

TP INFORMATIQUE – SEMESTRE 2

SÉRIE DE TP N°03 (ENREGISTREMENTS & FICHIERS)

SOLUTION DES EXERCICES

SOMMAIRE

Série de TP N°03 (Enregistrements & Fichiers).....2

Exercice N°01 : Enregistrements..... 2

Solution..... 2

1) Traduction de l’algorithme en programme PASCAL.....2

2) Déroulement de l’algorithme.....3

3) Déduire ce que fait l’algorithme.....3

4) Réécriture du programme en utilisant WITH (Avec).....4

5) Réécriture l’algorithme et le programme pour calculer le produit $Z_3=Z_1*Z_2$4

Exercice N°02 : Fichiers..... 6

Solution..... 6

1) Traduction de l’algorithme, compilation & exécution.....6

2) Explication des procédures : Assigner, Réécrire, Écrire et Fermer.....7

3) L’emplacement du fichier fich_cercles.bin.....7

4) Modification du programme pour la lecture du fichier fich_cercles.bin.....8

TP INFORMATIQUE – SEMESTRE 2

SÉRIE DE TP N°03 (ENREGISTREMENTS & FICHIERS)

Exercice N°01 : Enregistrements

Soit l'algorithme suivant :

```

Algorithme Exercice_01;
Type
    Complexe = Enregistrement
        x, y : réel;
Fin;
Variables
    Z1, Z2, Z3 : Complexe;
Début
    //Entrées
    écrire ('Donner la valeur de Z1 : ');
    lire(Z1.x, Z1.y);

    écrire ('Donner la valeur de Z2 : ');
    lire(Z2.x, Z2.y);

    //Traitement
    Z3.x = Z1.x + Z2.x;
    Z3.y = Z1.y + Z2.y;

    //Sorties
    écrire ('Z3 = ', Z3.x, ' + ', Z3.y, ' i');
Fin.
    
```

Questions

- 1- Traduire l'algorithme en Programme PASCAL, puis compiler et exécuter le programme pour :
 $Z1 = 12 + 3i$ et $Z2 = -2.5 + 5.25i$, telque i est le nombre imaginaire ($i^2 = -1$).
- 2- Dérouler l'algorithme pour les valeurs de Z1 et Z2 ci-dessus ?
- 3- Dédire ce que fait l'algorithme ?
- 4- Ré-écrire le programme en utilisant la structure **WITH** pour Z3 (dans le traitement et les sorties).
- 5- Ré-écrire l'algorithme/Programme pour calculer le produit $Z1 * Z2$.

Solution

1) Traduction de l'algorithme en programme PASCAL

Algorithme	#	Programme PASCAL
Algorithme Exemple_01_Q1;	01	Program Exemple_01_Q1;
Type	02	Type
Complexe = <u>Enregistrement</u>	03	Complexe = <u>Record</u>
x, y : réel;	04	x, y : real;
Fin;	05	End;
Variables	06	Var
Z1, Z2, Z3 : Complexe;	07	Z1, Z2, Z3 : Complexe;
Début	08	Begin
//Entrées	09	//Entrées
écrire ('Donner la valeur de Z1 : ');	10	Write ('Donner la valeur de Z1 : ');
Lire(Z1.x, Z1.y);	11	Read(Z1.x, Z1.y);

écrire ('Donner la valeur de Z2 : '); Lire(Z2.x, Z2.y); //Traitement Z3.x ← Z1.x + Z2.x; Z3.y ← Z1.y + Z2.y; //Sorties écrire ('Z3 = ', Z3.x, ' + ', Z3.y, ' i '); Fin.	12 13 14 15 16 17 18 19 20 21 22	Write ('Donner la valeur de Z2: '); Read(Z2.x, Z2.y); //Traitement Z3.x:= Z1.x + Z2.x; Z3.y := Z1.y + Z2.y; //Sorties write ('Z3 = ', Z3.x:0:2, ' + ', Z3.y:0:2, ' i '); End.
---	--	---

