```
##PIVOT DE GAUSS résoudre (AX=B)
#les 2 fonctions éssentielles sont
def echelonnement(A,B): #A devient triangulaire sup
    A1=A.copy()
    B1=B.copy()
    for i in range(len(A1)): #une seule boucle car on travaille sur
la diagonale !
        k=indice_pivot(A1,i) #on cherche le pivot sur la colonne
        A1=permutation(A1,i,k) #on permute les lignes
        B1=permutation(B1,i,k) #ne pas oublier de faire le même
travail sur A et B
        a=1/A1[i][i]
        A1=dilatation(A1,i,a) #1 sur la diagonale
        B1=dilatation(B1,i,a)
        for j in range(i+1,len(A1)):  #0 en dessous de la diagonale
            a=-A1[j][i]
            A1=transvection(A1,j,i,a)
            B1=transvection(B1,j,i,a)
    return A1,B1

#et
def reduction(Ae,Be): # A devient diagonale de 1
    A1=Ae.copy()
    B1=Be.copy()
    for i in range(len(A1)-1,0,-1): #il faut partir de la fin de la
matrice pour faire une transvection
        for j in range(i):
            a=-A1[j][i]
            A1=transvection(A1,j,i,a)
            B1=transvection(B1,j,i,a)
    return A1,B1

def Gauss_Jordan(A,B):
    Ae,Be=echelonnement(A,B)
    Ar,Br=reduction(Ae,Be)
    return Br

##########################################################
##je vous remets les fonctions des base
def permutation(M,i,j):
    N=M.copy()
    N[i],N[j]=M[j],M[i]
    return N

def dilatation(M,i,a):
    N=M.copy()
    N[i]=a*N[i]
    return N

def transvection(M,i,j,a):
    N=M.copy()
    N[i]=N[i]+a*M[j]
    return N

def indice_pivot(M,i):
```

```
#on cherche l'indice du pivot non nul dans une colonne mais pour
améliorer la précision de l'algo on cherche la valeur max (absolue)
    k=i
    for j in range(i+1,len(M)):
        if abs(M[j][i])>abs(M[k][i]):
            k=j
    return k

## vous pouvez tester
from numpy import *
A=array( [[-2.,4.,1.],[8.,2.,-1.],[2.,-1.,2.]] )
B=array( [[-18.],[6.],[27.]] )
print("1er test AX=B\n","avec A :\n",A,"\n B \n : ",B)
print("la solution est :")
print(Gauss_Jordan(A,B))


A=array([[0,2.,1.,0],[0,2.,1.,-3],[2.,0,1.,2.],[1.,-3.,0,-2]])
B=array( [[-1.],[2.],[3.],[1.]] )
print("\n 2eme test AX=B\n","avec A :\n",A,"\n B : \n",B)
print("la solution est :")
print(Gauss_Jordan(A,B))
```