

Algorithmique et programmation 1

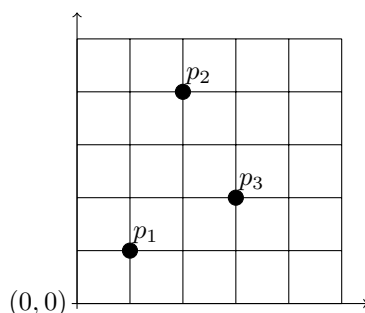
Feuille d'exercices 5 - Enregistrements

Exercice 1 _____ Dates

1. Définir un type `date_t` permettant de donner une date sous le format jour/mois/année.
2. Déclarer une variable `d` de type `date_t` et l'initialiser à la date d'aujourd'hui
3. Ecrire une fonction `comparedate` qui compare deux dates et renvoie -1 , 1 ou 0 suivant si la première date est avant, après, ou égale à la deuxième date

Exercice 2 _____ Points

1. Définir un type `point_t` permettant de représenter un point dans le plan.
2. Déclarer trois variables `p1`, `p2` et `p3` de type `point_t` correspondant aux trois points ci-dessous :



3. Ecrire une fonction `distance` qui prend en paramètres deux points (de type `point_t`), et renvoie la distance entre les deux points. On rappelle que la distance dans le plan entre le point (x_1, y_1) et le point (x_2, y_2) est donnée par la formule

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

On définit maintenant un type `triangle_t` représentant un triangle à partir de ses trois sommets :

Types

```
triangle_t : enregistrement
  s1, s2, s3 : point_t
```

4. Définir une variable `tr` correspondant au triangle dont les trois sommets sont les trois points p_1, p_2, p_3 .
5. Ecrire une fonction `rectangle` qui prend en paramètre un triangle et renvoie Vrai si le triangle est un triangle rectangle, et faux sinon (utiliser par exemple le théorème de Pythagore)
6. Ecrire une fonction `surface` qui prend en paramètre un triangle et renvoie la surface du triangle. Si le triangle est défini par les trois points (x_1, y_1) , (x_2, y_2) et (x_3, y_3) , la surface est donnée par la formule :

$$\frac{|x_1 y_2 - x_2 y_1 + x_2 y_3 - x_3 y_2 + x_3 y_1 - x_1 y_3|}{2}$$

7. Définir un type `polygone_t` permettant de représenter un polygone à moins de 10 sommets. Ecrire une variable `tp` correspondant au triangle dont les trois sommets sont les trois points p_1, p_2, p_3 .

8. Ecrire un fragment d'algorithme qui demande à l'utilisateur un réel r et qui crée une variable `carré` de type `polygone_t` représentant un carré de côté r centré en 0.
9. Ecrire une fonction `coteseaux` qui prend un polygone en entrée et qui renvoie Vrai si tous les côtés du polygone sont la même longueur, et Faux sinon.
10. Ecrire une fonction `surface_poly` qui prend un polygone en entrée et qui renvoie la surface du polygone. Si le polygone est défini par les points $(x_1, y_1) \dots (x_n, y_n)$, la surface du polygone est :

$$\frac{|x_1y_2 - x_2y_1 + x_2y_3 - x_3y_2 + \dots + x_ny_1 - x_1y_n|}{2}$$

Exercice 3 Fractions

Dans cet exercice on cherche à manipuler des nombres rationnels représentés par des fractions. Une fraction est la donnée d'un numérateur et d'un dénominateur, ce qui conduit naturellement à la définition de type suivante :

Types

```
fraction_t : enregistrement
    num : entier
    den : entier
```

La fonction suivante permet ainsi de convertir un entier en une fraction :

Fonction `entier2fraction(n : entier) : fraction_t`

Variables

```
unefraction : fraction_t
```

Début

```
unefraction.num ← n
unefraction.den ← 1
retourner unefraction
```

Fin

Finfonction

L'entier 5 est donc représenté par la fraction $\frac{5}{1}$.

1. Le dénominateur d'une fraction ne peut être nul. Ecrire une fonction `valide(f: fraction_t): booléen` qui teste si la fraction `f` est valide, donc si son dénominateur est non nul.

Dans toute la suite de l'exercice, on pourra supposer que les fractions fournies en argument des fonctions sont valides, i.e. ont un dénominateur non nul.