Le lien du programme PASCAL : <https://onlinegdb.com/MYzgEgcqG>

Exemple d'exécution

Donne la valeur de Z1 : 12 3
Donne la valeur de Z2: -2.5 5.25
Z3 = 9.50 + 8.25 i

2) Déroulement de l'algorithme

Déroulement pour $Z1 = 12 + 3i$ et $Z2 = -2.5 + 5.25i$

Instructions	Variables						Affichage
	Z1		Z2		Z3		
	x	y	x	y	x	y	
écrire ('Donner la valeur de Z1 : ');	/	/	/	/	/	/	Donner la valeur de Z1 :
Lire(Z1.x, Z1.y);	12	3	/	/	/	/	
écrire ('Donner la valeur de Z2 : ');	12	3	/	/	/	/	Donner la valeur de Z2 :
Lire(Z2.x, Z2.y);	12	3	-2.5	5.25	/	/	
$Z3.x \leftarrow Z1.x + Z2.x = 12 + (-2.5) = 9.5$	12	3	-2.5	5.25	9.5	/	
$Z3.y \leftarrow Z1.y + Z2.y = 3 + 5.25 = 8.25$	12	3	-2.5	5.25	9.5	8.25	
écrire ('Z3 = ', Z3.x, ' + ', Z3.y, ' i ');	12	3	-2.5	5.25	9.5	8.25	Z3 = 9.5 + 8.25 i

3) Dédire ce que fait l'algorithme

D'après le déroulement ci-dessus, on déduit que l'algorithme permet :

- Introduire les valeurs de deux variables complexes Z1 et Z2 ;
- Calculer la valeur de Z3, tel-que Z3 est la somme des variables complexes Z1 et Z2
- Afficher la valeur de Z3

Donc, l'algorithme permet de réaliser la valeur Z3 qui représente la somme de deux valeurs complexes Z1 et Z2 :

- La partie réelle de Z3 est : la partie réelle de Z1 + la partie réelle de Z2
- La partie imaginaire de Z3 est : la partie imaginaire de Z1 + la partie imaginaire de Z2

4) Réécriture du programme en utilisant WITH (Avec)

#	Programme PASCAL
01	<u>Program</u> Exemple_01_Q4;
02	<u>Type</u>
03	Complexe = <u>Record</u>
04	x , y : real;
05	<u>End</u> ;
06	<u>Var</u>
07	Z1, Z2, Z3 : Complexe;
08	<u>Begin</u>
09	//Entrées
10	Write ('Donne la valeur de Z1 :');
11	Read(Z1.x, Z1.y);
12	
13	Write ('Donne la valeur de Z2:');
14	Read(Z2.x, Z2.y);
15	
16	//Traitement
17	<u>With</u> Z3 <u>do</u>
18	<u>begin</u>
19	x:= Z1.x + Z2.x;
20	y := Z1.y + Z2.y;
21	<u>end</u>
22	//Sorties
23	write ('Z3 = ', x:0:2, ' + ', y:0:2, ' i');
24	<u>end</u> ;
25	<u>End</u> .
26	
27	

5) Réécriture l'algorithme et le programme pour calculer le produit $Z_3=Z_1*Z_2$

Avant d'écrire l'algorithme/programme PASAL, nous devons écrire la formule mathématique du produit de deux variables complexes :

$$\begin{aligned} Z_1 \times Z_2 &= (x_1+y_1*i) \times (x_2+y_2 * i) = x_1*x_2 + x_1*y_2*i + x_2 * y_1* i + (y_1*y_2)* i^2 \text{ Avec } i^2 = -1 \\ &= x_1*x_2 + (x_1 * y_2)*i + (x_2 * y_1)* i - y_1*y_2 \\ &= (x_1*x_2 - y_1*y_2) + (x_1 * y_2 + x_2 * y_1) * i \end{aligned}$$

Donc : $Z_3.x = (x_1 \times x_2 - y_1 \times y_2)$ et $Z_3.y = (x_1 \times y_2 + x_2 \times y_1)$

