

```

'''
Chp1 : TP N°2 Les Piles
Groupe : Ipein/SP3/GB
Date : 23-09-2023
'''
# Pile
def creer_pile():
    p = [] # p = list()
    return p # return []

def pile_vide(p):
    return len(p) == 0 # return p == []

def sommet(p):
    if not pile_vide(p):
        return p[-1] # p[len(p)-1]
    #return None

#utilisation
p = creer_pile()
s = sommet(p)
if s != None:
    s = s + 5

def taille(p):
    return len(p)

def empiler(p,x):
    p.append(x)

#utilisation
p = creer_pile()
empiler(p,2)

def depiler(p):
    return p.pop()

#utilisation
p = creer_pile()
try:
    depiler(p)
except IndexError:
    print("pile vide")

# EX1
def conversion(n):
    assert type(n)==int
    p = creer_pile()

    if n==0 : return '0b0'

    while n != 0:
        r , n = n % 2, n // 2
        empiler(p,r)

    ch = '0b'
    while taille(p)>0:
        ch = ch + str(depiler(p))

    return ch

# Ex 2

def verif_parentheses(ch):
    p = creer_pile()
    L = []
    for i in range(len(ch)):
        if ch[i] == '(' : empiler(p,i)
        elif ch[i] == ')':
            if pile_vide(p): return False
            L += [(depiler(p),i)]

    if pile_vide(p) : return L
    return False

def calc_expr_arith(expr):
    """expr : postfixée """
    p = creer_pile()
    for c in expr:
        if c not in "+-*/":
            empiler(p,int(c))

```

```

else:
    b,a = depiler(p),depiler(p)
    if c=='+':    r = a+b
    elif c=='-':  r = a-b
    elif c=='*':  r = a*b
    elif c=='/':
        if b==0:
            raise Exception("impo")
        else:
            r = a/b
    empiler(p,r)
    #empiler(p,eval(str(a)+c+str(b)))
resultat = depiler(p)
return resultat # return sommet(p)

# ex3
def permu_circ(p,n):
    p1 = creer_pile()
    for i in range(n):
        s = depiler(p)
        while not pile_vide(p):
            empiler(p1, depiler(p))
        empiler(p,s)
        while not pile_vide(p1):
            empiler(p,depiler(p1))

```