



Université
de Limoges

FACULTÉ
DES SCIENCES
ET TECHNIQUES

Département d'Informatique

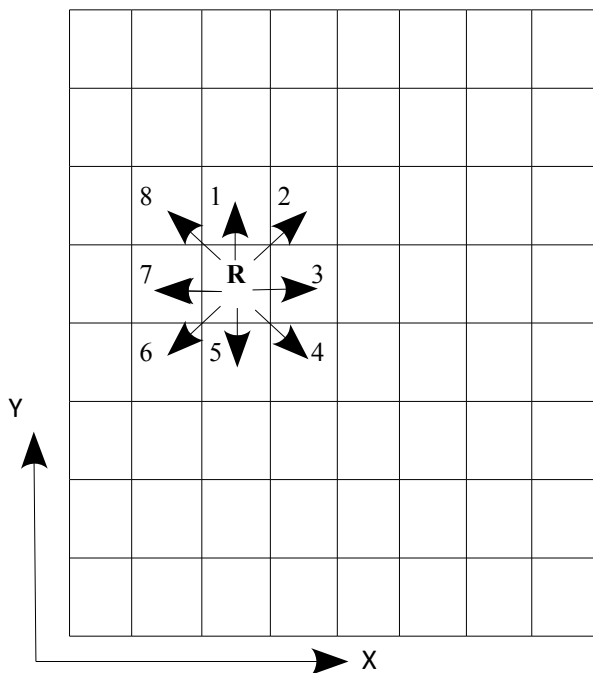
Devoir Blanc Novembre 2012

Durée : 1 heure 30 Cours, documents autorisés

NB : une rédaction soignée des algorithmes est exigée.

Question 1 (7 points)

Sur un échiquier, le Roi se déplace dans toutes les directions d'**une seule case** comme ci-dessous.



Écrire un algorithme qui :

- demande à l'utilisateur la position initiale du Roi et vérifie que la position est sur l'échiquier. Si ce n'est pas le cas, on redemandera la position à l'utilisateur.
- puis en boucle 10 fois de suite :
 - demande la direction à prendre : 1, 2, 3, 4, 5, 6, 7 ou 8 et vérifie que la direction soit valide (si ce n'est pas le cas, on redemandera une direction à l'utilisateur) ;
 - puis soit calcule et affiche sa nouvelle position, soit envoie un message d'erreur si le déplacement aboutit à faire sortir la pièce de l'échiquier.
- Affiche « Gagné ! » si les 10 déplacements ont pu être effectués sans que le Roi ne sorte de l'échiquier et « Perdu après B coups ! » sinon (où B est le nombre de coups joués avant de sortir de l'échiquier).

Remarque : Évidemment si le Roi aboutit hors de l'échiquier lors d'un de ses déplacements, l'algorithme ne redemandera pas l'utilisateur de saisir un nouveau déplacement.

Question 2 (3 points)

Exprimer les conditions suivantes en les développant sous leur forme la plus simple (c'est à dire en les simplifiant au maximum). Il est d'ailleurs peut être possible de donner directement le résultat (VRAI ou FAUX) de certaines de ces expressions. On suppose que X est une variable de type caractère et que A et B sont de type entier.

1. NON ((X='C') ET ((B>A) OU (A>2)))
2. NON (NON(2=2) ET (A<10))
3. NON ((X='X') OU (X≠'X'))
4. NON (NON(NON(A div B ≠ 0) OU (NON(A<2) OU NON(X='C'))))
5. NON (FAUX OU (NON (A=B) ET (B≥10)))
6. NON (((A>B) ET (B<10)) ET NON((A>B) ET (B<10)))

Question 3 (7 points)

Une compagnie d'assurance propose d'assurer l'automobile de ses clients. Afin de proposer un tarif au client, la compagnie doit tout d'abord déterminer son profil. Le profil est déterminé suivant le sexe et l'âge de l'assuré de la façon suivante :

Femmes de moins de 20 ans :	B
Femmes de 20 à 49 ans :	A
Femmes de 50 ans et plus :	C
Hommes de moins de 40 ans :	B
Hommes de 40 à 59 ans :	C
Hommes de 60 ans et plus :	D

Le montant annuel de la cotisation d'assurance est donné en fonction du profil et du nombre d'accidents que l'assuré a déjà subi :

		Nombre d'accidents		
		0	1 à 3	Plus de 3
Profil	A	230	240	250
	B	240	250	260
	C	250	260	270
	D	260	270	280

Remarque : Les femmes étant plus prudentes, on constate qu'elles ont un tarif préférentiel (sauf lorsqu'elles sont très jeunes conductrices ou particulièrement âgées).

Écrire un algorithme qui :

1. demande le nom, l'âge et le sexe d'un assuré et détermine son profil
2. demande le nombre d'accidents que l'assuré a déjà eu et calcule le montant de la cotisation annuelle
3. affiche les résultats.

