Université A. Mira de Bejaia
Faculté de Technologie
Département de Technologie

Année universitaire 2024/2025
1ère Année Technologie

TP Informatique 1

Corrigé de la série de TP N°2

But du TP :

Le but de ce TP est de comprendre et manipuler les conversions de types, les expressions arithmétiques, ainsi que les variables et identificateurs en langage Pascal. Vous permettra d'acquérir une maîtrise des concepts fondamentaux nécessaires à la programmation en Pascal.

Exercice N°01 :*(Systèmes de numérotation)*

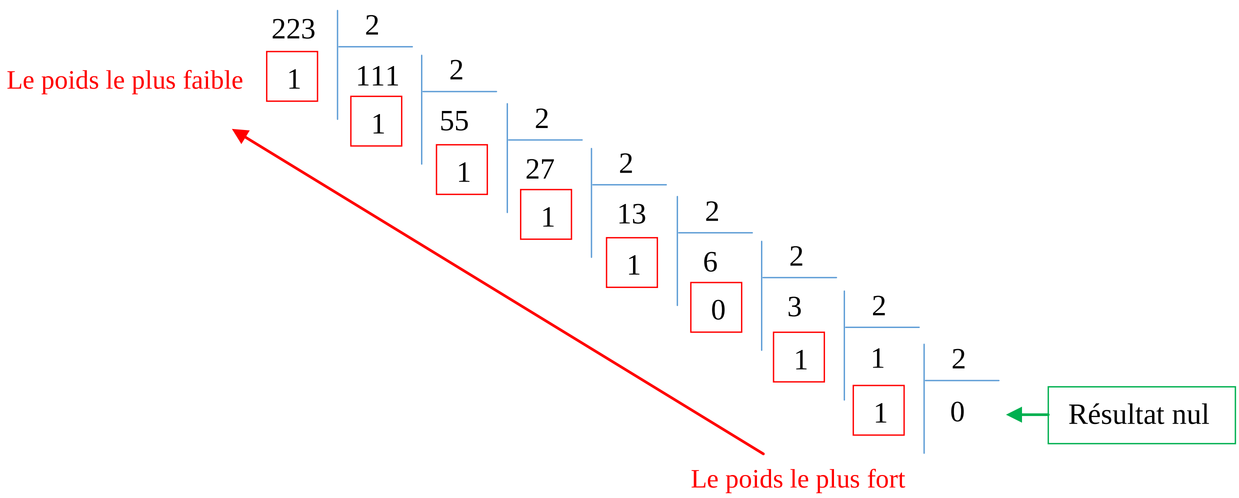
Effectuer les conversions suivantes :

$$223 = (?)_2$$

Rappel :

Conversion de la base 10 → base 2, 8, 16

Soit Nb un nombre exprimé dans la base 10, pour trouver son équivalent en base b, on applique la méthode des divisions successives sur b, jusqu'à l'obtention d'un résultat nul. Puis, on récupère les restes des divisions dans le sens inverse, i.e. le dernier reste trouvé représentera le poids le plus fort et le premier reste trouvé sera le poids le plus faible.



$$223 = (11011111)_2$$

$(101100101)_2 = (?)_{10}$

Rappel :

Conversion de la base 2, 8, 16 → base 10

Pour convertir un nombre $Nb = (a_{n-1}a_{n-2} \dots\ a_1 a_0)_b$ de la base b vers la base 10, on effectue le calcul suivant :

$$(Nb)_b = (a_{n-1} * b^{n-1} + a_{n-2} * b^{n-2} \dots +a_1 * b^1 + a_0 * b^0)_{10}$$

$$(Nb)_b = \sum_{i=0}^{n-1} a_i * b^i$$

$(101100101)_2 = (1 * 2^8 + 0 * 2^7 + 1 * 2^6 + 1 * 2^5 + 0 * 2^4 + 0 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0)_{10}$;

$(101100101)_2 = (357)_{10}$;

$(110011010)_2 = (?)_8 = (?)_{16}$

Rappel 1:

Conversion de la base 2 → base 8

Pour trouver l'équivalent d'un nombre binaire en octal, il suffit de former des **groupes de3 bits** chacun (Puisque **8 = 2^3**), en commençant du poids le plus faible (à partir de la droite), si le dernier groupe formé possède moins de 3 bits, il suffit de rajouter des 0, puis calculer l'équivalent en octal de chaque groupe.

