



Université
de Limoges

FACULTÉ
DES SCIENCES
ET TECHNIQUES

Département d'Informatique

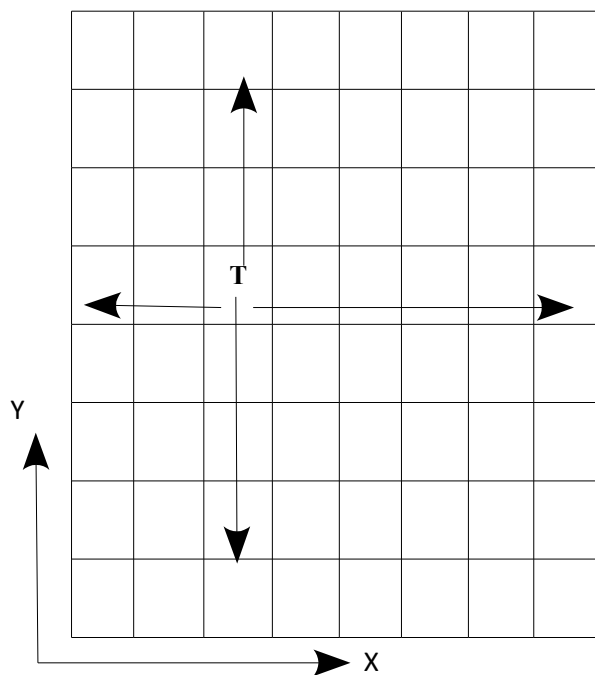
Devoir Blanc Novembre 2011

Durée : 1 heure 30 Cours, documents autorisés

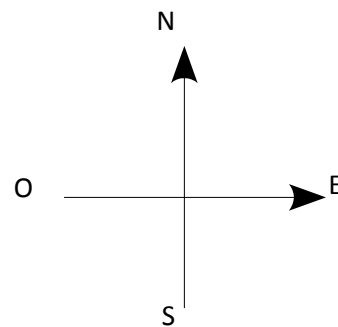
NB : une rédaction soignée des algorithmes est exigée.

Question 1 (7 points)

Sur un échiquier, les tours se déplacent en ligne droite comme ci-dessous.



Exemples de directions : N (Nord)
ou S (Sud) ou E (Est) ou O (Ouest)



Écrire un algorithme qui :

- demande à l'utilisateur la position initiale de la tour
- puis en boucle 10 fois de suite :
 - demande la direction à prendre (N, S, E ou O) et le nombre de pas
 - 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 la tour 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 1 : Évidemment si la tour aboutit hors de l'échiquier lors d'un de ses déplacements, l'algorithme ne redemandera pas l'utilisateur de saisir un nouveau déplacement.

Remarque 2 : Si la direction de déplacement saisie est erronée, l'algorithme sortira de la boucle et renverra un message expliquant l'erreur et pourquoi l'utilisateur a perdu.

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') OU ((B>A) ET (A>2)))
2. NON (NON(2=2) ET (A<10))
3. NON ((X='X') OU (X≠'X'))
4. NON (NON(NON(A mod 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)

Lors de compétitions cyclosporatives (*), les concurrents sont répartis en catégories, comme suit:

Femmes de moins de 38 ans :	A
Femmes de 38 ans et plus :	B
Hommes de moins de 24 ans :	C
Hommes de 24 à 39 ans :	D
Hommes de 40 à 52 ans :	E
Hommes de 53 ans et plus :	F

(*) *classement simplifié, il y a plus de catégories en réalité*

On donne ensuite les limites de temps qui serviront à délivrer des diplômes d'or, d'argent ou de bronze (pour simplifier, les temps sont donnés en minutes) :

Catégorie	Diplôme d'or	Diplôme d'argent	Diplôme de bronze
A	Moins de 230	De 231 à 260	261 et plus
B	Moins de 260	De 261 à 280	281 et plus
C	Moins de 180	De 181 à 220	221 et plus
D	Moins de 200	De 201 à 240	241 et plus
E	Moins de 220	De 221 à 250	251 et plus
F	Moins de 240	De 241 à 260	261 et plus

Écrire un algorithme qui :

1. demande l'âge et le sexe d'un concurrent et calcule sa catégorie
2. demande le temps qu'il a réalisé et calcule la nature de son diplôme
3. affiche les résultats selon le modèle d'écran ci-dessous:

Nom : Oscar
Sexe (M ou F) : M
Age : 42
Temps réalisé : 231

Modèle d'écran

Oscar est en catégorie E, il a obtenu le diplôme d'argent

Les parties soulignées sont affichées par le système. 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, B, C, X : entier

    Afficher("Saisissez un nombre entier : ")
    Saisir(A)
    Afficher("Saisissez un nombre entier : ")
    Saisir(B)
    Afficher("Saisissez un nombre entier : ")
    Saisir(C)

    Si (A < B) alors
        Si (B < C) alors
            X ← B
        sinon
            Si (A < C) alors
                X ← C
            sinon
                X ← A
    sinon
        Si (A < C) alors
            X ← A
        sinon
            Si (B < C) alors
                X ← C
            sinon
                X ← B

