

UF 03 : Exercices et solutions

<i>Énoncés</i>	<i>1</i>
<i>Solutions</i>	<i>23</i>
Table des énoncés	54
Table des solutions	54

Énoncés

Énoncé 1

Lancer l'éditeur, enregistrer le code source de l'exemple, compiler et exécuter le programme.

Énoncé 2

Quels sont les identificateurs corrects ou incorrects ?

```
A
A1
1b
n
n_1
racine_carree
racine_carrée
Racine-carree
Taxe_sur_la_valeur_ajoutée
Nombre 1
```

Énoncé 3

Reprendre l'exemple complet vu au cours (premier exercice) et à partir de cet exercice :

- o Voir comment réagit le programme si le type d'une variable n'est pas déclaré (supprimez la ligne "nombre : real ; ")
- o Voir ce qui se passe lorsqu'on tente de donner un mot réservé comme nom de variable
- o Vérifier qu'il est possible de donner comme nom de variable un identificateur prédéfini.

Énoncé 4

Ecrire un programme qui affiche tous les caractères dont le code ascii est compris entre deux valeurs fournies par l'utilisateur.

```
Donnez un premier nombre : 50
Donnez un second nombre : 60
Voici les caractères allant de 50 à 60 :
2
3
4
5
6
7
8
9
0
:
;
<
```

Énoncé 5

Écrire un programme qui affiche la partie entière d'un nombre, ainsi que l'entier le plus proche. Le programme devra se dérouler de cette manière :

```
Donnez un nombre : 5.63
Entier le plus proche : 6
Partie entière du nombre : 5
Merci
```

Énoncé 6

Écrire un programme qui demande un caractère à l'utilisateur. Le programme affichera la valeur ASCII correspondante. Le programme demandera ensuite à l'utilisateur d'entrer un nombre (entre 0 et 255). Le programme affichera le caractère correspondant à ce nombre. Exemple d'exécution du programme :

```
Donnez un caractère : Z
Valeur ASCII : 90
Donnez un nombre (0 à 255) : 123
Caractère correspondant : {
```

Énoncé 7

Réécrire le programme ci-dessous avec comme objectif de comparer deux caractères introduits par l'utilisateur et non pas deux entiers.

```
program Exemple_booleen ;
var n, p : integer ;
    condition : boolean ;
begin
    writeln('donnez 2 entiers :') ;
    readln(n) ;
    readln(p) ;
    condition := n < p ;
    writeln('Resultat : ', condition) ;
    if condition then
        writeln('croissant !')
    else
        writeln('non croissant !')
end.
```

Énoncé 8

Quelles erreurs ont été éventuellement commises dans les expressions suivantes :

1. var x = integer
2. var c : real, d : integer ;
3. var nombre : char ;
4. lettre : real ;
5. var x, y, z,
 t, u,
 v : integer ; z : real ;
6. var z = 3 ;
7. const n : 8 ;
8. const i = 'a', z = 2 ;

```
9. const un_demi = .5 ;
```

Énoncé 9

Écrire un programme qui demande à l'utilisateur d'introduire un nombre. Le programme indiquera si le nombre introduit est à la fois plus grand que 10 et plus petit que 20. Voici deux exemples d'exécution du programme.

```
Donnez un nombre : 12
Le nombre 12 est plus grand que 10 ET plus petit que 20.
```

Énoncé 10

Écrire un programme qui demande à l'utilisateur d'introduire un nombre. Le programme indiquera si le nombre introduit est plus petit que 10 ou plus grand que 20. Voici deux exemples d'exécution du programme.

```
Donnez un nombre: 8
Le nombre 8 est soit plus petit que 10, soit plus grand que 20.
```

Énoncé 11

Écrire un programme qui demande à l'utilisateur d'introduire un nombre. Le programme indiquera si le nombre introduit plus petit que 20 OU (exclusif) plus grand que 10. Voici deux exemples d'exécution du programme.

```
Donnez un nombre : 12
Le nombre 12 est plus petit que 20 ET plus grand que 10
```

Énoncé 12

Les expressions suivantes sont-elles correctes ? Si oui, donnez la valeur et le type :

