```python
# Mines_2017_IPT.py

A = 11*[False]
for k in [0, 2, 3, 10]:
     A[k] = True
print(A)

def _3_occupe(L, i):
    return L[i]

# Q4  2**n

def _5_egal(L1, L2):
    if len(L1) != len(L2):
        return False
    for k in range(len(L1)):
        if L1[k] != L2[k]:
            return False
    return True

# Q6 O(max(len(L1), len(L2))). Q7  Booléen
# Q8  [T, F, T, F, T, T, F, F, F, F, F]

def _9_avancer_fin(L, m):
    L1 = L[:]
    L1[m:m] = [False]
    L1.pop()
    return L1

def _10_avancer_debut(L, b, m):
    L1 = L[:]
    L1[m:m+1] = []
    L1 = [b] + L1
    return L1

def _11_av_debut_bloque(L, b, m):
    L1 = L[:]
    j = m-1
    while j >= 0 and L1[j]:
        j -= 1
    if j >= 0:
        L1[j:j+1] = []
        L1 = [b] + L1
    return L1

def _12_avancer_files(L1, b1, L2, b2):
    m = len(L1) // 2
    R1 = _9_avancer_fin(L1, m)
    R2 = _9_avancer_fin(L2, m)
    R1 = _10_avancer_debut(R1, b1, m)
    if R1[m]:
        R2 = _11_av_debut_bloque(R2, b2, m)
    else:
        R2 = _10_avancer_debut(R2, b2, m)
    return R1, R2

```

```
055| # Q15  9 étapes
056|
057| def _17_elim_double(L):
058|     if L == []:
059|         return []
060|     L1 = L[L[0]]
061|     for i in range(1, len(L)):
062|         if L[i] != L[i-1]:
063|             L1.append(L[i])
064|     return L1
065|
066| # Q21
067| # La première est de complexité linéaire.
068| # La 2e est bien plus efficace :
069| # complexité logarithmique (dichotomie).
070|
071| def _22_versEntier(L):
072|     s = 0
073|     for k in range(len(L)):
074|         s = 2*s + L[k]
075|     return s
076|
077| #  Q23 taille >= 1 + E(log(n, 2))
078| #  while n > 0
079|
080| # Q26
081| # SELECT  id_cr_fin FROM Voie WHERE id_cr_debut = c
082|
083| # Q27
084| # SELECT longitude, latitude FROM Croisement JOIN Voie ON
085| # Croisement.id = Voie.id_cr_fin WHERE Voie.id_cr_debut = c
086|
087| # Q28
088| #  Atteignables en 2 étapes à partir de c
089|
090|
091| '''
092| L = [True, False, True, True]
093| print(_22_versEntier(L))
094|
095| L1 = [True, False, True, True, False, True, True, False, False,
False, False]
096| L2 = [True, False, True, True, True, False, False, False, False,
False, False]
097|
098| for k in range(5):
099|     L1, L2 = _12_avancer_files(L1, False, L2, False)
100| print(L1)
101| print(L2)
102| '''
```