

CORRECTION ALGO

--ex-1--

VARIABLES

i EST_DU_TYPE NOMBRE

DEBUT_ALGORITHME

POUR i ALLANT_DE 1 A 25

DEBUT_POUR

AFFICHER i + " "

FIN_POUR

FIN_ALGORITHME

--ex-2--

VARIABLES

i EST_DU_TYPE NOMBRE

DEBUT_ALGORITHME

i PREND_LA_VALEUR 25

TANT_QUE (i > 0) FAIRE

DEBUT_TANT_QUE

AFFICHER* i

i PREND_LA_VALEUR i - 1

FIN_TANT_QUE

FIN_ALGORITHME

--ex-3--

VARIABLES

i EST_DU_TYPE NOMBRE

y EST_DU_TYPE NOMBRE

t EST_DU_TYPE NOMBRE

DEBUT_ALGORITHME

i PREND_LA_VALEUR 10

y PREND_LA_VALEUR 5

AFFICHER* "i vaut "+i

AFFICHER* "y vaut "+y

t PREND_LA_VALEUR i

i PREND_LA_VALEUR y

y PREND_LA_VALEUR t

AFFICHER* "i vaut "+i

AFFICHER* "y vaut "+y

FIN_ALGORITHME

Réponse alternative avec 2 variables

VARIABLES

```
i EST_DU_TYPE NOMBRE
y EST_DU_TYPE NOMBRE
t EST_DU_TYPE NOMBRE
```

DEBUT_ALGORITHME

```
i PREND_LA_VALEUR 10
y PREND_LA_VALEUR 5
```

```
AFFICHER* "i vaut "+i
AFFICHER* "y vaut "+y
```

```
i PREND_LA_VALEUR i+y
y PREND_LA_VALEUR i-y
i PREND_LA_VALEUR i-y
```

```
AFFICHER* "i vaut "+i
AFFICHER* "y vaut "+y
```

FIN_ALGORITHME

--ex-4--

VARIABLES

```
i EST_DU_TYPE NOMBRE
y EST_DU_TYPE NOMBRE
```

DEBUT_ALGORITHME

```
POUR i ALLANT_DE 1 A 25
  DEBUT_POUR
  AFFICHER* " "
  POUR y ALLANT_DE 1 A i
    DEBUT_POUR
    AFFICHER y + " "
    FIN_POUR
  FIN_POUR
```

FIN_ALGORITHME

--ex-5--

VARIABLES

i EST_DU_TYPE NOMBRE
y EST_DU_TYPE NOMBRE
n EST_DU_TYPE NOMBRE

DEBUT_ALGORITHME

LIRE n
y PREND_LA_VALEUR 0
POUR i ALLANT_DE 1 A n
 DEBUT_POUR
 y PREND_LA_VALEUR i + y
 FIN_POUR
AFFICHER y

FIN_ALGORITHME

--ex-6--

VARIABLES

n EST_DU_TYPE NOMBRE

DEBUT_ALGORITHME

LIRE n
SI (n%2==0) ALORS
 DEBUT_SI
 AFFICHER n+" Est pair"
 FIN_SI
SINON
 DEBUT_SINON
 AFFICHER n+" Est impair"
 FIN_SINON

FIN_ALGORITHME

--ex-7--

VARIABLES

i EST_DU_TYPE NOMBRE

DEBUT_ALGORITHME

POUR i ALLANT_DE 1 A 20
 DEBUT_POUR
 SI (i%2==0) ALORS
 DEBUT_SI
 AFFICHER* i+" Est pair"
 FIN_SI
 SINON
 DEBUT_SINON
 AFFICHER* i+" Est impair"
 FIN_SINON
 FIN_POUR

FIN_ALGORITHME

--ex-8--

VARIABLES

a EST_DU_TYPE NOMBRE
b EST_DU_TYPE NOMBRE
c EST_DU_TYPE NOMBRE

DEBUT_ALGORITHME

LIRE b
LIRE c
a PREND_LA_VALEUR $c*c + b*b$
a PREND_LA_VALEUR $\text{sqrt}(a)$
AFFICHER a

FIN_ALGORITHME

--ex-9--

VARIABLES

heure EST_DU_TYPE NOMBRE

DEBUT_ALGORITHME

LIRE heure
SI (heure>6 **ET** heure<12) **ALORS**
 DEBUT_SI
 AFFICHER "C'est le matin"
 FIN_SI
SI (heure>=12 **ET** heure <18) **ALORS**
 DEBUT_SI
 AFFICHER "C'est l'après midi"
 FIN_SI
SI (heure>=18 **ET** heure <22) **ALORS**
 DEBUT_SI
 AFFICHER "C'est le soir"
 FIN_SI
SI ((heure>=22 **ET** heure <= 24) **OU** (heure >0 **ET** heure<= 6)) **ALORS**
 DEBUT_SI
 AFFICHER "C'est la nuit"
 FIN_SI
SI ((heure>=0 **ET** heure <= 24)) **ALORS**
 DEBUT_SI
 AFFICHER ""
 FIN_SI
 SINON
 DEBUT_SINON
 AFFICHER "Veuillez rentrer une heure valide"
 FIN_SINON

FIN_ALGORITHME

--ex-10--

```
VARIABLES
    i EST_DU_TYPE NOMBRE
    y EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
    POUR i ALLANT_DE 1 A 100
        DEBUT_POUR
            SI (i%3==0 ET i%5==0) ALORS
                DEBUT_SI
                    AFFICHER* "foobar"
                FIN_SI
            SINON
                DEBUT_SINON
                    SI (i%3==0) ALORS
                        DEBUT_SI
                            AFFICHER* "foo"
                        FIN_SI
                    SINON
                        DEBUT_SINON
                            SI (i%5==0) ALORS
                                DEBUT_SI
                                    AFFICHER* "bar"
                                FIN_SI
                            SINON
                                DEBUT_SINON
                                    AFFICHER* i
                                FIN_SINON
                            FIN_SINON
                        FIN_SINON
                    FIN_SINON
                FIN_SINON
            FIN_POUR
        FIN_ALGORITHME    FIN_SI
```

--Alternative--

```
VARIABLES
    i EST_DU_TYPE NOMBRE
    temp EST_DU_TYPE CHAINE
DEBUT_ALGORITHME
    POUR i ALLANT_DE 1 A 100
        DEBUT_POUR
            temp PREND_LA_VALEUR ""
```

```

    SI (i % 3 == 0)    ALORS
        DEBUT_SI
            temp PREND_LA_VALEUR "foo"
        FIN_SI
    SI (i % 5 == 0)    ALORS
        DEBUT_SI
            temp PREND_LA_VALEUR temp+"bar"
        FIN_SI
    SI (temp != "")    ALORS
        DEBUT_SI
            AFFICHER* temp
        FIN_SI
    SINON
        DEBUT_SINON
            AFFICHER* i
        FIN_SINON
    FIN_POUR

```

FIN_ALGORITHME

--ex-11--

VARIABLES

i EST_DU_TYPE NOMBRE

y EST_DU_TYPE NOMBRE

DEBUT_ALGORITHME

POUR i ALLANT_DE 0 A 9

DEBUT_POUR

AFFICHER* ""

POUR y ALLANT_DE 0 A 9

DEBUT_POUR

AFFICHER (y * i) + "\t"

FIN_POUR

FIN_POUR

FIN_ALGORITHME