- a. `sqr(4)`
- b. `sqr(4.0)`
- c. `sqr(4)`
- d. `sqr(4.0)`

Même question avec la variable réelle x valant 10.6 :

- a. `trunc(x) + 1`
- b. `trunc(x) + 1.0`
- c. `succ (trunc(x))`
- d. `trunc (succ(x))`

Même exercice en y traitant les variables entières n=12, p=5 et q=3 :

- a. `n mod p + q`
- b. `n mod p div q`
- c. `n = p or n = q`
- d. `succ (n = p)`

Idem avec la variable c, qui est du type char et qui vaut e :

- a. `succ (c+1)`
- b. `ord@ + 2.5`
- c. `chr (ord (c))`
- d. `chr (succ (ord (c)))`

n est une variable entière, x est une variable réelle. Les expressions suivantes sont-elles correctes ?

- a. $x := n$
- b. $n := x + 1$
- c. $n := \text{trunc}(x) + 1$
- d. $n := \text{round}(x) + 1.5$
- e. $n := \text{round}(x + 1.5)$

Énoncé 13

Ecrire un programme permettant d'échanger les valeurs de deux variables réelles x et y.

```
Premier nombre : 12
Second nombre : 15
A=12.22
B=15.91
ECHANGE
A=15.91
B=12.22
```

Énoncé 14

Écrire un programme calculant la surface d'un cercle, dont le rayon est fourni par l'utilisateur (NB : la constante PI existe en PASCAL).

Donnez le rayon du cercle : 5.4

La surface d'un cercle de 5.40 de rayon = 91.61

La circonférence d'un cercle de 5.40 de rayon = 33.93

Énoncé 15

Écrire un programme demandant à l'utilisateur d'introduire des nombres et un caractère et se déroulant de la manière suivante :

```
Donnez un nombre entier : 12
Voici votre nombre
12
12
12
12
12
Donnez un nombre réel : 16.36
Merci pour : 16.3600
Donnez un caractère : A
Merci pour : A A A A A A A A A A
```

Énoncé 16

Écrire un programme qui affiche des nombres entiers de la manière suivante :

```
Premier nombre : 8
Second nombre : 15
Voici les entiers de 8 à 15
8
```

9
10
11
12
13
14
15

Énoncé 17

Écrire un programme affichant les 20 premiers entiers de cette manière :

```
Voici les 20 premiers entiers
 1  2  3  4  5  6  7  8  9 10
11 12 13 14 15 16 17 18 19 20
```

Énoncé 18x

Utiliser les fonctions **pred()** et **succ()** dans un programme devant réaliser les opérations suivantes :

```
Donnez un nombre entier : 134
Nombre précédent       : 133
Nombre suivant        : 135

Donnez un caractère    : Z
Caractère précédent    : Y
Caractère suivant     : [
```

Énoncé 19

Utilisez l'instruction **if** pour écrire un programme qui calcule un montant TTC et qui se déroule de la manière suivante :

```
Introduisez le montant HORS TAXE : 100
Donnez une lettre correspondant au taux à appliquer
(A = 7 %, B = 12.32 %, C = 18.6 %, D = 33.33 %)
Votre choix : C
-----
Prix TTC : 118.6
```

Si l'utilisateur introduit une autre lettre que les quatre prévues, le programme devra afficher un message spécifique. Par exemple :

```
Votre choix : e
-----
LETTRE ERRONEE
```

Énoncé 20

Réécrire l'exercice 19, mais cette fois en utilisant l'instruction **case... of** à la place des instructions **if**. Si la lettre introduite n'est pas correcte, le programme utilisera le taux 'C' par défaut et affichera ***** code incorrect *** Taux de 21.5% est utilisé.**

Énoncé 21

Écrire un programme qui demande à l'utilisateur d'introduire un nombre d'articles et un prix unitaire. Le programme calcule ensuite le montant TTC (prix unitaire * quantité + TVA (21,5%). Si le montant TTC est supérieur à 1000, une remise de 5% est offerte.

```
Combien d'articles : 27
Prix unitaire      : 176.85
Montant TTC       : 5801.56
Remise            : 290.08
Net a payer       : 5511.49
```

Énoncé 22

Écrire un programme qui demande à l'utilisateur d'introduire un nombre d'articles et un prix unitaire. Le programme devra calculer le montant HTVA et le montant TTC (la TVA étant de 21.5%). Une remise sera attribuée en fonction du nombre d'articles vendus :

- Jusqu'à 25 articles vendus, aucune remise n'est attribuée ;
- Pour une quantité allant de 26 à 50, 5% de remise sont accordés.
- Pour une quantité allant de 51 à 100, 10% sont accordés ;
- Pour une quantité dépassant 100, 15% sont accordés.