Algorithmme	#	Programme PASCAL
Algorithmme Exemple_01_Q5; Type Complexe = Enregistrement x, y : réel; Fin; Variables Z1, Z2, Z3 : Complexe; Début //Entrées écrire ('Donner la valeur de Z1 : '); Lire(Z1.x, Z1.y); écrire ('Donner la valeur de Z2 : '); Lire(Z2.x, Z2.y); //Traitement $Z3.x \leftarrow Z1.x * Z2.x - Z1.y * Z2.y;$ $Z3.y \leftarrow Z1.x * Z2.y - Z1.y * Z2.x;$ //Sorties écrire ('Z3 = ', Z3.x, ' + ', Z3.y, ' i '); Fin.	01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22	Program Exemple_01_Q5; Type Complexe = Record x, y : real; End; Var Z1, Z2, Z3 : Complexe; Begin //Entrées Write ('Donner la valeur de Z1 : '); Read(Z1.x, Z1.y); Write ('Donner la valeur de Z2: '); Read(Z2.x, Z2.y); //Traitement $Z3.x := Z1.x * Z2.x - Z1.y * Z2.y;$ $Z3.y := Z1.x * Z2.y + Z1.y * Z2.x;$ //Sorties write ('Z3 = ', Z3.x:0:2, ' + ', Z3.y:0:2, ' i '); End.

Le lien du programme PASCAL : <https://onlinegdb.com/MFA9vITTRc>

Exemple d'exécution

Donne la valeur de Z1 : 2 5
Donne la valeur de Z2: 3 9
Z3 = -39.00 + 33.00 i

Exercice N°02 : Fichiers

Soit l'algorithme suivant :

```

Algorithme Exercice_02;
Type
  Cercle = Enregistrement
    x, y, Rayon : réel;
Fin;
Variables
  F : Fichier de Cercle ;
  C : Cercle;
Début
  assigner(F, 'fich_cercles.bin');
  Réécrire(F);

  C.x ← -5 ; C.y ← 8 ; C.Rayon ← 3.25 ;
  Écrire(F, C);
  C.x ← 3 ; C.y ← -5 ; C.Rayon ← 5.5 ;
  Écrire(F, C);

  Fermer(F);
  Écrire('Fin d'exécution du prgramme.');
```

Questions

- 1- Traduire l'algorithme en Programme PASCAL, puis compiler et exécuter le programme.
- 2- Expliquer les procédures : Assigner, Réécrire, Écrire, Fermer
- 3- Après l'exécution du programme, le fichier *fich_cercles.bin* est créé. Où se trouve-t-il ?
- 4- Modifier le programme pour lire le contenu du fichier et l'afficher sur écran.

Solution

1) Traduction de l'algorithme, compilation & exécution

Algorithme	#	Programme PASCAL
Algorithme Exemple_02_Q1;	01	Program Exemple_02_Q1;
Type	02	Type
Cercle = <u>Enregistrement</u>	03	Cercle = <u>Record</u>
x, y : réel;	04	x , y : real;
Rayon : réel ;	05	Rayon : real ;
Fin;	06	End;
Variables	07	Var
F : <u>Fichier de</u> Cercle;	08	F : <u>File of</u> Cercle;
C : Cercle ;	09	C : Cercle ;
Début	10	Begin
Assigner(F, 'fich_cercles.bin');	11	Assign(F, 'fich_cercles.bin');
Réécrire(F);	12	Rewrite(F);
	13	
C.x ← -5 ; C.y ← 8 ; C.Rayon ← 3.25 ;	14	C.x := -5 ; C.y := 8 ; C.Rayon := 3.25 ;
Écrire(F, C);	15	Write(F, C);
	16	
C.x ← 3 ; C.y ← -5 ; C.Rayon ← 5.5 ;	17	C.x := 3 ; C.y := -5 ; C.Rayon := 5.5 ;
Écrire(F, C);	18	Write(F, C);
	19	

Fermer(F) ; Écrire('Fin d"exécution du prgramme.');	20 21 22	Close(F) ; Write('Fin d"exécution du prgramme.');
Fin.		End.
Le lien du programme PASCAL : https://onlinegdb.com/EH9ybuL2S		
<u>Compilation et exécution du Programme</u>		
Fin d'exécution du programme.		