Les affichages et saisies doivent être conforme avec le modèle d'écran ci-dessous :

Quel est le nom de l'assuré : Oscar
Quel est le sexe (H ou F) de l'assuré Oscar : H
Quel est l'age de l'assuré Oscar : 42
Quel est le nombre d'accidents que Oscar a déjà eu : 5

L'assuré Oscar a un profil C et il doit payer 270 euros de cotisation annuelle pour son assurance automobile.

Les parties soulignées sont affichées par le système et le reste est saisi par l'utilisateur. Vous devez d'abord calculer les résultats, c'est-à-dire stocker les résultats dans des variables, puis les faire afficher à la fin.

Question 4 (3 points) :

On considère l'algorithme suivant :

```

Algorithme comportement_inconnu
{
    variable A, AL, AS: entier
    c : caractère

    Afficher("Saisissez un nombre entier : ")
    Saisir(A)
    AS <-- A
    AL <-- AS mod 100
    Tant que (AL!=0)
    {
        Selon (AL)
        {
            10 : c <-- 'A'
            11 : c <-- 'B'
            12 : c <-- 'C'
            13 : c <-- 'D'
            14 : c <-- 'E'
            15 : c <-- 'F'
            16 : c <-- 'G'
            17 : c <-- 'H'
            18 : c <-- 'I'
            19 : c <-- 'J'
            20 : c <-- 'K'
            21 : c <-- 'L'
            22 : c <-- 'M'
            23 : c <-- 'N'
            24 : c <-- 'O'
            25 : c <-- 'P'
            26 : c <-- 'Q'
            27 : c <-- 'R'
            28 : c <-- 'S'
            29 : c <-- 'T'
            30 : c <-- 'U'
            31 : c <-- 'V'
            32 : c <-- 'W'
            33 : c <-- 'X'
            34 : c <-- 'Y'
            35 : c <-- 'Z'
            sinon : c <-- '?'
        }
        Afficher(c)
        AS <-- AS div 100
        AL <-- AS mod 100
    }
}

```

Exécuter cet algorithme avec les entrées 27302419232411 puis 1423161016.

Vous n'oubliez pas de dire ce que fait cet algorithme.

Corrige

Question 1 :

```
Algorithme Deplacement_roi
{
    constante entier NOMBRE_COUPS_MAX <--- 10
    variable      x, y : entier // pour la position de la tour
                  direction : entier
                  nombre_coups : entier
                  dedans : logique

    nombre_coups <--- 0

    Afficher("Saisir la position initiale du roi en x et en y : ")
    Saisir(x,y)

    Tant que ((x<=0) OU (x>=9) OU (y<=0) OU (y>=9))
    {
        Afficher("La position initiale saisie est en dehors de l'échiquier.\n")
        Afficher("Saisir une position initiale valide pour le roi en x et en y : ")
        Saisir(x,y)
    }

    // Ici on est sur que le roi est bien sur l'échiquier
    dedans <--- VRAI

    Tant que ((nombre_coups < NOMBRE_COUPS_MAX) ET dedans)
    {
        Afficher("Saisir la direction du déplacement à effectuer : ")
        Saisir(direction)

        Tant que ((direction<=0) OU (direction>=9))
        {
            Afficher("La direction saisie est invalide.\n")
            Afficher("Saisir à nouveau la direction du déplacement à effectuer : ")
            Saisir(direction)
        }

        Selon (direction)
        {
            1 : y <--- y + 1
            2 : y <--- y + 1
                x <--- x + 1
            3 : x <--- x + 1
            4 : y <--- y - 1
                x <--- x + 1
            5 : y <--- y - 1
            6 : y <--- y - 1
                x <--- x - 1
            7 : x <--- x - 1
            8 : y <--- y + 1
                x <--- x - 1
        }

        nombre_coups <--- nombre_coups + 1

        Si ((x<=0) OU (x>=9) OU (y<=0) OU (y>=9)) alors
            dedans <--- FAUX
        sinon
            Afficher("Coup n°",nombre_coups," : le roi est maintenant en (" ,x," ,y," )\n")
        }

        Si(dedans) alors
            Afficher("Gagné !\n")
        sinon
            Afficher("Perdu après ",nombre_coups," coups !\n")
        }
    }
```

Question 2 :

