

Algorithmique et Programmation – Examen (durée 1h30)

Deuxième session du 24 juin 2025

NOTES : Aucun document autorisé. Sont interdits les calculatrices, les téléphones, ainsi que tout autre ustensile de calcul ou de communication.

Exercice 1 : Questions de connaissances générales (5 points)

5 questions

Exercice 2 : Deux fonctions et un programme à compléter (6 points)

Nous souhaitons opérer des rotations autour de l'axe z pour un ensemble de points d'un espace tridimensionnel. Cette rotation, d'un angle theta, est possible via la multiplication de vecteurs tridimensionnels par la matrice suivante :

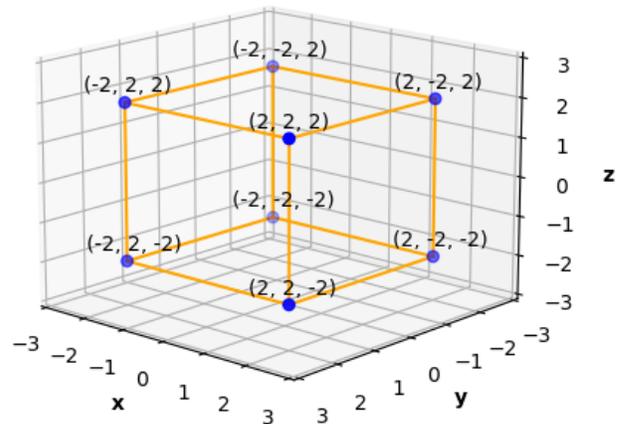
$$\begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

En l'occurrence, les points (x, y, z) qui nous intéressent sont les huit sommets d'un cube, donnés par chaque ligne du fichier ci-dessous et représentés graphiquement sur l'image suivante :

```

coordonnees.txt
-2 -2 -2
-2 -2 2
-2 2 -2
-2 2 2
2 -2 -2
2 -2 2
2 2 -2
2 2 2

```

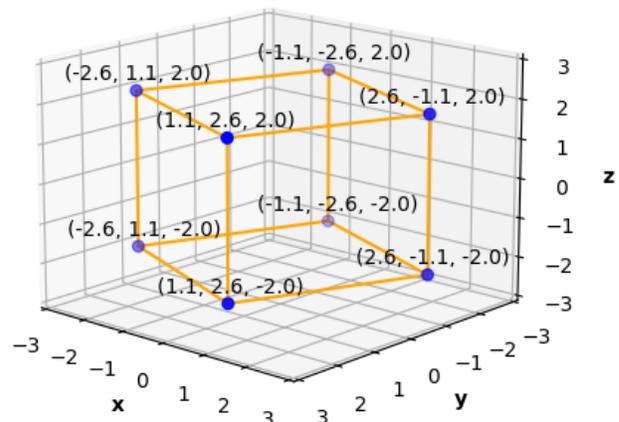


L'objectif du programme est ainsi de lire des coordonnées (x, y, z) de points par lecture d'un fichier et d'opérer la rotation selon la valeur d'angle donnée par l'utilisateur du programme, puis d'afficher les nouvelles coordonnées des points après rotation autour de l'axe z. Une illustration de l'exécution de ce programme et une représentation graphique des points obtenus sont disponibles ci-dessous :

```

C:\Invite de commandes
C:\ExemplesC> exercice2.exe
Donnez theta (rad) : 0.39
Points apres rotation :
(-1.1, -2.6, -2.0)
(-1.1, -2.6, 2.0)
(-2.6, 1.1, -2.0)
(-2.6, 1.1, 2.0)
( 2.6, -1.1, -2.0)
( 2.6, -1.1, 2.0)
( 1.1, 2.6, -2.0)
( 1.1, 2.6, 2.0)
C:\ExemplesC>

```



Remarquons que notre exemple illustre huit points, mais que ce programme doit pouvoir appliquer des rotations peut importe le nombre de points dans le fichier.

Pour écrire ce programme, nous allons nous munir de deux fonctions :

1. La fonction `rotation_z(theta, matrice)` permettant d'initialiser une matrice de rotation autour de l'axe z selon l'angle theta.
2. La fonction `multiplier(vecteur, matrice)` permettant de multiplier un vecteur et une matrice.

Les trois parties du code du programme ci-après sont à compléter séparément.

