Partiel n°1 : Correction

Exercice 1 : programme mystère. [5 points]

On considère le programme Ada suivant :

```
(1) Procedure CaFaitQuoi is
(2)   n, m, i, res : integer ;
(3) begin
(4)   put("Saisir deux entiers positifs") ;
(5)   get(n) ;
(6)   get(m) ;
(7)   i := 0 ;
(8)   res := 0 ;
(9)   while (i /= m) loop
(10)    res := res + 2*n ;
(11)    i := i + 1 ;
(12)  end loop ;
(13)  put("Le résultat est : ") ;
(14)  put(res) ;
(15) end CaFaitQuoi ;
```

instr.	n	m	i	res
(5)	2	–	–	–
(6)	2	3	–	–
(7)	2	3	0	–
(8)	2	3	0	0
(9)	2	3	0	0
(10)	2	3	0	4
(11)	2	3	1	4
(10)′	2	3	1	8
(11)′	2	3	2	8
(10)″	2	3	2	12
(11)″	2	3	3	12
(12)	2	3	3	12
(13)	2	3	3	12
(14)	2	3	3	12
(15)	2	3	3	12

(Q.1) Compléter dans le tableau la trace d'exécution du programme `CaFaitQuoi` à partir de l'instruction `(6)`, lorsqu'on saisit les entiers 2 et 3, correspondant respectivement aux valeurs de `n` et `m`. En déduire l'affiche final : Le résultat est : 12

(Q.2) Que se passe-t-il si on saisit `-2` pour l'entier `m` et un entier positif pour `n` ? Expliquer
On rentrera dans une boucle infinie. En effet, à la ligne (9), la condition sera toujours vraie car on aura toujours `i > m` (car `m=-2` et on part de `i=0`, qu'on incrémente à chaque tour de boucle).

(Q.3) Que se passe-t-il si on saisit `0` pour l'entier `m` et un entier positif pour `n` ? Expliquer
On ne rentre pas dans la boucle (car `i=m=0`), et le résultat final sera la valeur initiale de `res`, c'est-à-dire `0`.

(Q.4) Que se passe-t-il si on saisit `0` pour l'entier `n` et un entier positif pour `m` ? Expliquer
Le résultat final sera toujours `0`, peu importe la valeur de `m`. En effet, comme on part de `res=0` et que `n=0`, l'affectation de la ligne (10) donnera toujours `res=0+2*0=0`.

(Q.5) Que calcule le programme `CaFaitQuoi` si on saisit des entiers positifs n et m quelconques ? Le résultat final sera `2*n*m`. Le programme `CaFaitQuoi` calcule donc la fonction $(n, m) \mapsto 2nm$.

Exercice 2 : saisie de valeurs. [3 points]

(Q.1) Donner une condition sur n qui est vraie si et seulement si l'entier n est impair et strictement positif : `(n mod 2 = 1) and (n > 0)`

(Q.2) Donner une condition sur n qui est vraie si et seulement si l'entier n est un nombre entier pair divisible par 3 supérieur strictement à 11 : `(n mod 2 = 0) and (n mod 3 = 0) and (n > 11)`. De manière équivalente : `(n mod 6 = 0) and (n > 11)`

(Q.3) Écrire un programme `Ada` qui demande à l'utilisateur de saisir deux entiers initiaux `n0` et `m0`, puis redemande en boucle à l'utilisateur de saisir deux nouveaux entiers `n` et `m`, et qui ne s'arrête que lorsque les deux entiers saisis sont égaux aux entiers initiaux.

Réponse :

```
Put("Saisir un premier entier initial n0 : ");
Get(n0);
Put("Saisir un second entier initial m0 : ");
Get(m0);
loop
  Put("Saisir un entier : ");
  Get(n);
  Put("Saisir un autre entier : ");
  Get(m);
  exit when (n = n0) and (m = m0);
end loop;
```

Exercice 3 : affichage de figures. [6 points]

(Q.1) Écrire un programme Ada qui demande à l'utilisateur de saisir un entier positif impair et qui affiche la figure suivante :

```
Saisir un entier impair : 5
-----
| | |
-----
| | |
-----
```

Réponse :

```
while ( (n <= 0) or (n mod 2 /= 1) ) loop
  Put("Saisir un entier impair : ");
  Get(n);
end loop;

for i in 1 .. n loop
  for j in 1 .. n loop
    if ( i mod 2 = 1) then
      Put("-");
    elsif (j mod 2 = 1) then
      Put("|");
    else
      Put(" ");
    end if;
  end loop;
  New_Line;
end loop;
```

(Q.2) Écrire un programme Ada qui demande à l'utilisateur de saisir un entier positif, puis affiche un triangle ayant cette hauteur, de la façon suivante :

```
Entrez la hauteur du triangle : 5
------+------
-----+++-----
----+++++----
---+++++++---
--+++++++++--
```