Le programme s'exécute de la manière suivante :

```
Combien d'articles : 110
Prix unitaire      : 15.15
Montant HTVA      : 1666.50
Montant TTC       : 2024.80
Remise            : 303.72
Net a payer       : 1721.08
```

Remarque : la remise est accordée sur l'ensemble des articles vendus.

Énoncé 23

En utilisant l'instruction **case...of**, écrire un programme qui demande à l'utilisateur d'introduire un nombre. Le programme indiquera ensuite si le nombre est pair ou impair. Le programme se déroulera de la manière suivante (un exemple pour chaque cas de figure est repris) :

```
Donnez un nombre : 12
Le nombre 12 est pair.
-----

Donnez un nombre : 45
Le nombre 45 est impair.
-----
```

Énoncé 24

Écrire le code source correspondant à l'exécution suivante :

```
Donnez le nombre a diviser      : 12
Donnez le diviseur (2, 3 ou 5) : 5
-----
12 n'est pas divisible par 5
Le reste de la division vaut 2
```

Énoncé 25

Ecrire un programme qui calcule la moyenne de 5 nombres introduit par l'utilisateur. Le dialogue se passera comme ci-dessous :

```
Entrez le nombre 1 : 14
Entrez le nombre 2 : 23
Entrez le nombre 3 : 123
Entrez le nombre 4 : 90
Entrez le nombre 5 : 6
-----
Moyenne des 5 nombres : 51.20
```

Énoncé 26

Ecrire un programme qui calcule le carré des nombres de 1 à 10

```
1 au carré : 1
2 au carré : 4
3 au carré : 9
4 au carré : 16
5 au carré : 25
6 au carré : 36
7 au carré : 49
8 au carré : 64
9 au carré : 81
10 au carré : 100
```

Énoncé 27

Ecrire un programme se déroule de cette manière :

```
Combien de nombres voulez-vous introduire : 10
Nombre 1 : 12.2
Nombre 2 : 2
Nombre 3 : 6
Nombre 4 : 45.67
Nombre 5 : 3
Nombre 6 : 12
Nombre 7 : 3
Nombre 8 : 5.89
Nombre 9 : 87.5
Nombre 10 : 4
-----
Somme des 10 nombres : 181.26
```

Énoncé 28

Ecrire un programme qui calcule la moyenne de n nombres, n étant introduit par l'utilisateur :

```
Combien de nombres voulez-vous introduire : 4
Nombre 1 : 12.4
Nombre 2 : 34.5
Nombre 3 : 5.7
Nombre 4 : 6.9
-----
Moyenne des 4 nombres : 14.88
```

Énoncé 29

Ecrire un programme qui se déroulera de la manière suivante :

```
Combien d'articles différents vendus : 3

Quantité vendue article 1 : 145
Prix de vente article 1 : 67.13
TOTAL HTVA ARTICLE 1 : 9733.85

Quantité vendue article 2 : 220
Prix de vente article 2 : 35.23
TOTAL HTVA ARTICLE 2 : 7750.60

Quantité vendue article 3 : 255
Prix de vente article 3 : 78.89
TOTAL HTVA ARTICLE 3 : 20116.95

-----
Total des ventes HTVA : 37601.40
TVA (21%) : 7896.29
MONTANT TOTAL DES VENTES : 45497.69
```

Énoncé 30

Écrire un programme qui se déroule de la manière suivante :

```
Donnez un nombre compris entre 1 et 100 : -12
Donnez un nombre compris entre 1 et 100 : 200
Donnez un nombre compris entre 1 et 100 : 101
Donnez un nombre compris entre 1 et 100 : 23

Merci pour 23
```

