Université A. Mira Béjaia
Faculté de Sciences Exactes
Département Mathématique

L1 Math (tronc commun)
Module : ASD1
2024/2025

Corrigé du EMD 1 ASD1

Exercice 1(6 points)

1. Déroulement de l'organigramme pour les deux cas:
   a) a=10, b=2

1,5

	(a>0) et (b>0)	a>=b	a	b	R	Affichage
initialement	V	/	10	2	0	/
		V	8	2	0+1=1	/
		V	6	2	1+1=2	/
		V	4	2	2+1=3	/
		V	2	2	3+1=4	/
		V	0	2	4+1=5	/
		F	0	2	5	5

   b) a=13, b=3

1.5

	(a>0) et (b>0)	a>=b	a	b	R	Affichage
initialement	V	/	13	3	0	/
		V	10	3	0+1=1	/
		V	7	3	1+1=2	/
		V	4	3	2+1=3	/
		V	1	3	3+1=4	/
		F	1	3	4	4

2. Cet organigramme permettant de calculer et afficher le résultat de la **division entière** de deux nombres entiers strictement positifs **a div b**.

1

3. Transformer l'organigramme en un algorithme.

```
Algorithme Division ;
Var a,b,R : entier ;
 Debut
  Repeter
    Lire (a,b) ;
  Jusqu'à ((a>0)et (b>0));
   R←0;
   Tantque (a>=b) faire
     a←a-b;
     R←R+1 ;
   Fintantque;
   Ecrire(R);
 Fin.
```

2

Exercice 2 (7 points)

```
Algorithme Nombre;
Var  n, i, s : entier;
```

1)

```
Fonction Puiss (a,b: entier): Entier ;
Var P, i : entier ;
 Debut
   P←1;
   Pour i allant de 1 à b faire
     P←P*a;
   Finpour;
   retourner(P);
 Fin;
```

1,5

2)

```
Fonction NB_chiffres (x: entier): Entier ;
Var nb: entier ;
 Debut
   nb←0;
   Tantque (x<>0) faire
     nb←nb+ 1;
     x←x Div 10 ;
   Fintantque;
   retourner(nb);
 Fin;
```

1,5

3)

```
Debut  {A. P}
    Pour i allant de 1 à 9999 faire
      nbc←Nb_chiffres(i);
      s←0; n←i;
      Tantque (n<>0) faire
        s←s+ Puiss(n mod 10, nbc);
        n←n Div 10 ;
      Fintantque;
      Si (i= s) alors
        Ecrire(i);
      Finsi;
    Finpour;
Fin.
```

2,5

4) Transformation de la fonction **Nb_chiffres** en une procédure :

1,5

1. **Procedure** Nb_chiffres (x: entier ; **Var nb: entier**);
2. Enleve l'instruction **retourner(nb)**
3. L'appelle de la procedure dans l'algorithme principal devient :

   Nb_chiffres(i, nbc)

Exercice 3 **(7 points)**

```
Algorithme Exercice3;
Type Tab= Tableau [1..100] de entier;
Var  n, m, i, j, k : entier;   T1, T2, T: Tab;

Procedure Remplir _Tab_Positif(Var T:Tab; n: entier);
Var i : entier ;
 Debut
    Pour i allant de 1 à n faire
      repeter
       Lire(T[i]);
      Jusqu'à (T[i]> 0);
    Finpour;
 Fin;

Fonction existe (T: Tab; n,v: entier): Booleen ;
Var i: entier ; Tr: booleen;
 Debut
   i←1; Tr← Faux;
    Tantque (i<=n) et (Tr=Faux)faire
       Si (T[i]=v) alors
          Tr← Vrai;
       Sinon
          i←i+1;
       Finsi;
    Fintantque;
   retourner(Tr);
 Fin;

Fonction premier (n: entier): Booleen ;
Var i: entier ; Pr: booleen;
 Debut
   i←2; Pr← Vrai;
    Tantque (i<=n div 2) et (Pr=Vrai)faire
       Si (n mod i=0) alors
          Pr← Faux;
       Sinon
          i←i+1;
       Finsi;
    Fintantque;
      retourner(Pr);
 Fin;


Procedure Afficher _Tab(T:Tab; n: entier);
Var i : entier ;
 Debut
    Pour i allant de 1 à n faire
       Ecrire(T[i]);
    Finpour;
 Fin;
```

```
Debut {A. P}
1)  Repeter                                          0,5
      Lire (n,m) ;
    Jusqu'à ((n>0)et (m>0));
    Remplir_Tab_Positif(T1, n);                      2
    Remplir_Tab_Positif(T2, m);
2)  j←0;
    Pour i allant de 1 à n faire
      Si (existe(T2, m, T1[i])= faux) alors          2
          j←j+1;
          T[j]←T1[i];
      Finsi;
    Finpour;
3)  Afficher_Tab(T, j);                              1
4)  Pour i allant de 1 à n faire
      Si (premier(T1[i])= Vrai) alors                1,5
          Ecrire(T[i]);
      Finsi;
    Finpour;
Fin.
```