2. Une fraction est sous forme standard si son dénominateur est strictement positif. Ecrire une fonction `standard(f: fraction_t): fraction_t` qui convertit la fraction `f` en une fraction sous forme standard représentant le même rationnel. Par exemple si $f = \frac{-4}{5}$ la fonction doit renvoyer $\frac{4}{-5}$.
3. Ecrire une fonction `fraction2reel(f: fraction_t): réel` qui convertit la fraction `f` en un réel. Ainsi, si la fonction est appelée avec une fraction dont le numérateur vaut 2664 et le dénominateur 400, la fonction doit retourner $2664/400 = 6.66$
4. Ecrire une fonction `produit(f: fraction_t, g: fraction_t): fraction_t` qui renvoie le produit des deux fractions `f` et `g`.
5. Ecrire une fonction `somme(f: fraction_t, g: fraction_t): fraction_t` qui renvoie la somme des deux fractions `f` et `g`. On effectuera la somme de la façon la plus simple possible, en multipliant le numérateur de la première fraction par le dénominateur de la deuxième, et réciproquement, sans simplifier. Ainsi

$$\frac{3}{4} + \frac{5}{6} = \frac{3 \times 6}{4 \times 6} + \frac{5 \times 4}{4 \times 6} = \frac{3 \times 6 + 5 \times 4}{4 \times 6} = \frac{38}{24}$$

6. Ecrire une fonction `egal(f: fraction_t, g: fraction_t): booléen` qui teste si les deux fractions `f` et `g` représentent le même nombre rationnel.
7. Ecrire une fonction `pluspetit(f: fraction_t, g: fraction_t): booléen` qui teste si le nombre rationnel représenté par la fraction `f` est plus petit que celui représenté par la fraction `g`. L'utilisation de la fonction `standard` pourrait s'avérer opportune.

Exercice 4 Zine

Campus-Zine est une revue hebdomadaire consacrée à la vie du Campus de la Faculté des Sciences et Technologies de l'Université de Lorraine, et consiste en une feuille recto simple au format A4 (21 x 29.7 cm).

Chaque semaine, la rédactrice en chef du journal Campus-Zine reçoit, de chacun de ses journalistes et chroniqueurs, un article. Le journaliste lui donne le format de l'article (par exemple : 10 cm sur 5 cm) et l'emplacement exact où il/elle souhaite l'afficher sur la feuille A4 (par exemple : aux coordonnées (2cm, 5cm)).

La rédactrice en chef doit vérifier que les articles ne se chevauchent pas, et doit chercher combien de place reste disponible sur la feuille de papier une fois tous les articles imprimés.

La rédactrice n'aime pas avoir à effectuer cette tâche subalterne et préférerait qu'elle soit automatisée dans un logiciel, afin de pouvoir consacrer son temps à des choses plus importantes. Elle décide donc de confier la réalisation d'un tel logiciel à ses ami(e)s étudiant(e)s de L1.

1. Définir un type de données `article_t`.
2. Ecrire une fonction qui permet de tester si un article rentre dans une feuille A4 (c'est à dire s'il ne dépasse pas d'une des marges)
3. Ecrire un fragment d'algorithme qui permet à la rédactrice de saisir un ensemble d'au maximum 100 articles, et les dispose dans un tableau préalablement déclaré.
4. On suppose avoir accès à une fonction `chevauche(x, y: article_t): booléen` qui dit si les articles `x` et `y` se chevauchent. Continuer l'écriture de l'algorithme de la fonction précédente pour qu'il teste que tous les articles ne se chevauchent pas.
5. Ecrire une fonction qui calcule la surface occupée par un article donnée. Continuer l'écriture de l'algorithme de la question précédente pour qu'il calcule la surface totale occupée par tous les articles (en supposant qu'ils ne se chevauchent pas), et qu'il affiche à la rédactrice la place restante.
6. Ecrire la fonction `chevauche`.

Exercice 5 BluRay

On s'intéresse dans cet exercice à un logiciel très rudimentaire de gestion d'un modeste vendeur de BluRay.

Le vendeur a un catalogue d'un peu moins de 2000 BluRay, mais ils ne sont pas tous en stock.

Le logiciel permet d'effectuer les opérations suivantes :

1. Rentrer un nouveau BluRay dans le catalogue, en rentrant son nom, son prix, et combien d'exemplaires le vendeur a acheté
2. Afficher la liste de tous les BluRay
3. Afficher la liste des BluRay en stock (en affichant combien d'exemplaires il y a en stock)
4. Effectuer la vente d'un BluRay
5. Changer le nombre d'exemplaires disponibles d'un BluRay
6. Afficher le chiffre d'affaire (la somme totale d'argent résultant des ventes de BluRay)
7. Trouver le BluRay le plus vendu
8. Trouver le BluRay le plus vendu qui n'est pas en stock actuellement
9. Supprimer un BluRay de son catalogue

Travail à effectuer :

- a) Lire bien attentivement les opérations à effectuer. En déduire la définition d'un type `BluRay_t`, puis la définition d'un type `CollectiondeBluRay_t`.
- b) Ecrire le programme.

Exercice 6 _____ S'il reste du temps

1. (S'il reste du temps) Ecrire une fonction `simplifie(f: fraction_t): fraction_t` qui simplifie la fraction `f`. Par exemple sur l'entrée `650/200` il faut renvoyer `13/4`.