

```

# ===== #
# TRI PAR INSERTION #
# Evaluation expérimentale de la complexité moyenne. #
# (nombre de comparaisons et nombre d'inversions effectuées). #
# Version du 21/11/2015 #
# ===== #

# Importations
# =====
from random import randint
from time import perf_counter

# La fonction de tri
# =====
def tri_insert(T):
    global c
    lT = len(T)
    for i in range(1,lT):
        x = T[i]
        k = i
        while k>0 and x<T[k-1]:
            c += 1
            T[k] = T[k-1]
            k -= 1
        T[k] = x

# Saisie des paramètres et initialisations
# =====
c = 0
ntris = int(input('Combien de tris souhaitez-vous ? '))
n = int(input('\nLongueur des listes à trier ? '))
valmax = int(input('\nValeur maxi des nombres générés ? '))
complex = 0

for i in range(ntris):
    TNA = [randint(0,valmax) for i in range(n)]
    t_start = perf_counter()
    tri_insert(TNA)
    t_end = perf_counter()
    complex += (t_end - t_start)

# Affichage de la valeur moyenne de "temps de tri/n²" qui doit être à peu près
# "constant" et de la valeur moyenne de "nombre de comparaisons/n² qui doit être
# proche de 1/4.
# =====
print('\nDurée du tri moyenne : ',complex/(ntris*n**2))
print('\nNombre moyen de comparaisons : ',c/(ntris*n**2))

# =====
# FIN DU PROGRAMME

```