1. $\text{NON} ((X='C') \text{ ET } ((B>A) \text{ OU } (A>2)))$
équivalent à $\text{NON} (X='C') \text{ OU } \text{NON} ((B>A) \text{ OU } (A>2))$
équivalent à $(X\neq'C') \text{ OU } ((B\leq A) \text{ ET } (A\leq 2))$
2. $\text{NON} (\text{NON}(2=2) \text{ ET } (A<10))$
équivalent à $\text{NON} (\text{FAUX} \text{ ET } (A<10))$
équivalent à $\text{NON} (\text{FAUX})$
équivalent à **VRAI**
3. $\text{NON} ((X='X') \text{ OU } (X\neq'X'))$
équivalent à $\text{NON} (\text{VRAI})$ puisque forcément $(X='X')$ « **ou** » son opposé $(X\neq'X')$ est **VRAI**
équivalent à **FAUX**
4. $\text{NON} (\text{NON}(\text{NON}(A \text{ div } B \neq 0) \text{ OU } (\text{NON}(A<2) \text{ OU } \text{NON}(X='C'))))$
équivalent à $\text{NON}(A \text{ div } B \neq 0) \text{ OU } (\text{NON}(A<2) \text{ OU } \text{NON}(X='C'))$
équivalent à $(A \text{ div } B = 0) \text{ OU } ((A \geq 2) \text{ OU } (X\neq'C'))$
5. $\text{NON} (\text{FAUX} \text{ OU } (\text{NON} (A=B) \text{ ET } (B \geq 10)))$
équivalent à $\text{NON} (\text{NON} (A=B) \text{ ET } (B \geq 10))$
équivalent à $\text{NON}(\text{NON} (A=B)) \text{ OU } \text{NON}(B \geq 10)$
équivalent à $(A=B) \text{ OU } (B < 10)$
6. $\text{NON} ((A>B) \text{ ET } (B<10)) \text{ ET } \text{NON}((A>B) \text{ ET } (B<10))$
C'est **VRAI** puisque $((A>B) \text{ ET } (B<10))$ « **et** » son opposé (soit $\text{NON}((A>B) \text{ ET } (B<10))$) est forcément **FAUX**.

Question 3 :

Algorithme Assurance_automobile

```
{
    variable      age, nbAccidents, tarif : entier
                  nom : chaîne de caractères
                  profil, sexe : caractère

    Afficher("Quel est le nom de l'assuré : ")
    Saisir(nom)
    Afficher("Quel est le sexe (H/F) de l'assuré ", nom, " : ")
    Saisir(sexe)
    Afficher("Quel est l'age de l'assuré ", nom, " : ")
    Saisir(age)
    Afficher("Quel est le nombre d'accidents que ", nom, " a déjà eu : ")
    Saisir(nbAccidents)

    Si ((sexe='F')) alors
    {
        Si (age < 20) alors
            profil <--- 'B'
        sinon
            Si (age > 49) alors
                profil <--- 'C'
            sinon
                profil <--- 'A'
        }
    sinon // on supposera que l'utilisateur saisit bien 'F' ou 'H'
    {
        Si (age < 40) alors
            profil <--- 'B'
        sinon
            Si (age > 59) alors
                profil <--- 'D'
            sinon
                profil <--- 'C'
        }
    }

    Selon (profil)
    {
        'A' : Si (nbAccidents = 0) alors
            tarif <--- 230
        sinon
            Si (nbAccidents < 3) alors
                tarif <--- 240
            sinon
                tarif <--- 250
        'B' : Si (nbAccidents = 0) alors
            tarif <--- 240
        sinon
            Si (nbAccidents < 3) alors
                tarif <--- 250
            sinon
                tarif <--- 260
        'C' : Si (nbAccidents = 0) alors
            tarif <--- 250
        sinon
            Si (nbAccidents < 3) alors
                tarif <--- 260
            sinon
                tarif <--- 270
        'D' : Si (nbAccidents = 0) alors
            tarif <--- 260
        sinon
            Si (nbAccidents < 3) alors
                tarif <--- 270
            sinon
                tarif <--- 280
        }
    }

    Afficher("\nL'assuré ", nom, " a un profil ", profil, " et il doit payer ", tarif, " euros de cotisation annuelle pour son assurance automobile.\n")
}
```

Question 4 :

A	AL	AS	C	commentaires
0	0	0	"	
27302419232411				// la saisie
		27302419232411		
	11			
			'B'	// on affiche 'B'
		273024192324		
	24			
			'O'	// on affiche 'O'
		2730241923		
	23			
			'N'	// on affiche 'N'
		27302419		
	19			
			'J'	// on affiche 'J'
		273024		
	24			
			'O'	// on affiche 'O'
		2730		
	30			
			'U'	// on affiche 'U'
		27		
	27			
			'R'	// on affiche 'R'
		0		
	0			

A	AL	AS	C	commentaires
0	0	0	"	
1423161016				// la saisie
		1423161016		
	16			
			'G'	// on affiche 'G'
		14231610		
	10			
			'A'	// on affiche 'A'
		142316		
	16			
			'G'	// on affiche 'G'
		1423		
	23			
			'N'	// on affiche 'N'
		14		
	14			
			'E'	// on affiche 'E'
		0		
	0			

L'algorithme affiche un message correspondant au nombre saisi (en quelque sorte, il « décrypte » un message codé par le nombre si cela est possible).