

Nom de famille :

Prénom(s) :

Numéro Candidat :

Né(e) le : / /

(Le numéro est celui qui figure sur la convocation ou la feuille d'émargement)

CONSIGNES

- Remplir soigneusement, sur CHAQUE feuille officielle, la zone d'identification en MAJUSCULES.
- Ne pas signer la composition et ne pas y apporter de signe distinctif.
- Rédiger avec un stylo à encre foncée (bleue ou noire) ; éviter le stylo plume à encre noire.
- N'effectuer aucun collage ou découpage de sujets ou de feuille officielle. Ne joindre aucun brouillon.



Document réponse de :

PHYS

SVT

NSI

SI

Document réponse de Numérique et Sciences Informatiques

I-1.

① *i*

 ② *score(j, coups[b[0]])*

 ③ *[i]*

 ④ *b.append(i)*

 ⑤ *alea(len(b))*

I-3.

① *k*

 ② *proba(3-j, a[ba])*

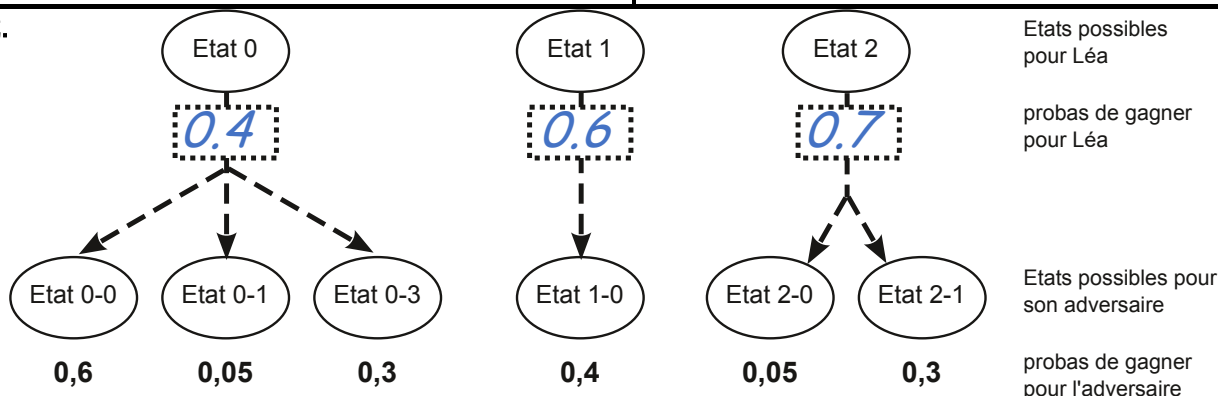
 ③ *ba = k*

 ④ *p*

 ⑤ *i*

 ⑥ *b*

I-2.



I-4.

```
def proba_rec(j, etat, nb):
    a = etats_suiv(3-j, etat)
    if nb <= 1 or len(a) == 0 :
        return proba(j, ①)
    else:
        p = 0.0
        for i in range(0, len(a)) :
            pa = proba_rec(3-j, a[②], ③)
            if pa > p :
                ④
        return ⑤

def emma(j, coups):
    (b, p) = (0, 0.0)
    for i in range(0, len(coups)) :
        pc = proba_rec(j, coups[⑥], 3)
        if pc > p :
            (b, p) = ⑦
    return ⑧
```

① *etat*

 ② *i*

 ③ *nb - 1*

 ④ *p = pa*

 ⑤ *1 - p*

 ⑥ *i*

 ⑦ *(i, pc)*

 ⑧ *b*

NE RIEN ÉCRIRE

DANS CE CADRE

II-1. ['F1', 'G4', 'F2', 'G7', 'F3', 'G2', 'G3', 'G6', 'G1', 'G8', 'G5']

II-2. ['F2', 'F3', 'G5', 'G1', 'F4', 'G7', 'F1', 'G3', 'G2', 'G4', 'G6']

II-3. def fille(etu):
return ① == 'F' ① etu[0]

II-4. def suivante(etus, i):
if i >= len(etus) or fille(etus[i]):
return ① ① i
return suivante(etus, ②) ② i + 1

II-5. def forcer_parite(etus):
etus = etus.copy()
(classt, nbf) = ([], 0)
idxf = suivante(etus, 0)
while etus:
if fille(①): ① etus[0]
classt.append(etus.pop(0))
nbf = nbf + 1
idxf = suivante(etus, ②) ② 0
elif nbf < ③ and idxf < ④ : ③ 0.5 * (len(classt) + 1)
classt.append(etus.pop(⑤)) ④ len(etus)
nbf = nbf + 1
idxf = suivante(etus, ⑥) ⑤ idxf
else:
classt.append(etus.pop(0))
idxf = idxf - 1
return classt ⑥ idxf

II-6. ① affectation[i] ② i, etu

II-7. ① prefs[etu][0] ④ affect[sport]
② etu ⑤ 0
③ classts[sport] ⑥ elimine

II-8. def mariages_stables(prefs, classts, quotas):
prefs = prefs.copy()
affectation = { sport: [] for sport in classts.keys() }
for ① in ② :
affectation_etudiant(③ , ④ , ⑤ , ⑥ , quotas)
return affectation

① xxx ④ affectation
② prefs ⑤ prefs
③ xxx ⑥ classts