    Afficher("Le résultat est : ", X)
}
```

On simulera trois exécutions des l'algorithme en supposant qu'un utilisateur obéissant saisisse les entrées (8, -1, 2) pour A, B et C lors de la première exécution puis (-5, 1, 4) lors de la seconde et (7, 6, 5) lors de la dernière.

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

Corrige

Question 1 :

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

    nombre_coups <--- 0

    Afficher("Saisir la position initiale de la tour en x et en y : ")
    Saisir(x,y)

    Si ((x>0) ET (x<9) ET (y>0) ET (y<9)) alors
    {
        dedans <--- VRAI
        erreur <--- FAUX
    }
    sinon
    {
        dedans <--- FAUX
        erreur <--- VRAI
    }

    Tant que ((nombre_coups < NOMBRE_COUPS_MAX) ET dedans ET NON(erreur))
    {
        Afficher("Saisir la direction du déplacement à effectuer : ")
        Saisir(direction)
        Afficher("Saisir le nombre de pas dont vous voulez déplacer la tour : ")
        Saisir(nombre_pas)

        Selon (direction)
        {
            'N' : y <--- y + nombre_pas
            'S' : y <--- y - nombre_pas
            'E' : x <--- x + nombre_pas
            'O' : x <--- x - nombre_pas
            sinon : erreur <--- VRAI
        }

        Si (NON(erreur)) alors
        {
            nombre_coups <--- nombre_coups + 1
            Si ((x<1) ou (y<1) ou (x>8) ou (y>8)) alors
                dedans <--- FAUX
            sinon
                Afficher("Coup n°",nombre_coups," : la tour est maintenant en (" ,x," ,",y," )\n")
        }
    }

    Si (erreur) alors
    {
        Si (dedans) alors
            Afficher("La direction du déplacement saisi n'existait pas. On décide donc que vous avez perdu
après les ",nombre_coups," coups précédents.\n")
        sinon
            Afficher("La position d'origine était en dehors de l'échiquier.\n")
        }
    sinon
    {
        Si(dedans) alors
            Afficher("Gagné !\n")
        sinon
            Afficher("Perdu après ",nombre_coups," coups !\n")
        }
    }
}
```

Question 2 :

1. $\text{NON}((X='C') \text{ OU } ((B>A) \text{ ET } (A>2)))$
équivalent à $\text{NON}(X='C') \text{ ET } \text{NON}((B>A) \text{ ET } (A>2))$
équivalent à $(X \neq 'C') \text{ ET } ((B \leq A) \text{ OU } (A \leq 2))$
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. $\text{NON}(\text{NON}(\text{NON}(A \bmod B = 0) \text{ OU } (\text{NON}(A<2) \text{ OU } \text{NON}(X='C'))))$
équivalent à $\text{NON}(A \bmod B = 0) \text{ OU } (\text{NON}(A<2) \text{ OU } \text{NON}(X='C'))$
équivalent à $(A \bmod B \neq 0) \text{ OU } ((A \geq 2) \text{ OU } (X \neq 'C'))$
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 Competition_cyclo

```
{  
    variable    age, temps : entier  
               nom, diplome : chaîne de caractères  
               categorie, sexe : caractère  
  
    Afficher("Quel est le nom du concurrent ?")  
    Saisir(nom)  
    Afficher("Quel est le sexe (M/F) du concurrent ", nom, " ?")  
    Saisir(sexe)  
    Afficher("Quel est l'age du concurrent ", nom, " ?")  
    Saisir(age)  
    Afficher("Quel est le temps (en minutes) du concurrent ", nom, " ?")  
    Saisir(temps)
```

```

Si ((sexe='F')) alors
{
    Si (age < 38) alors
        categorie <--- 'A'
    sinon
        categorie <--- 'B'
}
sinon // on supposera que l'utilisateur saisit bien 'F' ou 'M'
{
    Si (age < 24) alors
        categorie <--- 'C'
    sinon
        Si (age < 40) alors
            categorie <--- 'D'
        sinon
            Si (age < 53) alors
                categorie <--- 'E'
            sinon
                categorie <--- 'F'
}

Selon (categorie)
{
    'A' : Si (temps < 231) alors
        diplome <--- "diplôme d'or"
    sinon
        Si (temps < 261) alors
            diplome <--- "diplôme d'argent"
        sinon
            diplome <--- "diplôme de bronze"
    'B' : Si (temps < 261) alors
        diplome <--- "diplôme d'or"
    sinon
        Si (temps < 281) alors
            diplome <--- "diplôme d'argent"
        sinon
            diplome <--- "diplôme de bronze"
    'C' : Si (temps < 181) alors
        diplome <--- "diplôme d'or"
    sinon
        Si (temps < 221) alors
            diplome <--- "diplôme d'argent"
        sinon
            diplome <--- "diplôme de bronze"
    'D' : Si (temps < 201) alors
        diplome <--- "diplôme d'or"
    sinon
        Si (temps < 241) alors
            diplome <--- "diplôme d'argent"
        sinon
            diplome <--- "diplôme de bronze"
    'E' : Si (temps < 221) alors
        diplome <--- "diplôme d'or"
    sinon
        Si (temps < 251) alors
            diplome <--- "diplôme d'argent"
        sinon
            diplome <--- "diplôme de bronze"
    'F' : Si (temps < 241) alors
        diplome <--- "diplôme d'or"
    sinon
        Si (temps < 261) alors
            diplome <--- "diplôme d'argent"
        sinon
            diplome <--- "diplôme de bronze"
}
Afficher(nom," est en catégorie ",categorie," , il a obtenu le ",diplome,".\n")
}

```

Question 4 :

A	B	C	X	commentaires
0	0	0	0	
8	-1	2	0	// les différentes saisies
			2	// on affiche le résultat est 2

A	B	C	X	commentaires
0	0	0	0	
-5	1	4	0	// les différentes saisies
			4	// on affiche le résultat est 4

A	B	C	X	commentaires
0	0	0	0	
7	6	5	0	// les différentes saisies
			6	// on affiche le résultat est 6

L'algorithme affiche la valeur intermédiaire du triplet A, B et C (c'est à dire, ni le maximum et ni le minimum).