TD d'algorithmique – TD1

Exercice 1

Quelles seront les valeurs des variables *a* et *b* à la fin de l'algorithme *Algo1* ?

Algo1

Variables : entiers *a* et *b*
Début
    a ← 1
    b ← *a*+3
    a ← 3
Fin

Exercice 2

Quel est le résultat de l'algorithme *Algo2*(3,5) ?

Algo2

Entrées : entiers *a* et *b*
Variables : entier *c*
Début
    c ← *a*+*b*
    a ← 2
    c ← *b*-*a*
    renvoyer *c*
Fin

Exercice 3

Écrire un algorithme qui prend en entrée deux variables entières *a* et *b* et échange leurs valeurs.

Exercice 4

Écrire en pseudo-code l'algorithme de multiplication égyptienne, décrit ci-contre. Vous supposerez que vous connaissez un algorithme *DiviseParDeux*(*n*) qui pour un entier *n* fourni en entrée renvoie *n*/2, un algorithme *MultiplieParDeux*(*n*) qui pour un entier *n* fourni en entrée renvoie 2*n*, et un algorithme *EstPair*(*n*) qui renvoie VRAI si *n* est pair et renvoie FAUX sinon.

Multiplication égyptienne :
Étant donné deux entiers *a* et *b*, en entrée, on commence par initialiser une variable entière *x* à la valeur 0. On applique ensuite les étapes suivantes :
(1) si *b* est égal à 0, on renvoie *x*.
(2) si *b* est pair, on multiplie *a* par 2 et on divise *b* par 2.
(3) sinon, on enlève 1 à *b* et on ajoute *a* à *x.*
(4) on recommence l'étape (1).

TD d'algorithmique – TD2

Exercice 1

• A quel nombre (en décimal) correspond le nombre suivant en binaire : 10010110 ?
• Ecrivez 42 en binaire
• Ecrivez 84 en binaire
• Que remarquez-vous ?
• Déduisez-en un algorithme **DiviseParDeuxPair** qui prend en entrée une chaîne de caractères qui contient un nombre binaire *n* et s'il est pair, renvoie une chaîne de caractères qui contient la valeur de *n*/2 écrite en binaire. Vous utiliserez l'algorithme **Caractère** qui prend en entrée une chaîne de caractères *chaine* et un entier *i* et renvoie le *i*-ième caractère de chaine, ainsi que l'algorithme **SousChaine** qui renvoie la partie de la chaîne de caractères *chaine* allant du *i*-ième au *j*-ième caractère (inclus), et l'algorithme **Longueur** qui renvoie le nombre de caractères de la chaîne de caractères *chaine*.

Exercice 2

• Dessinez l'organigramme, puis écrivez en pseudo-code un algorithme **EcritureBinaire** qui prend en entrée un entier *n* et renvoie la chaîne de caractères contenant l'écriture binaire de *n*. En plus des algorithmes de la question 1, vous pourrez utiliser l'algorithme **Concatène** qui prend en entrée deux chaînes de caractères *chaine1* et *chaine2* et renvoie la concaténation de ces deux chaines (c'est-à-dire une chaîne de caractères qui contient *chaine1* suivie de *chaine2*.
• Déduisez-en un algorithme **EntierBinaire** qui prend en entrée un entier *n* et renvoie l'entier qui correspond à l'écriture binaire de *n* (par exemple, **EntierBinaire**(7) renverra l'entier 111).

TD d'algorithmique – TD3

Exercice 1 – Le tournoi de foot

Le PSG, le MHSC, l'OL et l'OM décident de faire un tournoi entre eux. Chaque match gagné rapporte 3 points, un match nul rapporte 1 point, et un match perdu ne rapporte rien.
Vous allez écrire un algorithme **Tournoi** qui simule un tournoi de foot, en utilisant les réponses aux questions Q1, Q2, Q3, Q4 et Q5 :
• Q1. Écrivez une suite d'instructions pour créer un tableau **Equipes** qui va stocker les noms des équipes.
• Q2. Écrivez une suite d'instructions pour créer un tableau **Points** qui va stocker pour chaque équipe son nombre de points, et donnez 0 point à chaque équipe pour commencer.
• Q3. Écrivez une suite d'instructions pour créer un tableau **Valeurs** qui va stocker pour chaque équipe sa « valeur », estimée par un nombre de points entre 0 (mauvaise équipe) et 5 (très bonne équipe). À vous de choisir les valeurs de chaque équipe !

• Q4. Supposez que vous avez à disposition un algorithme **Aléatoire** qui prend en entrée deux entiers *i* et *j* et renvoie un nombre aléatoire compris entre *i* (inclus) et *j* (inclus). Écrivez un algorithme **JoueMatch** qui prend en entrée un numéro d'équipe *Equipe1*, un numéro d'équipe *Equipe2*, ainsi que le tableau *Valeurs* de valeurs des équipes, et qui renvoie un tableau à deux cases contenant le score du match (nombre de buts marqués par l'équipe 1 dans la case 1, par l'équipe 2 dans la case 2). Le score du match est calculé aléatoirement de la manière suivante : chaque équipe a marqué un nombre aléatoire de buts compris entre 0 et sa « valeur ».
• Q5. Écrivez un algorithme **MetPointsAJour** qui prend en entrée un numéro d'équipe *Equipe1*, un numéro d'équipe *Equipe2*, un tableau *Points* de scores et un tableau *Valeurs* de valeurs, et simule le match entre les deux équipes à l'aide de **JoueMatch** pour ensuite mettre à jour le tableau *Points* en fonction du résultat.
• Q6. Avec les bouts d'algorithme des questions Q1, Q2 et Q3, et en utilisant l'algorithme **MetPointsAJour**, simulez un tournoi où chacune des équipes joue exactement une fois contre les autres, et affichez le nombre de points final de chaque équipe, puis affichez le nom du gagnant (ou des gagnants ex-aequo).

