

EMD n°2 d'Informatique (Durée : 50mn)

Remarque importante : Pour chaque question il y a une seule réponse juste

Généralités

Question 1 : la boucle « pour » a besoin toujours :

- A) d'un « début – fin » pour son bloc d'instructions,
- B) d'une instruction d'incrémentation de son compteur,
- C) d'une indication de la valeur du « pas »,
- D) toutes ces déclarations sont fausses.

Question 2 : la boucle « tant que » :

- A) est plus simple que la boucle « pour » car elle utilise moins d'instructions,
- B) est valable uniquement lorsqu'on ne connaît pas le nombre de répétitions,
- C) nous protège contre l'oubli de certaines instructions importantes,
- D) toutes ces déclarations sont fausses.

Question 3 : le remplacement de la boucle « pour » par la boucle « tant que » :

- A) est toujours possible sans ajouter des instructions,
- B) est toujours possible mais en ajoutant des instructions,
- C) n'est pas toujours possible,
- D) n'est jamais possible.

Question 4 : la boucle « pour » :

- A) a toujours besoin qu'on lui spécifie le « pas » (step),
- B) a besoin qu'on lui spécifie le pas uniquement s'il est supérieur à « 1 »,
- C) a besoin qu'on lui spécifie le pas uniquement s'il est différent de « 1 »,
- D) toutes ces réponses sont fausses.

Partie A – Soit le programme suivant :

```
Program calcul ;
Var a ; b : integer ;

Begin
write ('Donnez une valeur correcte pour votre donnée') ;
read (a) ;
b:=a*a;
while a <= 0 do
begin
write ('Donnez une valeur correcte pour votre donnée') ;
read (a);
end ;

b := 2*a ;
Ecrire ('le résultat = ', b);
End.
```

Question 5 : une erreur syntaxique est glissée dans ce programme, elle est située au niveau de :

- A) la déclaration,
- B) la 1^{ère} instruction read (a),
- C) la 2^{ème} instruction read (a),
- D) la boucle « while ».

Dans les questions suivantes, nous supposons que cette erreur est corrigée

Question 6 : Est-ce que ce bloc comporte une instruction inutile ?

- A) Oui, c'est la première instruction : read (a),
- B) Oui, c'est l'instruction : $b := a * a$,
- C) Oui, c'est l'instruction : $b := 2 * a$,
- D) toutes ces déclarations sont fausses

Question 7 : Quel est le rôle de l'instruction de boucle « while $a \leq 0$ do » ?

- A) elle n'a aucun rôle car elle contient les mêmes instructions du début du bloc,
- B) elle empêche le déroulement de la suite du programme si $a > 0$,
- C) elle exige que la valeur de « a » soit positive,
- D) toutes ces déclarations sont fausses

Question 8 : L'instruction « while $a \leq 0$ do » :

- A) peut être remplacée par la boucle « For » mais en utilisant la variable « a » elle-même comme compteur,
- B) peut être remplacée par la boucle « For » à condition d'ajouter une nouvelle variable comme compteur,
- C) ne peut pas être remplacée par la boucle « For » car on ne connaît pas le nombre de répétitions,
- D) toutes ces déclarations sont fausses.

Partie B - Soit la boucle suivante :

Pour i allant de 0 jusqu'à 50 faire ;
Si $(i \bmod 5 = 0)$ alors $s \leftarrow s + i$;

Question 09 : Cette boucle comporte-t-elle une erreur syntaxique ?

- A) non, syntaxiquement elle est juste,
- B) oui il manque des parenthèses pour l'instruction : « $s \leftarrow s + i$ »,
- C) oui il manque un « début - fin » pour le bloc de la boucle,
- D) oui il y a un point-virgule qui ne devait pas exister dans son endroit.

S'il y a erreur syntaxique, nous supposons dans ce qui suit qu'elle est corrigée.

Question 10 : Que fait cette boucle :

- A) elle fait la somme des 50 premiers entiers,
- B) elle fait la somme des multiples de « 5 » 50 fois,
- C) elle fait la somme des multiples de « 5 » qui sont inférieurs ou égal à 50,
- D) elle fait la somme des diviseurs de « 5 » 50 fois.

Question 11 : Pour que la boucle fonctionne correctement on doit ajouter :

- A) l'initialisation de la variable s ,
- B) l'initialisation du compteur : $i \leftarrow 0$,
- C) l'incrémentation du compteur $i \leftarrow i + 1$,
- D) le bloc « début ... fin ».

Question 12 : cette boucle « pour » peut être remplacée par la boucle « tant que » :

- A) en laissant le bloc tel qu'il est
- B) en ajoutant « début – fin » au bloc
- C) en ajoutant une initialisation et une incrémentation du compteur « i »
- D) en considérant à la fois les propositions B et C

Partie C – soit le programme suivant :

```
1. Program Formule ;
2. Var A, B, C, D, CD, FA, FB : integer ; i, j : integer;
3.   S, T : Real ;
4. Begin
5.   S :=0; A :=5; B :=2; C :=4; D :=1;
6.   For i := 1 to 4 do
7.     begin
8.       FA := 1;
9.       For j:=1 to A do
10.        FA:= FA * j;

11.        FB := 1;
12.        For j:=1 to B do
13.         FB:= FB * j;

14.        CD := D * D ;
15.        T := (FA * C + CD) / (FB - D) ;
16.        S := S + T;
17.        A := A + 2 ; B := B + 3 ; C := C + 4 ; D := D + 1 ;
18.      end ;
19.   Write ('La somme finale est égale à : ', S);
20. End.
```