2) Explication des procédures : Assigner, Réécrire, Écrire et Fermer

Procédure	Explication
Assigner	Lier le fichier logique avec le fichier physique : lier la variable du fichier avec le chemin et le nom du fichier.
Réécrire	Ouvrir le fichier en mode écriture : Création de fichier. Si le fichier existe, il sera écrasé
Écrire (lignes : 15 et 18)	Permet d'enregistrer les valeurs de l'enregistrement C dans le fichier F
Écrire (ligne : 21)	Afficher le message 'Fin d"exécution du prgramme.' sur l'écran.
Fermer	Fermer le fichier F.

3) L'emplacement du fichier **fich_cercles.bin**

Après l'exécution du programme, le fichier '**fich_cercles.bin**' sera créé dans un emplacement d'une mémoire secondaire : Disque dur, flash-disk, ...

Pour connaître l'emplacement où le fichier sera créé, on revient à l'instruction Assigner. Par exemple :

- Assigner(F, '[D:/fich_tp.bin](#)'); : permet de créer le fichier **fich_tp.bin** dans le chemin (la partition) [D:/](#)
- Assigner(F, '[E:/tp/f3.data](#)'); : permet de créer le fichier **f3.data** dans le chemin [E:/TP/](#)
(La partition [E:/](#) dans le dossier TP).

Dans le programme de l'exercice N°02 (à la ligne 11) : Assigner(F, '**fich_cerclce.bin**'); Dans, cette instruction, on n'a pas indiqué le chemin complet du fichier (On a indiqué juste le nom du fichier). Dans ce cas, on appelle ça le chemin relatif, et *le chemin du fichier sera le même chemin du programme exécuté. C'est-à-dire, le fichier '**fich_cerclce.bin**' sera dans le dossier où le programme PASCAL est enregistré.*

4) Modification du programme pour la lecture du fichier `fich_cercles.bin`

Pour lire le contenu fichier '`fich_cerclce.bin`', on utilise l'instruction Réouvrir (Reset) au lieu de Réécrire (Rewrite), et on utilise une boucle avec la fonction booléenne : FDF (EOF) (FDF : Fin de fichier / EOF : End Of File). Comme indiqué ci-dessous :

<u>Algorithme</u>	#	<u>Programme PASCAL</u>
<u>Algorithme</u> Exemple_02_Q4;	01	<u>Program</u> Exemple_02_Q4;
<u>Type</u>	02	<u>Type</u>
Cercle = <u>Enregistrement</u>	03	Cercle = <u>Record</u>
x, y : réel;	04	x , y : real;
Rayon : réel ;	05	Rayon : real ;
<u>Fin;</u>	06	<u>End;</u>
<u>Variables</u>	07	<u>Var</u>
F : <u>Fichier de</u> Cercle;	08	F : <u>File of</u> Cercle;
C : Cercle ;	09	C : Cercle ;
<u>Début</u>	10	<u>Begin</u>
Assigner(F, ' <code>fich_cercles.bin</code> ');	11	Assign(F, ' <code>fich_cercles.bin</code> ');
Réouvrir(F);	12	Reset(F);
	13	
<u>Tant-que (Non(FDF(f))) faire</u>	14	<u>While (Not(EOF(f))) do Begin</u>
Lire(F, c) ;	15	Read(F, c) ;
Écrire(c.x, c.y, c.rayon);	16	Writeln('x=', c.x:0:2, ' y=',c.y:0:2, ' et rayon=',c.rayon:0:2);
<u>Fin-Tant-que;</u>	17	<u>End;</u>
	18	
Fermer(F) ;	19	Close(F) ;
Écrire('Fin d"exécution du prgramme.');	20	Write('Fin d"exécution du prgramme.');
<u>Fin.</u>	21	<u>End.</u>

Le lien du programme PASCAL : <https://onlinegdb.com/AI22z16mo>

Compilation et exécution du Programme

x=-5.00 y=8.00 et rayon=3.25
x=3.00 y=-5.00 et rayon=5.50

Fin d'exécution du prgramme.