

Nom et Prénom : \_\_\_\_\_ Groupe : \_\_\_\_\_

Nom et Prénom : \_\_\_\_\_ Groupe : \_\_\_\_\_

**1ère année à l'ESSAI**  
**Test de Travaux Pratique**  
**25/11/21011 et 02/12/2011**

Nous désirons programmer numériquement la décomposition LU d'une matrice carré inversible :

**Entrées :**  $A$  une matrice carré d'ordre  $n$  tel qu'il existe  $L$  une matrice triangulaire inférieure avec des 1 sur la diagonale et  $U$  une matrice triangulaire supérieure inversible vérifiant  $A = LU$ .

**Sortie :** La matrice  $L$  et la matrice  $U$ .

**Question 1.** Donner les conditions nécessaires et suffisantes qui assurent l'existence de la factorisation  $A = LU$ .

**Question 2.** Expliquer la commande des lignes 1, 2 et 3 du programme suivant. Corriger, compléter puis implémenter ce programme sous `sage`.

```
def factLU(n):  
  
1.   A = random_matrix(QQ,n,n)  
  
2.   L = matrix(QQ,n,n,1)  
  
   U = copy(A)  
  
   if det(U)==0: return(".....")  
  
   else for k in range(n-1):  
  
       if U[k,k]==0: return('.....')  
  
       else: for i in range(k+1,n):  
  
           L[i,k]=U[i,k]/U[k,k]  
  
           for i in range(k+1,n): for j in range(k+1,n):  
  
               U[i,j]=U[i,j]-(U[i,k].U[k,j])/U[k,k]  
  
           for j in range(k+2,n): U[j,k]=0  
  
3.   return(A,L,U,L*U,L*U==A)  
  
@interact def (n=(3,(2..5))) show(factLU(n))
```

**Question 3.** Voir le Tableau blanc.