```python
'''
Chp1 : TP N°2 Les Piles
Groupe : Ipein/SP2/GB
Date : 23-09-2023
'''
# Pile
def creer_pile():
   l = list()
   return l
   # return []

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

def taille(p):
   return len(p)

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

#test
'''
p = creer_pile()
s = sommet(p)
if s != None:
   print("sommet:",s+1)
'''
def sommet1(p):
   assert not pile_vide(p)
   return p[-1]

#test
'''p = creer_pile()
try:
   s = sommet1(p)
except :
   print("pas de sommet")
else:
   print("sommet : ",s)'''

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

#test
'''p = creer_pile()
empiler(p,3)'''

def depiler(p):
   if not pile_vide(p):
       x = p.pop()
       return x
   else:
       raise Exception("Pile vide")

#test
'''p = creer_pile()
try:
   s = depiler(p)
except:
   print("pile vide, pas sommet")'''

def depiler1(p):
   return p.pop()

# Ex 1

def conversion(n):
   p = creer_pile()
   if n==0: return '0b0'
   while n!=0 :
       r = n % 2
       empiler(p,r)
       q = n // 2
       n = q

   ch ='0b'
   for i in  range(taille(p)):
       ch = ch + str(depiler(p))

   return ch
```

```python
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
           t = (depiler(p),i)
           L.append(t)
   if pile_vide(p): return L
   return False

def calc_expr_arith(expr):
   # expr : post fixée
   p = creer_pile()
   for e in expr:
       if e.isdigit():
           empiler(p,int(e))
       else:
           b = depiler(p)
           a = depiler(p)
           #e : opération de type str
           if e == '+': r = a + b
           elif e=='-': r = a - b
           elif e=='*': r = a * b
           elif e=='/': r = a / b
           empiler(p,r)
   return sommet(p)


# Ex 3
def permut_circ(p,n):
   p1 = creer_pile()
   for i in range(n):
       s = depiler(p)
       #vider p dans p1
       while taille(p)>0:
           empiler(p1,depiler(p))
       # mettre s au fond de p
       empiler(p,s)

       #remettre les elts de p
       while taille(p1)>0:
           empiler(p,depiler(p1))

# Ex 4
def somme(p):
   if taille(p)==0 :
       return 0
   else:
       s= depiler(p)
       if type(s)==int:
           return s + somme(p)
       else:
           return somme(s) + somme(p)
```