Question 13 : les variables T et CD :

- A) doivent être initialisées avant la 1^{ère} boucle « For... »
- B) doivent être initialisés dans la boucle mais en début du bloc
- C) n'ont pas besoin d'initialisation
- D) toutes ces précédentes déclarations sont fausses

Question 14 : la variable T :

- A) devrait bien être de type « entier » puisque toutes les variables qui composent son expression de calcul sont de type « entier »,
- B) ne peut être que de type « réel » vu son expression de calcul,
- C) devrait suivre le type de la variable S,
- D) nous avons bien le choix de lui mettre le type qu'on veut.

Question 15 : Quelle formule mathématique calcule ce programme ?

A)
$$S = \frac{5! * 4 + 1}{2! - 1} + \frac{7! * 8 + 4}{5! - 2} + \frac{9! * 12 + 9}{8! - 3} + \frac{11! * 16 + 16}{11! - 4} + \frac{13! * 20 + 25}{14! - 5}$$

X B)
$$S = \frac{5! * 4 + 1}{2! - 1} + \frac{7! * 8 + 4}{5! - 2} + \frac{9! * 12 + 9}{8! - 3} + \frac{11! * 16 + 16}{11! - 4}$$

C)
$$S = \left(\frac{5! * 4}{2! - 1} + 1\right) + \left(\frac{7! * 8}{5! - 2} + 4\right) + \left(\frac{9! * 12}{8! - 3} + 9\right) + \left(\frac{11! * 16}{11! - 4} + 16\right) + \left(\frac{13! * 20}{14! - 5} + 25\right)$$

D)
$$S = \left(\frac{5! * 4}{2! - 1} + 1\right) + \left(\frac{7! * 8}{5! - 2} + 4\right) + \left(\frac{9! * 12}{8! - 3} + 9\right) + \left(\frac{11! * 16}{11! - 4} + 16\right)$$

Partie D - Soit le bloc de programme suivant :

```

Begin
Readln(x);
For i := 1 to x do
  Begin
  For j := 1 to x do
    Write (i + j, ' ');
  Writeln;
  End;
End.

```

***** Rappel important *****:

L'instruction : « write (i + j); » équivaut les 2 instructions « k ← i + j; write (k); »

Question 16 : Sachant que l'instruction « **writeln** » fait un saut de ligne, que sera l'affichage de ce bloc pour x = 4 :

A	X B	C	D
1 2 3 4	2 3 4 5	1 1 1 1	2 4 6 8
2 4 6 8	3 4 5 6	2 2 2 2	8 6 4 2
3 5 7 9	4 5 6 7	3 3 3 3	1 3 5 7
4 6 8 10	5 6 7 8	4 4 4 4	7 5 3 1

Question 17 : Comment modifier ce bloc pour avoir l'affichage suivant pour x = 4 ?

3 3 3 3
2 2 2 2
1 1 1 1
0 0 0 0

- A) En remplaçant write (i+j, ' ') par write (i-j, ' ')
- B) En remplaçant write (i+j, ' ') par write (x-j, ' ')
- X C) En remplaçant write (i+j, ' ') par write (x-i, ' ')
- D) En remplaçant write (i+j, ' ') par write (x-(i*j), ' ')

Question 18 : Combien de fois est exécutée l'instruction « write (i + j, ' ') » pour x = 4 ?

- A) 4 fois,
- B) 8 fois,
- C) 16 fois,
- D) Toutes ces réponses sont fausses

Question 19 : Soit la suite suivante : $S = 0 + 256 + 512 + 768 + \dots + 2048$

Quel est le bloc de programme qui calcule cette somme ?

A	<input checked="" type="checkbox"/> B	C	D
For i :=0 to 2048 do If (i mod 256=0) then writeln(i)	x := 0 ; while(x <= 2048) do Begin write(x) ; x :=x+256; End;	For i := 0 to 9 do write(i*256);	Aucun de ces blocs ne calcule cette somme

Question 20 : On dit que « x » est un « nombre parfait » si la somme de ses diviseurs est égale à son double (2x). Parmi ces programmes, lequel détermine si l'utilisateur a saisi un nombre parfait ?

A	B	C	D
s := 0; For i :=1 to x do if (x mod i =0) then s := s + i; if (s = x*2) then writeln (x, ' est un nombre parfait') ;	s := 0; For i :=1 to x div 2 do if (x mod i =0) then s := s + i; if (s = x*2) then writeln(x, ' est un nombre parfait') ;	s := x; For i :=1 to x div 2 do If (x mod i =0) then s := s + i; if (s = x) then writeln(x, 'est un nombre parfait') ;	Aucun de ces blocs ne détermine si un nombre est parfait

Bon courage

Dr Guessoum

Dr Talai

Corrigé Type

Barème par question : 1,000000

N°	Rép.
1	D
2	D
3	B
4	C
5	A
6	B
7	C
8	C
9	D
10	C
11	A
12	D
13	C
14	B
15	B
16	B
17	C
18	C
19	D
20	A



الأستاذة الدكتورة
إشريط رفيايل ياسين

Dr GUESSEM
[Signature]