Réponse :

```
while (n <= 0) loop
  Put("Entrez la hauteur du triangle : ");
  Get(n);
end loop;

for i in 1 .. n loop
  Put("--");
  for j in 1 .. 2*n-1 loop
    if ( (j <= n-i) or (j >= n+i) ) then
      Put("-");
    else
      Put("+");
    end if;
  end loop;
  Put("--");
  New_Line;
end loop;
```

(Q.3) Écrire un programme Ada qui demande à l'utilisateur de saisir un entier pair positif et qui affiche le losange suivant :

```
Saisir un entier pair : 6
  /\
 //\\
///\\\
\\\///
 \\//
  \/
```

Réponse :

```
while ( (n <= 0) or (n mod 2 /= 0) ) loop
  Put("Saisir un entier pair : ");
  Get(n);
end loop;

for i in 1 .. n/2 loop                                    -- partie du haut
  for j in 0 .. n loop
    if ( j <= n/2 and j >= n/2 - i+1 ) then
      Put("/");
    elsif ( j >= n/2 and j <= n/2 + i ) then
      Put("\");
    else
      Put(" ");
    end if;
  end loop;
  New_Line;
end loop;

for i in reverse 1 .. n/2 loop                             -- partie du bas
  for j in 0 .. n loop
    if ( j <= n/2 and j >= n/2 - i+1 ) then
      Put("\");
    elsif ( j >= n/2 and j <= n/2 + i ) then
      Put("/");
    else
      Put(" ");
    end if;
  end loop;
  New_Line;
end loop;
```

Exercice 4 : polygone. [6 points]

(Q.1) Écrire un programme Ada qui demande à l'utilisateur les coordonnées des sommets d'un polygone (à au moins 3 côtés), puis affiche le périmètre du polygone. L'algorithme devra lire une suite de points du plan, chaque point étant donné par deux nombres entiers : son abscisse et son ordonnée. La suite se termine quand l'utilisateur saisit un point correspondant au tout premier point saisi *(indication : reprendre le programme de la question (Q.3) de l'exercice 2 pour tester si les coordonnées d'un sommet saisi sont les mêmes que celles du sommet initial)*.

Réponse :

```
with Ada.Integer_Text_IO; use Ada.Integer_Text_IO;
with Ada.Text_IO; use Ada.Text_IO;
with Ada.Float_Text_IO; use Ada.Float_Text_IO;
with Ada.Numerics.Elementary_Functions;use Ada.Numerics.Elementary_Functions;

procedure PERIMETRE is
  x0, y0, x, y, xx, yy : integer ;
  nbre_cotes : integer := 1 ;
  perimetre : float := 0.0 ;
begin
  Put("Saisir l'abscisse du 1er point : ") ;
  Get(x0) ;
  Put("Saisir l'ordonnée du 1er point : ") ;
  Get(y0) ;
  x := x0 ;
  y := y0 ;

  loop
    Put("Saisir l'abcisse du point suivant : ") ;
    Get(xx) ;
    Put("Saisir l'ordonnée du point suivant : ") ;
    Get(yy) ;

    if (xx=x0 and yy=y0 and nbre_cotes <= 2) then
      Put("Le polygone doit avoir au moins 3 sommets distincts. Ressaisir au
moins un sommet de plus") ;
    end if ;

    if (xx/=x0 or yy/=y0) then
      nbre_cotes := nbre_cotes + 1 ;
    end if ;

    perimetre := perimetre + sqrt( float( (xx-x) ** 2 + (yy-y) ** 2 ) ) ;
    x := xx ;
    y := yy ;
    exit when ( xx=x0 and yy=y0 and nbre_cotes > 2) ;
  end loop ;

  Put("Son périmètre est : ") ;
  Put(perimetre,2,2,0) ;

end PERIMETRE ;
```

(Q.2) Modifier le programme pour qu'en plus, il affiche le nombre de côtés du polygone et donne son nom (triangle, quadrilatère, pentagone, hexagone, heptagone, octogone, ennéagone, décagone et polygone à plus de 11 côtés).

Réponse :

```
  Put("Le polygone a " & integer'image(nbre_cotes) & " côtés.") ;
  Put("C'est donc un ") ;

  case nbre_cotes is
    when 3 => Put("triangle.") ;
    when 4 => Put("quadrilatère.") ;
    when 5 => Put("pentagone.") ;
    when 6 => Put("hexagone.") ;
    when 7 => Put("heptagone.") ;
    when 8 => Put("octogone.") ;
    when 9 => Put("ennéagone.") ;
    when 10 => Put("décagone.") ;
    when others => Put("polygone avec beaucoup de côtés (plus de 10 en
tout cas).") ;
  end case ;
```