Énoncé 31

Écrire le code source correspondant à l'exécution suivante :

```
Donnez le nombre a diviser      : 123
Donnez le diviseur (2, 3 ou 5) : -5
Donnez le diviseur (2, 3 ou 5) : 4
Donnez le diviseur (2, 3 ou 5) : 3
-----
123 est divisible par 3
```

Énoncé 32

Écrire le programme suivant en utilisant les boucles **for**, **repeat...until** et **while**. Les trois programmes devront arriver au même résultat présenté ci-dessous.

```
Donnez un nombre compris entre 1 et 15 : -12
Donnez un nombre compris entre 1 et 15 : 35
Donnez un nombre compris entre 1 et 15 : 10

1 a pour carré 1
2 a pour carré 4
3 a pour carré 9
4 a pour carré 16
5 a pour carré 25
6 a pour carré 36
7 a pour carré 49
8 a pour carré 64
9 a pour carré 81
10 a pour carré 100
```

Énoncé 33

Écrire un programme qui se déroule de la manière suivante. Rien de bien nouveau par rapport ce qui a été vu, si ce n'est que les données sont validées systématiquement au fur et à mesure qu'elles sont introduites.

```
Combien d'articles différents vendus (positif) : -12
Combien d'articles différents vendus (positif) : 3

Quantité vendue article 1 (entre 1 et 100) : 0
Quantité vendue article 1 (entre 1 et 100) : 150
Quantité vendue article 1 (entre 1 et 100) : 50
Prix de vente article 1 (positif !) : -12.2
Prix de vente article 1 (positif !) : 12.5
TOTAL HTVA ARTICLE 1 : 625.00

Quantité vendue article 2 (entre 1 et 100) : 75
Prix de vente article 2 (positif !) : 45.56
TOTAL HTVA ARTICLE 2 : 3417.00

Quantité vendue article 3 (entre 1 et 100) : 200
Quantité vendue article 3 (entre 1 et 100) : 80
Prix de vente article 3 (positif !) : 45.23
TOTAL HTVA ARTICLE 3 : 3618.40
```

```
CHOISISSEZ LE TAUX DE T.V.A. (A="21%", B="15%", C="12%")
Votre choix : D
Votre choix : E
Votre choix : A
-----
Total des ventes HTVA      : 7660.40
TVA (21%)                  : 1608.68
MONTANT TOTAL DES VENTES  : 9269.08
-----
```

Énoncé 34

Écrire le programme correspondant à l'exécution suivante :

```
Donnez un nombre (entre 1 et 15): -4
-----
Donnez un nombre (entre 1 et 15): 20
-----
Donnez un nombre (entre 1 et 15): 6
-----
+      +
+      +
+  +
+
+  +
+  +
+      +
+      +
```

Énoncé 35

Écrire le programme correspondant à l'exécution suivante. Le programme demande à l'utilisateur d'introduire des nombres afin de calculer une moyenne. La saisie des notes s'arrête lorsque le nombre introduit est 0. Exemple d'exécution :

```
Donnez la note 1 (0 pour finir) : 12.25
Donnez la note 2 (0 pour finir) : 15
Donnez la note 3 (0 pour finir) : 11.75
Donnez la note 4 (0 pour finir) : 18.25
Donnez la note 5 (0 pour finir) : 9
Donnez la note 6 (0 pour finir) : 0
```

La moyenne des 5 notes = 13.25

Énoncé 36

Idem exercice précédent, mais cette fois, le programme empêchera qu'une note inférieure à 0 soit saisie. Si c'est le cas, le programme redemande d'introduire cette même note.

```
Donnez la note 1 (0 pour finir) : 12.25
Donnez la note 2 (0 pour finir) : -15
Donnez la note 2 (0 pour finir) : 15
Donnez la note 3 (0 pour finir) : 11.75
Donnez la note 4 (0 pour finir) : -18.25
Donnez la note 4 (0 pour finir) : 18.25
Donnez la note 5 (0 pour finir) : 9
Donnez la note 6 (0 pour finir) : 0
```

La moyenne des 5 notes = 13.25

Énoncé 37

Écrire un programme qui indique si un nombre est premier ou non. Pour rappel, un nombre est dit premier s'il n'est divisible que par un et par lui-même (1 n'est pas premier, ni les nombres négatifs). Le programme se déroulera de la manière suivante :

Premier exemple :

```
Donnez un nombre positif : -29
Donnez un nombre positif : 29
29 est un nombre premier.
```

Second exemple :

```
Donnez un nombre positif : 24
24 est divisible par 2
```

Il existe l'instruction `break` en Turbo Pascal que vous pouvez utiliser ici, le cas échéant. Cette instruction permet de sortir d'une boucle.