1) [2 points]

```

----- <----->
----- <----->

---- rotation_z (double theta, double matrice[3][3]) _
  matrice_____ = cos(theta) _
  matrice_____ = -sin(theta) _
  matrice_____ = 0 _
  matrice_____ = sin(theta) _
  matrice_____ = cos(theta) _
  matrice_____ = 0 _
  matrice_____ = 0 _
  matrice_____ = 0 _
  matrice_____ = 1 _
-

```

2) [2 points]

```

---- multiplier (double vecteur[3], double matrice[3][3]) _
  double x = _, y = _, z = _ _
  ___ i _
  ---- (i = _ _ i _ _ _ i++) _
    x += vecteur___ * matrice_____ _
    y += vecteur___ * matrice_____ _
    z += vecteur___ * matrice_____ _
-
  vecteur[0] = _ _
  vecteur[1] = _ _
  vecteur[2] = _ _
-

```

3) [2 points]

```

---- () _
FILE *f _
double theta, x, y, z, point[_], matrice_z[_][_] _

_____("Donnez theta (rad)~: ") _
_____("%lf", _____) _

_____(______, matrice_z) _

_ = fopen("_____", "_") _

_____("Points apres rotation~: __") _
while (!feof(_)) _
  __ (fscanf(_, "%lf %lf %lf", __, __, __) == _) _
  point[_] = x _
  point[_] = y _
  point[_] = z _
  _____(______, matrice_z) _
  _____("(%4.1f, %4.1f, %4.1f )\n", point[0], point[1], point[2]) _
  -
-

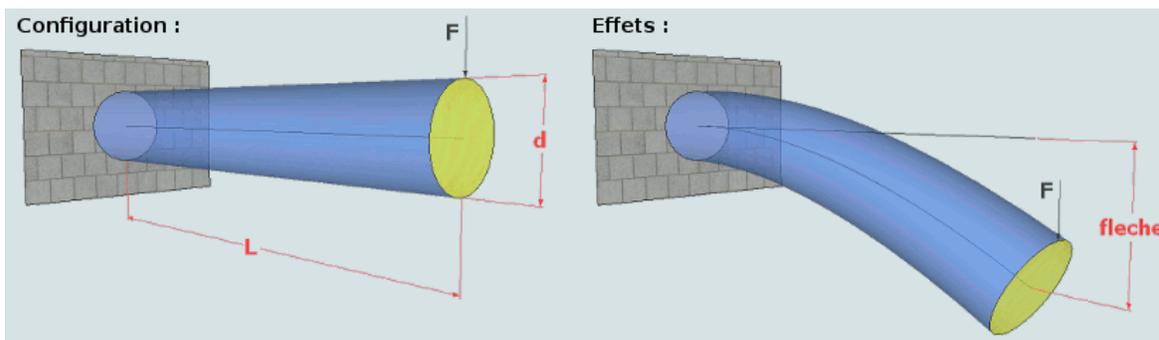
fclose(_) _

_____ 0 _
-

```

Exercice 3 : Trois fonctions et deux programmes à écrire (9 points)

Nous souhaitons appliquer des calculs de résistance des matériaux pour déterminer le diamètre à utiliser pour des poutres circulaires en acier encastées à une extrémité :



Les calculs porteront sur la flèche à l'extrémité (en mm) et la contrainte maximum au point d'encastement (en MPa), lors de la flexion d'une poutre circulaire encastée par application d'une force notée F (en N) à son extrémité. Les paramètres des calculs seront les suivants :

- Le matériau est toujours un acier approvisionné avec ces caractéristiques :
 1. Un module d'Young noté E de 210 GPa.
 2. Une limite élastique en traction notée Re de 250 MPa.

- La longueur de la poutre en mm (notée L)
- Le diamètre de la poutre en mm (noté D)
- Les formules pour calculer la flèche à l'extrémité de la poutre et la contrainte maximum à l'encastrement de la poutre seront les suivantes :

$$IGz = \frac{\pi \cdot D^4}{64}$$

$$IGz = \frac{\pi(D^4 - d^4)}{64}$$

$$\text{flèche à l'extrémité} = \frac{F \cdot L^3}{3 \cdot E \cdot 10^3 \cdot IGz}$$

$$\text{contrainte maximum à l'encastrement} = \frac{L \cdot F \cdot D}{2 \cdot IGz}$$

La limite élastique en cisaillement du matériau (notée Te) se calcule en fonction de la limite élastique en traction (notée Re) comme suit :

Pour les matériaux fragiles : $Te = Re/2$ Pour les matériaux ductiles : $Te = Re/\sqrt{3} \approx Re/1,732$

Dans la suite nous allons choisir $Re/2$ comme limite, car plus restrictive.