TD d'algorithmique – TD4 - La séance de cinéma

Demain, les films suivants passent dans la salle 1 de la Ferme du Buisson à Noisiel : *Toutes nos envies* d'Eric Lioret à 14h15 (durée : 2h), *Ceci n'est pas un film* de Jafar Panahi à 16h30 (durée : 1h15), *Drive* de Nicolas Winding Refn à 18h30 (durée : 1h45), et *Plus jamais peur* de Mourad Ben Cheikh à 20h45 (durée : 1h15).
L'objectif de ce TD est de proposer une application qui informe sur le prochain film à voir en fonction de l'heure.
• Q1. Écrivez une suite d'instructions pour créer un tableau *Films* qui va stocker les noms des films.
• Q2. Écrivez une suite d'instructions pour créer un tableau *HeureDebut* qui va stocker pour chaque film son heure de début. On vous conseille d'utiliser des nombres à virgule pour coder les heures, par exemple 16,5 pour 16h30 ou 14,25 pour 14h15.
• Q3. Écrivez une suite d'instructions pour créer un tableau *Duree* qui va stocker les durées des films (aussi codées par des nombres à virgule).
• Q4. Écrivez un algorithme **CodeHeure** qui prend en entrée un entier *heure* et un entier *minute*, et qui renvoie en sortie l'heure correspondante codée par un nombre à virgule. Par exemple, comme vu à la question Q2, **CodeHeure**(14,15) va renvoyer le nombre à virgule 14,25. ***Indication :*** *utiliser la règle de trois !*
• Q5. Écrivez un algorithme **ProchainFilm** qui prend en entrée un entier *heure* et un entier *minute* qui indiquent l'heure qu'il est, un tableau *Films* (comme celui de la Q1), un tableau *HeureDebut* (comme celui de la Q2) et renvoie le nom du prochain film s'il y en a un, ou "Plus de film aujourd'hui" sinon.
• Q6. Écrivez un algorithme **FilmEnCours** qui prend en entrée un entier *heure* et un entier *minute* qui indiquent l'heure qu'il est, un tableau *Films* (comme celui de la Q1), un tableau *HeureDebut* (comme celui de la Q2) et un tableau *Duree* (comme celui de la Q3) et renvoie le nom du film en cours s'il y en a un, ou "Aucun film en cours" sinon.
• Q7. Écrivez un algorithme **TauxDiffusion** qui prend en entrée un tableau *Duree* (comme celui de la Q3) et renvoie la fraction du temps passé dans la journée à diffuser des films. Par exemple pour le tableau *Duree* de demain à Noisiel, cette fraction est : (2+1,25+1,75+1,25)/24=0,26

TD d'algorithmique – TD5 – Images en noir et blanc

Le but de ce TD est de manipuler des tableaux de tableaux de booléens qui représentent des images en noir et blanc. En effet, on va coder une image en noir et blanc de la manière suivante : chaque pixel noir correspond au booléen FAUX (pixel éteint) et chaque pixel blanc au booléen VRAI (pixel allumé). Un tableau de booléens correspond donc à une colonne de pixels, et un tableau de tableaux de booléens correspond à une image. Par exemple, l'image ci-contre correspond au tableau {{VRAI,VRAI,VRAI},{VRAI,VRAI,FAUX},{FAUX,FAUX,VRAI}}.

• Q1. Dessinez l'image qui correspond au tableau {{VRAI,VRAI},{FAUX,VRAI},{VRAI,FAUX}}.
• Q2. Écrivez l'instruction en pseudo-code qui permet de stocker dans une variable *zigzag* le tableau de tableaux de booléens correspondant à l'image suivante :  ?
• Q3. Quelle instruction en pseudo-code stocke dans une variable *PixelEnBasADroite* la valeur (VRAI ou FAUX) correspondant à la couleur (blanc ou noir) du pixel en bas à droite de *zigzag* ?
• Q4. Écrivez un algorithme **LargeurImage** qui prend en entrée un tableau de tableaux de booléens codant une image en noir et blanc et qui renvoie la largeur de l'image correspondante. Par exemple, **LargeurImage**(*zigzag*) renvoie 4.
• Q4'. Même question pour l'algorithme **HauteurImage** (bien sûr **HauteurImage**(*zigzag*) renvoie 2).
• Q5. Écrivez un algorithme **CompteBlancsColonne** qui prend en entrée un tableau de booléens et renvoie le nombre de cases contenant VRAI dans ce tableau. Par exemple, **CompteBlancsColonne**({VRAI,FAUX,VRAI,VRAI}) renvoie 3.
• Q6. Sans faire de boucle (en utilisant un appel de l'algorithme **CompteBlancsColonne**), écrivez un algorithme **CompteNoirsColonne** qui prend en entrée un tableau de booléens et renvoie son nombre de cases contenant FAUX.
• Q7. Écrivez un algorithme **CompteBlancs** qui prend en entrée un tableau de tableaux de booléens codant une image et renvoie le nombre de pixels blancs dans l'image codée par ce tableau de tableau de booléens.
• Q8. On dispose d'un système qui permet d'afficher un pixel blanc pour 0.2€ par heure, et d'afficher un pixel noir pour 0.1€ par heure. Écrivez un algorithme **CoutAffichage** qui prend en entrée un tableau de tableaux de booléens *tabImage* codant une image et deux entiers *h* et *m*, et renvoie le coût d'affichage de l'image pendant *h* heures et *m* minutes.