

TP 1 & 2 : Structures de Sélection en Langage C et Python

Consignes

- Le TP est à faire de manière individuelle.
- Il est demandé de rendre une archive `nom_prenom.zip` qui contient les fichier `.c` et `.py`.
- L'archive doit être rendue sur Moodle à la fin des 4h de TP.
- Il est conseillé d'utiliser un éditeur directement sur le PC (e.g. Visual Studio Code).

Exercice 1 : Calculatrice

Nous souhaitons réaliser une mini-calculatrice permettant de réaliser 5 opérations au choix entre deux nombres réels :

- 1- une addition $a + b$
- 2- une soustraction $a - b$
- 3- une multiplication $a \times b$
- 4- une division a/b (nous vérifierons que $b \neq 0$)
- 5- une racine carrée \sqrt{a} (nous vérifierons que $a \geq 0$).

La fonction implémentée permet de choisir et de réaliser l'opération sélectionnée.

- 1) Implémenter une solution en C dans le fichier `calculatrice.c`.
- 2) Implémenter une solution en Python dans le fichier `calculatrice.py`.

Exercice 2 : Affranchissement de Lettres

En 2021, pour l'envoi de lettres en France, il existe trois types de timbres : le timbre **rouge** (prioritaire (Délai d'acheminement indicatif : J+1)), le timbre **vert** (écologique (Délai d'acheminement indicatif : J+2)) et le timbre **gris** (écopli (Délai d'acheminement indicatif : J+4)).



Jusqu'à **20** g, la lettre devra être affranchie avec **un** seul timbre.

Entre **20** et **100** g, la lettre devra être affranchie avec **deux** timbres.

Entre **100** et **250** g, la lettre devra être affranchie avec **quatre** timbres.

Entre **250** et **500** g, la lettre devra être affranchie avec **six** timbres.

Entre **500** et **3** kg, la lettre devra être affranchie avec **huit** timbres.

Il n'est pas possible d'envoyer une lettre de plus de 3 kg, il faut alors passer par l'envoi d'un colis. Attention en tarif économique, la limite de poids d'envoi d'une lettre est de 250 g.

Les programmes C et Python permettent de déterminer le tarif d'envoi d'une lettre en fonction de son poids et de son temps de distribution (en jours).

- 1) Implémenter une solution en C dans le fichier `affranchissement.c`.
- 2) Implémenter une solution en Python dans le fichier `affranchissement.py`.

Exercice 3 : Recherche des Solutions d'une Équation

Soit l'équation définie par :

$$Ax^2 + Bx + C = 0 \quad \text{où } A, B \text{ et } C \text{ sont des nombres réels quelconques.}$$

L'implémentation doit trouver, si elles existent, les solutions réelles de cette équation.

- 1) Implémenter une solution en C dans le fichier `equation.c`.
- 2) Implémenter une solution en Python dans le fichier `equation.py`.

Exercice 4 : Quel est le Jour de la Semaine ?

La méthode de calcul du numéro du jour d'une date comprise entre les années **1600** et **2199** est la suivante :

1. Nous prenons comme nombre initial les deux derniers chiffres de l'année ;
2. Nous lui ajoutons **un quart** du nombre composé des deux derniers chiffres de l'année, obtenu en faisant une division entière (Nous ignorons les restes.) ;
3. Nous lui ajoutons le numéro du mois qui est obtenue à l'aide du tableau ci-dessous :

Janvier	Février	Mars	Avril	Mai	Juin
1	4	4	0	2	5

Juillet	Août	Septembre	Octobre	Novembre	Décembre
0	3	6	1	4	6

4. Nous soustrayons 1 à ce numéro de jour **si l'année est bissextile et si le mois est janvier ou février**

Nous rappelons qu'une année bissextile est une année divisible par 4, mais pas par 100, ou bien divisible par 400.

5. Nous lui ajoutons selon le siècle de la date une valeur indiquée dans le tableau ci-dessous :

Années 1600	Années 1700	Années 1800	Années 1900	Années 2000	Années 2100
6	4	2	0	6	4

6. Nous ajoutons ensuite le reste de la division par 7 du jour.

7. Nous divisons cette somme par 7 et nous gardons le reste.
Ce reste correspond au jour de la semaine recherché.

Dimanche	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi
1	2	3	4	5	6	0

Les programmes Python et C permettent de savoir quel est le jour de la semaine d'une date comprise entre les années 1600 et 2199. Cette date est saisie au clavier.

- 1) Implémenter une solution en C dans le fichier `jour.c`.
- 2) Implémenter une solution en Python dans le fichier `jour.py`.

Exercice 5 : Location de Voiture

Une société de location de véhicule propose trois types de véhicules : *utilitaire*, *monospace*, et *citadine*.

Pour chaque location, le tarif de location comprend une partie **fixe**, plus une partie fonction de **la durée** de location, et enfin une partie dépendant du nombre de **kilomètres parcourus**.

Le tableau ci-dessous récapitule les informations de tarification :

Type de véhicule	Montant fixe	Prix par jour	Prix par kilomètre
Utilitaire	50 €	80 €	0,30 €
Monospace	25 €	75 €	0,25 €
Citadine	20 €	60 €	0,20 €

Par ailleurs, pour les véhicules de type *citadine* et *monospace*, les clients ont deux options possibles :

- . un **tarif de base** correspondant aux données ci-dessus ;
- . un **tarif forfaitaire** pour lequel le prix par jour est divisé par 2 et le prix par kilomètre est multiplié par 1,5.

Enfin, à partir de **4 jours** de location ou de **2000 kilomètres** parcourus, nous appliquons une remise de **5%** pour tout type de véhicule.

La fonction implémentée permet la saisie des informations nécessaires (Type de véhicule, durée de location, ...) et affiche le prix à payer.

- 1) Implémenter une solution en C dans le fichier `location.c`.
- 2) Implémenter une solution en Python dans le fichier `location.py`.