Remarque : La valeur de π pourra être donnée par la constante M.PI.

a) [3 points] Écrivez trois fonctions permettant respectivement de calculer le moment quadratique (IGz), la flèche à l'extrémité et la contrainte maximum à l'encastrement, ayant les prototypes suivants :

- double moment (double D);
- double fleche (double E, double L, double D, double F);
- double contrainte_maximum (double L, double D, double F);

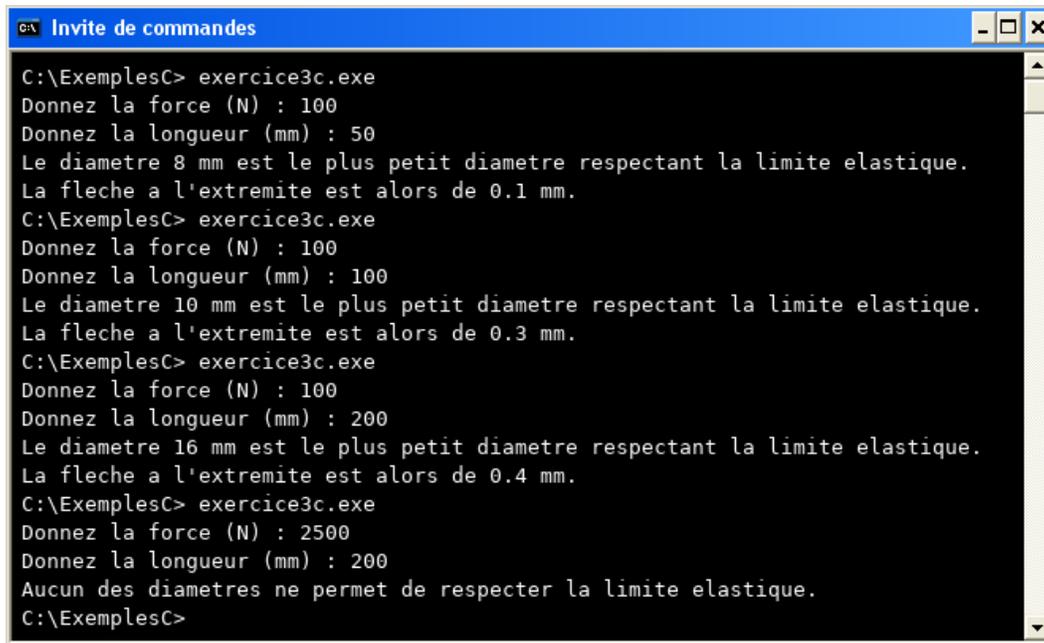
b) [3 points] Pour tester les fonctions du (a), écrivez un programme qui calcule la flèche à l'extrémité, pour une poutre circulaire encastree de 100 mm de longueur et 10 mm de diamètre, lorsque la force donnée par l'utilisateur du programme s'applique à son extrémité, puis indique si la limite élastique est respectée ou non. Deux exemples d'exécution de ce programme sont illustrés ci-dessous :

```

C:\ExemplesC> exercice3b.exe
Donnez la force (N) : 100
La fleche a l'extremite est de 0.3 mm.
La contrainte maximum appliquee a la poutre est de 101.9 MPa.
Respecte la limite elastique.
C:\ExemplesC> exercice3b.exe
Donnez la force (N) : 200
La fleche a l'extremite est de 0.6 mm.
La contrainte maximum appliquee a la poutre est de 203.7 MPa.
Depasse la limite elastique.
C:\ExemplesC>

```

c) [3 points] Maintenant, écrivez un autre programme qui permet de trouver le plus petit diamètre de poutre à utiliser, parmi un choix de six diamètres 4, 6, 8, 10, 16 et 32 mm, pour que la contrainte élastique soit respectée pour une poutre circulaire encastrée d'une longueur donnée par l'utilisateur du programme, lorsqu'une certaine force, également donnée par l'utilisateur, devra pouvoir s'appliquer à son extrémité sans dommages. Le programme indiquera également la valeur de la flèche obtenue lorsque la limite élastique est respectée. Quatre exemples d'exécution de ce programme sont illustrés ci-dessous :



```
C:\ExemplesC> exercice3c.exe
Donnez la force (N) : 100
Donnez la longueur (mm) : 50
Le diametre 8 mm est le plus petit diametre respectant la limite elastique.
La fleche a l'extremite est alors de 0.1 mm.
C:\ExemplesC> exercice3c.exe
Donnez la force (N) : 100
Donnez la longueur (mm) : 100
Le diametre 10 mm est le plus petit diametre respectant la limite elastique.
La fleche a l'extremite est alors de 0.3 mm.
C:\ExemplesC> exercice3c.exe
Donnez la force (N) : 100
Donnez la longueur (mm) : 200
Le diametre 16 mm est le plus petit diametre respectant la limite elastique.
La fleche a l'extremite est alors de 0.4 mm.
C:\ExemplesC> exercice3c.exe
Donnez la force (N) : 2500
Donnez la longueur (mm) : 200
Aucun des diametres ne permet de respecter la limite elastique.
C:\ExemplesC>
```