Rappel 2:

Conversion de la base 2 → base 16

Pour trouver l'équivalent d'un nombre binaire en Hexadécimal, il suffit de former des **groupes de 4 bits** chacun (Puisque **16 = 2^4**), en commençant du poids le plus faible (à partir de la droite), si le dernier groupe formé possède moins de 4 bits, il suffit de rajouter des 0, puis calculer l'équivalent en Hexadécimal de chaque groupe.

$(110|011|010)_2 = (632)_8$

6 3 2

$(0001|1001|1010)_2 = (19A)_{16}$

1 9 A

$(3716)_8 = (?)_2$

Rappel :

Conversion de la base 8 → base 2

Pour convertir un nombre Nb exprimé en base 8 vers la base 2, nous procédons comme suit:

$$8 = 2^3$$

Il faut donc utiliser **3** bits pour exprimer un seul chiffre octal en binaire.

Chiffre en hexadécimal	Chiffre équivalent en binaire (2^3 $2^2 2^1$ 2^0)
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
A	1 0 1 0
B	1 0 1 1
C	1 1 0 0
D	1 1 0 1
E	1 1 1 0
F	1 1 1 1

Chiffre en octal	Chiffre équivalent en binaire ($2^2 2^1$ 2^0)
0	0 0 0
1	0 0 1
2	0 1 0
3	0 1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1

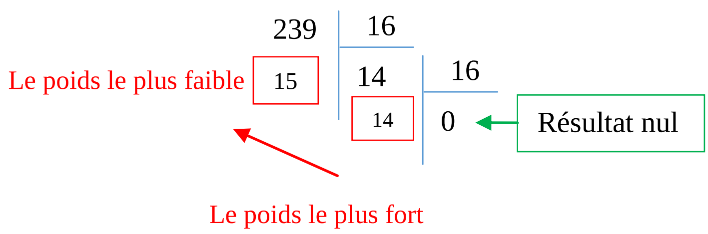
$(3716)_8 = (011111001110)_2$ *ou bien* $(11111001110)_2$;

$(239)_{10} = (?)_{16}$

Rappel :

Conversion de la base 10 → base 2, 8, 16

Soit Nb un nombre exprimé dans la base 10, pour trouver son équivalent en base b, on applique la méthode des divisions successives sur b, jusqu'à l'obtention d'un résultat nul. Puis, on récupère les restes des divisions dans le sens inverse, i.e. le dernier reste trouvé représentera le poids le plus fort et le premier reste trouvé sera le poids le plus faible.



$(239)_{10} = (EF)_{16}$

$(3DB)_{16} = (?)_{10}$

Rappel :

Conversion de la base 2, 8, 16 → base 10

Pour convertir un nombre $Nb = (a_{n-1}a_{n-2} \dots a_1 a_0)_b$ de la base b vers la base 10, on effectue le calcul suivant :

$$(Nb)_b = (a_{n-1} * b^{n-1} + a_{n-2} * b^{n-2} \ldots +a_1 * b^1 + a_0 * b^0)_{10}$$

$$(Nb)_b = \sum_{i=0}^{n-1} a_i * b^i$$

$(3DB)_{16} = (3 * 16^2 + 13 * 16^1 + 11 * 16^0)_{10}$

$$(3\text{DB})_{16} = (987)_{10}$$

$$(3\text{DA})_{16} = (?)_8$$

Rappel :

Conversion de la base 16 → base 8

Pour convertir un nombre Nb exprimé en base 16 vers la base 8 ou vice versa, nous devons **passer par une base intermédiaire**tel que le **décimal ou le binaire**, mais le passage par le binaire est beaucoup plus simple.

