

## Travaux Dirigés – TD 6

### Exercice 43. $\pi$

Etant donné un carré ABCD de côté 1 et le quart de cercle de centre A et de rayon 1 (inscrit dans le carré), écrire un programme qui donne une approximation de  $\pi$  en utilisant la méthode de Monte Carlo qui peut être décrite comme suit : Un joueur tire aléatoirement  $n$  fléchettes dans le carré et comptabilise le nombre de fois  $p$  où l'impact est dans le quart de cercle. La ration  $p/n$  donne une approximation de  $\pi/4$  (la probabilité pour que l'impact soit dans le quart de cercle est égale à l'aire du quart du cercle / aire du carré). La précision dépend de  $n$ . Pour simuler ce jeu, il suffit de tirer  $n$  fois de suite et aléatoirement deux réel  $x, y$  dans  $[0, 1]$  ( $M$  de coordonnées  $(x, y)$  est le point d'impact) et comptabiliser le nombre de fois où  $M$  est dans le quart de cercle.

- Définir un type de structure permettant de définir un point  $M$ .
- Ecrire une fonction booléenne qui détermine si un essai peut être comptabilisé.
- Ecrire une fonction qui donne l'approximation de  $\pi$  selon le nombre  $n$ .
- Donner approximativement le nombre d'essais  $n$  nécessaire pour avoir une précision à  $10^{-3}$ .

### Exercice 44. Nombre complexe

- Définir un premier type de structure permettant de définir un nombre complexe sous forme cartésienne (partie réelle et partie imaginaire). Définir un deuxième type de structure permettant de définir un nombre complexe sous forme polaire (module et argument). Définir un troisième type de structure (Cplx) permettant de représenter un complexe quelle que soit sa forme.
- Ecrire une fonction/procédure qui permet d'obtenir la forme polaire à partir de la forme cartésienne ainsi que la fonction inverse.
- Ecrire deux fonctions/procédures (SomeCplx, SousCplx) permettant de calculer la somme, la soustraction de deux nombres complexes.
- Ecrire deux fonctions/procédures (MulCplx, DivCplx) permettant de calculer le produit et la division de deux nombres complexes.
- Ecrire un programme qui résout une équation de second degré à coefficients complexes.

### Exercice 45. Gestion de parking

Un parking veut faire un programme pour gérer ses places de parking. Le parking peut accueillir des voitures, des camions et des motos. Chacun des véhicules dispose d'un nom de

propriétaire (maximum 20 caractères), d'un matricule (avec un format AA-000-AA), d'une année de mise en circulation (entre 1990 et 2016) et d'un modèle. Les modèles considérés sont : Renault, Peugeot, Citroen, BMW et Audi. Pour les voitures, le nombre de portes est aussi précisé. Pour le camion, le gabarit en mètres est précisé. Les motos ont une vitesse maximale.

Le parking dispose de 20 places de voiture, 10 places de camion et 10 places de moto. Le parking commence avec des places vides. Une place est caractérisée par un pointeur vers le véhicule qui l'occupe, l'heure de réservation et le nombre d'heures de réservation. Les prix du parking diffèrent selon les jours. Un jour de semaine coûtera 5euros/heure et un jour de week-end coûtera 3euros/heure. L'utilisateur doit préciser au moment de la réservation le nombre d'heures et le jour de la réservation. On enregistre l'heure de réservation pour chaque place.

- Créer la structure **vehicule** qui regroupe les trois types. Utiliser les **unions** pour les champs spécifiques de chaque type et une énumération pour préciser le type (voiture, camion et moto).
- Créer les structures parking et place
- Écrire la procédure **reserver** qui permet de réserver une place pour une voiture, un camion ou une moto. Le prix de la réservation devra être affiché avant la réservation.
- Écrire une procédure **update** qui permet de mettre à jour le parking (enlever les véhicules pour lesquels l'horaire de parking est expiré)
- Écrire une procédure **etat** qui permet d'afficher le parking et de donner le nombre de places libres
- Écrire une procédure **retirer** qui permet de retirer un véhicule de sa place.
- Écrire un programme qui permet de gérer le parking avec les fonctionnalités faites précédemment.