Énoncé 38

Idem exercice précédent mais cette fois, le programme répétera tout le traitement, et ce tant que le nombre introduit est différent de 0.

```
Donnez un nombre positif : -5
Donnez un nombre positif : 12
12 n'est pas un nombre premier.
Donnez un nombre positif : 11
11 est un nombre premier.
Donnez un nombre positif : -15
Donnez un nombre positif : 15
15 n'est pas un nombre premier.
Donnez un nombre positif : 0
Travail terminé !
```

Énoncé 39

Écrire un programme permettant de savoir combien de possibilités différentes il y a pour obtenir 1 euro avec des pièces de 2, de 5 et 10 cents. Le programme devra afficher :

```
1 : 1 Euro = 50 x 2cents
2 : 1 Euro = 45 x 2cents 2 x 5cents
3 : 1 Euro = 40 x 2cents 4 x 5cents
4 : 1 Euro = 35 x 2cents 6 x 5cents
5 : 1 Euro = 30 x 2cents 8 x 5cents
6 : 1 Euro = 25 x 2cents 10 x 5cents
7 : 1 Euro = 20 x 2cents 12 x 5cents
8 : 1 Euro = 15 x 2cents 14 x 5cents
9 : 1 Euro = 10 x 2cents 16 x 5cents
10 : 1 Euro = 5 x 2cents 18 x 5cents
11 : 1 Euro = 20 x 5cents
12 : 1 Euro = 45 x 2cents 1 x 10cents
13 : 1 Euro = 40 x 2cents 2 x 5cents 1 x 10cents
14 : 1 Euro = 35 x 2cents 4 x 5cents 1 x 10cents
15 : 1 Euro = 30 x 2cents 6 x 5cents 1 x 10cents
16 : 1 Euro = 25 x 2cents 8 x 5cents 1 x 10cents
17 : 1 Euro = 20 x 2cents 10 x 5cents 1 x 10cents
```

```

18 : 1 Euro = 15 x 2cents 12 x 5cents 1 x 10cents
19 : 1 Euro = 10 x 2cents 14 x 5cents 1 x 10cents
20 : 1 Euro = 5 x 2cents 16 x 5cents 1 x 10cents
21 : 1 Euro = 18 x 5cents 1 x 10cents
22 : 1 Euro = 40 x 2cents 2 x 10cents
23 : 1 Euro = 35 x 2cents 2 x 5cents 2 x 10cents
24 : 1 Euro = 30 x 2cents 4 x 5cents 2 x 10cents
25 : 1 Euro = 25 x 2cents 6 x 5cents 2 x 10cents
26 : 1 Euro = 20 x 2cents 8 x 5cents 2 x 10cents
27 : 1 Euro = 15 x 2cents 10 x 5cents 2 x 10cents
28 : 1 Euro = 10 x 2cents 12 x 5cents 2 x 10cents
29 : 1 Euro = 5 x 2cents 14 x 5cents 2 x 10cents
30 : 1 Euro = 16 x 5cents 2 x 10cents
31 : 1 Euro = 35 x 2cents 3 x 10cents
32 : 1 Euro = 30 x 2cents 2 x 5cents 3 x 10cents
33 : 1 Euro = 25 x 2cents 4 x 5cents 3 x 10cents
34 : 1 Euro = 20 x 2cents 6 x 5cents 3 x 10cents
35 : 1 Euro = 15 x 2cents 8 x 5cents 3 x 10cents
36 : 1 Euro = 10 x 2cents 10 x 5cents 3 x 10cents
37 : 1 Euro = 5 x 2cents 12 x 5cents 3 x 10cents
38 : 1 Euro = 14 x 5cents 3 x 10cents
39 : 1 Euro = 30 x 2cents 4 x 10cents
40 : 1 Euro = 25 x 2cents 2 x 5cents 4 x 10cents
41 : 1 Euro = 20 x 2cents 4 x 5cents 4 x 10cents
42 : 1 Euro = 15 x 2cents 6 x 5cents 4 x 10cents
43 : 1 Euro = 10 x 2cents 8 x 5cents 4 x 10cents
44 : 1 Euro = 5 x 2cents 10 x 5cents 4 x 10cents
45 : 1 Euro = 12 x 5cents 4 x 10cents
46 : 1 Euro = 25 x 2cents 5 x 10cents
47 : 1 Euro = 20 x 2cents 2 x 5cents 5 x 10cents
48 : 1 Euro = 15 x 2cents 4 x 5cents 5 x 10cents
49 : 1 Euro = 10 x 2cents 6 x 5cents 5 x 10cents
50 : 1 Euro = 5 x 2cents 8 x 5cents 5 x 10cents
51 : 1 Euro = 10 x 5cents 5 x 10cents
52 : 1 Euro = 20 x 2cents 6 x 10cents
53 : 1 Euro = 15 x 2cents 2 x 5cents 6 x 10cents
54 : 1 Euro = 10 x 2cents 4 x 5cents 6 x 10cents
55 : 1 Euro = 5 x 2cents 6 x 5cents 6 x 10cents
56 : 1 Euro = 8 x 5cents 6 x 10cents
57 : 1 Euro = 15 x 2cents 7 x 10cents
58 : 1 Euro = 10 x 2cents 2 x 5cents 7 x 10cents
59 : 1 Euro = 5 x 2cents 4 x 5cents 7 x 10cents
60 : 1 Euro = 6 x 5cents 7 x 10cents
61 : 1 Euro = 10 x 2cents 8 x 10cents
62 : 1 Euro = 5 x 2cents 2 x 5cents 8 x 10cents
63 : 1 Euro = 4 x 5cents 8 x 10cents
64 : 1 Euro = 5 x 2cents 9 x 10cents
65 : 1 Euro = 2 x 5cents 9 x 10cents
66 : 1 Euro = 10 x 10cents

