

La **tercera part** consisteix a fer una funció que, donats dos vectors `NOMS` i `ADJECTIUS` formi totes les possibles combinacions de *adjectiu1 adjectiu2* i *nom* (separats per un espai).

Concretament, la funció serà aquesta:

```
calculaTriples(NOMS, ADJECTIUS)
```

PRE `NOMS` és un vector que conté paraules que acaben en la lletra **o**. `ADJECTIUS` és un vector que conté paraules que acaben en la lletra **a**.

POST Torna una llista amb totes les combinacions que es pot fer `adjectiu1 adjectiu2 nom`, separats per un espai. En aquest cas, cal fer **totes** les possibles combinacions, sempre i quan `adjectiu1 ≠ adjectiu2`. Per exemple, si `ADJECTIUS = ['bona', 'granda']` i `NOMS = ['viro']`, el resultat hauria de ser la llista

```
R = ['bona granda viro', 'granda bona viro'].
```

Com es pot observar, l'ordre de les combinacions ha de respectar l'ordre que hi ha en els vectors `NOMS` i `ADJECTIUS`.

Recordeu que podeu concatenar `strings` fent servir l'operador de suma. Per exemple, `"bona" + " " + "domo"` avalua a `"bona domo"`.

Observació

Només cal que enviïs el fitxer amb la funció (i les funcions auxiliars que hagi fet) que et demanem i prou. El fitxer `main.py` et pot servir per a fer la teva solució, però no cal que n'enviïs el contingut.

Entrada

paraula1 paraula2 ...
acabada en la paraula FINO

Sortida

Dos vectors que contenen els noms (acabats en *o*) del canal d'entrada i els adjectius (acabats en *a*).

Donats els dos vectors anteriors, totes les parelles *adjectiu nom*.

Donats els dos primers vectors, totes les parelles *adjectiu1 adjectiu2 nom* tals que $adjectiu_1 \neq adjectiu_2$.

Exemple d'entrada 1

afabla
akvo
alta
besto
domo
FINO

Exemple de sortida 1

Noms: (3)
akvo
besto
domo
Adjectius: (2)
afabla
alta
Parelles:
afabla akvo
afabla besto
afabla domo
alta akvo
alta besto
alta domo
Triples:
afabla alta akvo
afabla alta besto
afabla alta domo
alta afabla akvo
alta afabla besto
alta afabla domo

Exemple d'entrada 2

ronda
linio
malpeza
mondo
nova
FINO

Exemple de sortida 2

Noms: (2)
linio
mondo
Adjectius: (3)
ronda
malpeza
nova
Parelles:
ronda linio
ronda mondo
malpeza linio
malpeza mondo
nova linio
nova mondo

Triples:

ronda malpeza linio
ronda malpeza mondo
ronda nova linio
ronda nova mondo
malpeza ronda linio

malpeza ronda mondo
malpeza nova linio
malpeza nova mondo
nova ronda linio
nova ronda mondo
nova malpeza linio
nova malpeza mondo

Informació del problema

Autoria: INFO EPSEVG

Generació: 2026-01-25T18:57:00.051Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>