

15: On considère l'algorithme suivant :

```
Algorithme test;
Var i, j, s=0;
Début
Pour i←0 à 5 faire
  j←0;
  Tantque(j<4)
    s←s+1;
    j←j+1;
  FinTQ
FinPour
Ecrire (i, j, s);
Fin
```

L'exécution de l'algorithme affiche :

6 3 24
 4 5 20
 6 4 24
 5 3 24

CORRECTION

16: On considère l'algorithme suivant :

```
Algorithme sommeCF ;
Var n, m, Som: Entier;
Début
Ecrire ("saisir un entier positif");
Lire(n) ;
Som ← 0 ;
Tantque (n>0) faire
  m ← n MOD 10 ;
  Som ← Som+ m;
  n ← n DIV 10;
  Si (n=0) alors
    Ecrire(Som);
  Finsi
FinTQ
Fin
```

A L'exécution, l'algorithme affiche pour n=82:

10 1
 28
 1 10
 10

CORRECTION

17: Une variable en informatique représente:

- une case de la memoire
 aucune de ces réponses
 une donnée
 un équivalent d'une variable mathématique

18: Donner le résultat de l'algorithme suivant pour le couple (N = 4, M = 12):

```
Algorithme test
Variables i, N, M, D: entiers
Début
  Ecrire("Entrer deux entiers:");
  Lire(N,M);
  i ← 1;
  Tantque(i<=N et i<=M) faire
    si(N mod i=0 et M mod i=0) alors
      D ← i
    FinSi
    i ← i+1
  FinTantQue
  Ecrire("le résultat est :",D)
Fin
```

- Le résultat est: 4
 Le résultat est: 12
 Le résultat est: 2
 Aucune de ces réponses

19: Donner le résultat de l'exécution de cet algorithme :

```
S ← 0;
pour i ← 1 à i ← 6 faire
  i ← i + 1;
  S = S + i;
Finpour
Ecrire (" S = ", S);
```

- Aucune réponse
 S = 9
 S = 21
 S = 12

20: Donner le résultat de l'exécution de cet algorithme :

```
S ← 0;
pour i ← 1 à i ← 6 faire
  S = S + i;
  i ← i + 1;
Finpour
Ecrire ("la somme S = ", S);
```

- S = 9
 S = 21
 S = 12
 Aucune réponse

21: Combien de fois la boucle suivante va s'exécuter ?

```
i ← 11;
répéter
  i ← i+1;
  écrire("Bonjour");
tantque (i<10)
```

- 1 fois
 Aucune réponse
 0 fois
 Boucle infinie

22: Combien de fois la boucle suivante va s'exécuter ?

```
i ← 11;
répéter
  i ← i - 1;
  écrire("bonjour ");
tantque ( i < 11);
```

- Aucune réponse
 1 fois
 0 fois
 Boucle infinie

23: Combien de fois la boucle suivante va s'exécuter ?

```
i ← 11;
répéter
  i ← i - 1;
  écrire("bonjour ");
jusqu'à ( i < 11);
```

- Boucle infinie
 Aucune réponse
 0 fois
 1 fois

24: On considère l'algorithme suivant :
 Donner le résultat qui serait affiché avec les données suivantes : N = 6, T1 = 1,3,8,9,12,15 et T2 = 1,3,8,9,12,16

```
a ← 0;
pour i ← 0 à i ← N faire
  si(T1[i]=T2[i]) alors
    a ← a + 1;
  finsi
si(a=n) alors
  écrire ("Ok");
sinon
  écrire("Non Ok");
fin
```

- OK
 Aucune réponse
 Non OK

25: On considère l'algorithme suivant :

```
Algorithme exo1
variables A, B X, Y,R : entiers ;
Debut
  écrire (" Donner deux entiers A et B");
  lire (A,B);
  x ← A;
  y ← B;
  R ← 0;
  tantque (x >= 1) faire
    si(x mod 2 = 1) alors
      R ← R + y
    finsi
    x ← x div 2
    y ← y * 2
  fintantque
  écrire ("la valeur trouvée est : ", R);
Fin
```

Donner le résultat de l'exécution de l'algorithme pour le couple (A=12, B=4).

- 48
 4
 16
 Aucune réponse

26: On considère l'algorithme suivant :
 Donner le résultat qui serait affiché avec les données suivantes : N = 6, TAB = 1,2,8,9,12,15

```
i ← 1;
tantque ((i<N) et (TAB[i]<=TAB[i+1]))
  i ← i + 1;
fintantque
si (i = N) alors
  écrire(" OK ");
sinon
  écrire("Non OK");
fin
```

- Non OK
 Aucune réponse
 OK

27: L'algorithme suivant calcul le PGCD de deux entiers m et n saisis au clavier :

```
Début
Var
  M, N, R : entier
  écrire("Donner deux entiers : ")
  lire( M , N)
  R ← M mod N
  tantque (R > 0)
    M ← N
    N ← R
    R ← M mod N
  fintantque
  écrire(" Le PGCD est : ", ..... )
Fin
```

Compléter l'algorithme en remplaçant les ... par ce qu'il faut.

- M
 N
 Aucune réponse
 R

28: On considère l'algorithme suivant :

```
Algorithme Test ;
Var A, B, P, I: entier;
Début
Ecrire("Donner deux entiers A et B");
Lire (A, B);
Si ((A=0) ou (B=0)) Alors
  P ← 0;
Sinon
  P ← 0;
  Pour I ← 1 à B Faire
    P ← P+A;
  FinPour
Fsi
Ecrire (A, B, I, P);
Fin.
```

Le résultat de l'algorithme pour A=7 et B= 5 affiche :

- 7 5 5 28
 7 5 6 35
 7 5 4 35
 7 5 4 28

CORRECTION

29: On considère l'algorithme suivant :

```

Algorithme test ;
Var P, R, x, BIN : entier ;
Début
P ← 1; BIN ← 0;
Ecrire (" donnez la valeur de x : " );
Lire(x) ;
Répéter
    R ← x Mod 2;
    BIN ← BIN + R * P;
    P ← P * 10;
    x ← x Div 2;
Jusqu'à (x=0) ;
Ecrire (R, BIN, P, x);
Fin

```

Donner le résultat qui serait affiché pour x=24.

- 1 11000 10000 1
- 0 11000 100000 0
- 1 11000 100000 0
- 1 1000 10000 1

30: On considère le code suivant :

```

var note : réel
Début
. repeter
.   ecrire(" donner une note: ");
.   lire (note);
.   jusqu'à (Condition);
Fin

```

Parmi les expressions suivantes laquelle permettra au programme ci-dessus de faire la lecture d'une note tant qu'elle n'est pas comprise entre 0 et 20.

- (note < 0) ou (note > 20)
- (note > 0) et (note > 20)
- (note >= 0) et (note <= 20)
- (note < 0) ou (note < 20)