```python
def test_perm_mean(x,y,side="upper",B=100) :
    # test par permutation pour comparer les moyennes de 2 groupes
    # entrees
    #   x : variable observée (ex: age)
    #   y : variable de groupe (ex: classification)
    #   side : sens du test,
    #          side="upper"  si H1 : mu2>mu1 (defaut)
    #          side="lower"   si H1 : mu2<mu1
    #          side="both"  si H1 : mu2!=mu1
    #   B : nombre de permutations
    # sorties
    #   zobs : statistique de test observée
    #   pv : p-value
    #   z    : echantillon des statistiques de test sous H0

    n_first_class = np.size(np.where(y==1))              # Détermine le nombre d'éléments dans la première classe
    m1 = np.mean(x[0:n_first_class])                     # Calcule la moyenne des données de la première classe
    m2 = np.mean(x[n_first_class:])                      # Calcule la moyenne des données de la seconde classe
    zobs = m2-m1                                         # Statistique observée
    diff_sim = np.zeros(B)
    for b in range(B):
        xb = permutation(x)                                  # Mélange les données
        m1 = np.mean(xb[0:n_first_class])                    # Calcule la moyenne des données se retrouvant dans le premier paquet
        m2 = np.mean(xb[n_first_class:])                     # Calcule la moyenne des données se retrouvant dans le second paquet
        diff_sim[b] = m2-m1 # dans quel sens fait-on le test?  # Statistique observée
        if (side=="upper"):
            pvalue = np.sum(diff_sim>zobs)/B
        elif (side=="lower"):
            pvalue = np.sum(diff_sim<zobs)/B
        else:
            pvalue = min(np.sum(diff_sim<zobs)/B,np.sum(diff_sim>zobs)/B)*2
    return([zobs,pvalue,diff_sim])
```