$(3\text{DA})_{16} = (001|111|011|010)_2 = (1732)_8$

1 7 3 2

$(3\text{DA})_{16} = (1732)_8$

$(32103)_4 = (?)_2$

Rappel :

Conversion de la base 4→ base 2

Pour convertir un nombre Nb exprimé en base 4 vers la base 2, nous procédons comme suit:

$$4 = 2^2$$

Il faut donc utiliser 2bits pour exprimer un seul chiffre octal en binaire.

$(32103)_4 = (11|10|01|00|11)_2$;

3 2 1 0 3

$(32103)_4 = (1110010011)_2$;

Chiffre en base 4	Chiffre équivalent en binaire (2^1 2^0)
0	0 0
1	0 1
2	1 0
3	1 1

Exercice N°02 :*(Expressions arithmétiques en Algorithme/Pascal)*

a) Traduire les expressions suivantes en langage Pascal :

Expressions mathématiques	**Pascal**
$y1 = b^2 - 4ac$	$y1 := sqr(b) - 4 * a * c$ ou bien $y1 := b * b - 4 * a * c$
$y2 = e^{3a} + \|b\|$	$y2 := exp(3 * a) + abs(b)$
$y3 = x^2 + \sqrt{\frac{\|2x\| + \sqrt{x}}{2e^x}}$	$y3 := sqr(x) + sqrt((abs(2 * x) + sqrt(x))/(2 * exp(x)))$
$y4 = e^{\sqrt{5x + \|-3x\|}}$	$y4 := exp(sqrt(5 * x + abs(-3 * x)))$
$y5 = \frac{e^{\sqrt{a5}} - 4e^{2b} + \sqrt{\|ba^2 - b\|}}{(\frac{a}{b})b^2}$	$y5 := (\exp(sqrt(a * 5)) - 4 * \exp(2 * b) + sqrt(abs(b * sqr(a) - b)))/((a/b) * sqr(b))$

b) l'expression arithmétique correspondante de l'expression écrite en pascal suivante :

$$Z = \frac{\sqrt{\left|2x - 1 + (\frac{y}{2})\right|}}{x^2 - 2xy} + 2(x + y)^2$$

c) Définir les opérateurs **DIV** et **MOD** en donnant deux exemples numériques pour chacun.

Div : permet d'obtenir une division entière (ou la partie entière d'une division)

Mod : permet d'obtenir le reste de la division.

Exemples :7 div 2= 3 ; 8 div 4 = 2

7 mod 2= 1 ; 6 mod 3 = 0

Exercice N°03 :*(Evaluation des expressions)*

Rappel :

L'évaluation d'une expression consiste à calculer, au fur et à mesure, les résultats des calculs jusqu'à obtenir un résultat finale. Cela se fait en plusieurs étapes :

- Écrire l'expression sous forme linéaire (Il faut noter qu'en algorithmique, les expressions s'écrivent sous forme linéaire: $\frac{(x+z)}{(y*2)} \rightarrow (x + z)/(y * 2);$
- Remplacer les identifiants (c'est à dire les noms) des variables et des constantes par leurs valeurs ;
- Évaluer (Calculer) étape par étape chacune des sous-expressions en commençant par les sous-expressions qui sont dans les parenthèses les plus internes.
- Indiquer à chaque calcul, le rang d'évaluation.

La priorité des opérateurs dans les expressions arithmétiques, logiques et relationnelles est comme suit :

1. Les parenthèses ;
2. Les fonctions ;
3. Le moins unaire, le Not ;
4. *, /, Div, Mod, And
5. +, -, Or
6. =, <>, <, >, <=, >=

❖ **Remarque :** Si les opérateurs ont le même rang de priorité, l'évaluation se fait de gauche à droite.

