GEIPI-POLYTECH v1 ©EXATECH Nom de famille: (Suivi, s'il y a lieu, du nom d'usage)															
Prénom(s) :															
Numéro Candidat : (Le numéro est celui qui figure sur la convocation ou la feuille d'émargement)															
Remplir soigneusement, sur CHAQUE feuille officielle, la zone d'identification en MAJUSCULES.															

CONSIGNES

- Ne pas signer la composition et ne pas y apporter de signe distinctifi
 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.



I-1-	<pre>def creer_file (): return</pre>	I-2- def est_vide(F): return (F) == 0								
I-3-	def enfiler(F, x): F. append (x)	<pre>I-4- def defiler(F): return F.pop()</pre>								
I-5-	1-1;2-1; 3-1; 1-2; 2-2	; 3-2; 1-3; 3-3; 3-4								
I-6-	5- ① x ['temps'] + 1									
	② x ['temps'] < x ['duree']									
	<pre>③ enfiler (F, x)</pre>									
	enfiler (proc, {'id': enfiler (proc, {'id':	ly Wachowski'),								
II-2-	n-5	FROM films e >= 2000 AND annee < 2010 ;								
II-4-	SELECT titre, annee FR	ROM films JOIN realisation								
	WHERE films.idfilm = realisation.idfilm									
	AND nom = 'Peter Pactol';									

```
NE RIEN ÉCRIRE

DANS CE CADRE
```

```
liste_reals
         real
II-5- for
            film
                                  liste films
              real ['idfilm'] == film ['idfilm']
             print(film['idfilm'], film['titre'], film['annee'], real['nom'])
II-6- def contient id(films, idfilm):
       for f in films:
          if f['idfilm'] == idfilm
             return _ True
              False
       return
II-7- ① contient_id_rec (liste, id, pivot + 1, imax)
    ② contient_id_rec (liste, id, imin, pivot - 1)
    3 True
II-8- 3 if not contient_id_rec(films, r['idfilm'], O, len(films) - 1):
      { f ['idfilm'] : f for f in films }
II-9- 4
III-1- L1 = [1, 2, 3, 4]
                           L3 = [2, 1, 4, 3]
                                                L5 = [9, 5, 4, 2, 3, 2]
    L2 = [4, 3, 2, 1]
                           L4 = [3, 4, 1, 2]
                                                  L6 = [8, 3, 4, 1, 2, 3]
                                                         X T.5
    Quelles listes représentent un tas : 🔲 💵
                                    X L2
                                           □ L3
                                                  □ T.4
                                                                 X 1.6
III-2- def parent(i):
                                 III-3-def est feuille(T, i):
                                        return gauche (i) >= len (T)
       return (i-1)//2
III-4- ① T[droite(i)]
    2 droite (i)
    _{3} T[j] > T[i]
    descendre (T,j)
III-5- ⑤ O
                                  6 0
    ② descendre (T, O)
```