```

Il y a 66 possibilités différentes.

La méthode la plus facile pour résoudre cet exercice est de commencer avec 2 pièces (5 et 10 cents par exemple) et d'ajouter les autres une fois que le principe est compris.

Énoncé 40

Même principe que l'exercice 37 mais cette fois, le programme redemande à l'utilisateur d'introduire un nombre et le programme va vérifier si ce nombre est premier ou non.

L'opération sera répétée tant que le nombre introduit est différent de 1. Lorsque le nombre introduit est 1, le programme s'arrête.

Exemple d'exécution

```
Donnez un nombre positif (1 pour terminer) : 24
24 n'est pas un nombre premier, il est divisible par 2
Donnez un nombre positif (1 pour terminer) : 9
9 n'est pas un nombre premier, il est divisible par 3
Donnez un nombre positif (1 pour terminer) : -23
Donnez un nombre positif (1 pour terminer) : 1
Travail terminé
```

Énoncé 41

Écrire un programme qui affiche une croix de la manière indiquée ci-après. Le nombre est introduit par l'utilisateur. Si ce nombre est pair et positif, il sera incrémenter de 1. Si le nombre est pair et négatif, il sera décrémenter de 1 (par exemple, 6 → 7 ; -6 → -7). Ceci permettra d'accepter les nombres négatifs.

```
Donnez un nombre différent de 0 : 0
Donnez un nombre différent de 0 : -6
Voici une croix de dimension 7
-----
      +
      +
      +
+++++++
      +
      +
      +
    MERCI
```

Énoncé 42

Écrire un programme se déroulant de la manière suivante :

```
NOMBRE DE LIGNES (entre 1 et 15): -12
NOMBRE DE LIGNES (entre 1 et 15): 20
NOMBRE DE LIGNES (entre 1 et 15): 10
-----
1| *****
2| *           *
3| *           *
4| *           *
5| *           *
6| *           *
7| *           *
8| *           *
9| *           *
10| *****
```