Expression 1 : $50 + \underbrace{3\ \mathbf{MOD}\ 2}_{(1)} - 4\ \mathbf{DIV}\ 3 + 40$

$50 + 1 - \underbrace{4\ \mathbf{DIV}\ 3}_{(2)} + 40$

$\underbrace{50 + 1}_{(3)} - 1 + 40$

$\underbrace{51 - 1}_{(4)} + 40$

$\underbrace{50 + 40}_{(5)} = 90$

N.B : Lorsque les opérateurs ont la même priorité, on commence par le plus à gauche

Expression 2 : $a/b + ((d * c + 3)/5 * a) + 2 * c$

$4/2 + ((\underbrace{3 * 4}_{(1)} + 3)/5 * 4) + 2 * 4$ On commence par remplacer les variables par leurs valeurs

$4/2 + (\underbrace{(12 + 3)}_{(2)}/5 * 4) + 2 * 4$

$4/2 + (\underbrace{15/5}_{(3)} * 4) + 2 * 4$

$4/2 + \underbrace{(3 * 4)}_{(4)} + 2 * 4$

$\underbrace{4/2}_{(5)} + 12 + 2 * 4$

$2 + 12 + \underbrace{2 * 4}_{(6)}$

$\underbrace{2 + 12}_{(7)} + 8$

$\underbrace{14 + 8}_{(8)} = 22$

Expression 3 $\left(a > \underbrace{9\ \mathbf{DIV}\ 4}_{(1)}\right)$ $\mathbf{AND}$ $(a <> b)$ $\mathbf{OR\ NOT}$ $(c = b)$ avec $a = 2;\ b = 5;\ c = 5;$

$$\mathit{False}\ \mathbf{AND}\ \underbrace{(a <> b)}_{(2)}\ \mathbf{OR\ NOT}\ (c = b)$$

$$\mathit{False}\ \mathbf{AND}\ \mathit{True}\ \mathbf{OR\ NOT}\ \underbrace{(c = b)}_{(3)}$$

$$\mathit{False}\ \mathbf{AND}\ \mathit{True}\ \mathbf{OR}\ \underbrace{\mathbf{NOT}\ \mathit{True}}_{(4)}$$

$$\mathit{False}\ \mathbf{AND}\ \underbrace{\mathit{True}\ \mathbf{OR}\ \mathit{False}}_{(5)}$$

$$\underbrace{\mathit{False}\ \mathbf{AND}\ \mathit{True}}_{(6)} = \mathit{False}$$

<u>Exercice N°04 :</u>*(Type de variables)*

<u>Rappel :</u>

Lesvariablessontdesobjetscontenantdesvaleurspouvantêtesmodifiées.

Onacinqtypesdedonnéesdebase:

- Entiers / Integer
- Réels / Real
- Caractères / Char
- Chaînedecaractères / String
- Booléens / Boolean (True ou False);

Type des variables

Variable	Type
2024	Entier /Integer
10.05	Réel / Real
22.5e-8	Réel / Real
'A'	Caractère / Char
TRUE	Booléen /Boolean
'Hello World'	Chaîne de caractère /String
False	Booléen /Boolean

<u>Exercice N°05 :</u>*(Identificateurs)*

<u>Rappel :</u> Un identificateur est une chaîne de caractères contenant uniquement des caractère alpha-numériques (alphabétiques de [a-z] et [A-Z] et numérique[0-9]) et tiré 8'_'(trait souligné),et qui doit commencer soit par une lettre alphabétique ou.

Un identificateur permet d'identifier d'une manière unique un algorithme (ou un programme), une variable, une constante, une procédure ou une fonction.

Dans un langage de programmation donnée, on n'a pas le droit d'utiliser les mots réservés (mots clés) du langage comme des identificateurs.

Parmi les mots clés du langage Pascal: *program, begin, end, if, else, then, while, for, do, to, downto, repeat, until, goto, procedure, function, label, var, const, type, uses, array, of, real, integer, boolean, char, string,...*

Les identificateurs valides et non valides :

Identificateurs valides	**Identificateurs non valides**
A1	1A (L'identificateur commence par un chiffre)
T280	Begin (Mot-clé réservé en Pascal)
_exo	R?(L'identificateur contient un caractère spécial (?))
Algo	12R(L'identificateur commence par un chiffre)
Prix_HT	Prix-HT(L'identificateur contient un caractère spécial (tiret de 6 (-))
Exo_04	Exo 04(L'identificateur contient un espace)
Exem2ple	Exo-04(L'identificateur contient un caractère spécial (tiret de 6 (-)))
	Program(Mot-clé réservé en Pascal)