

Corrigé type

Exercice 1 : (8 points)

Question 1, 2, 3

Algorithme Ex 1

Variables :

T: tableau [1..20] d'entier ; i :entier ;

Procédure **Somme_tab** (T : tableau [1..N] d'entier) (1 P)

Variables : i, s : entier ;

Debut

s ← 0 ;

Pour i ← 1 jusqu'à N

 s ← s + T[i] ;

Fin pour ;

 Ecrire (s) ;

Fin ;

Procédure **Somme_nbr_premiers** (T : tableau [1..N] d'entier) (1 P)

Variables : i, j, sp, nbr_div: entier ;

Debut

sp ← 0 ;

Pour i ← 1 jusqu'à N

 nbr_div ← 0 ;

 Pour j ← 2 jusqu'à T[i]/2

 Si T[i] mod j = 0 ;

 nbr_div ← nbr_div + 1 ;

 fin si ;

 Fin pour ;

 Si nbr_div ≠ 0

 sp ← sp + T[i] ;

 fin si

 Fin pour ;

 Ecrire (sp) ;

Fin ;

Debut (1 P)

Pour $i \leftarrow 1$ jusqu'à 20

lire ($T[i]$) ;

Fin pour ;

Somme_tab (T) ;

Somme_nbr_premiers (T) ;

Fin.

Question 4

Non (0,5 P)

Justification : Les variables "s", déclarée pour calculer la somme des éléments du tableau, et "sp", déclarée pour calculer la somme des Nbr premiers, sont **des variables locales**. En effet, elles ne peuvent pas être utilisées dans l'algorithme principal pour calculer la somme des éléments non premiers car l'espace de ces variables locales est libéré dès que l'exécution des procédures est terminée **(0,5 P)**

Question 5

Algorithme Ex 1

Variables :

T: tableau [1..20] d'entier ; i, **S_tab, S_premiers, S_non_premiers** : entier;

Fonction Somme_tab (T : tableau [1..N] d'entier) : entier (0.75 p)

Variables : i, s : entier ;

Debut

$s \leftarrow 0$;

Pour $i \leftarrow 1$ jusqu'à N

$s \leftarrow s + T[i]$;

Fin pour ;

Somme_tab $\leftarrow s$;

Fin ;

Procédure Somme_nbr_premiers (T : tableau [1..N] d'entier) : entier (0.75 p)

Variables : i, j, sp, nbr_div: entier ;

Debut

$sp \leftarrow 0$;

Pour $i \leftarrow 1$ jusqu'à N

$nbr_div \leftarrow 0$;

Pour $j \leftarrow 2$ jusqu'à $T[i]/2$

Si $T[i] \bmod j = 0$;

$nbr_div \leftarrow nbr_div + 1$;

fin si ;

Fin pour ;

Si $nbr_div \neq 0$

$sp \leftarrow sp + T[i]$;

fin si

```

Fin pour ;
Somme_nbr_premiers ← sp;
Fin ;

```

→ Oui il est possible de calculer la somme des éléments non-premier dans l'algorithme principal comme suit **(0.5p)**

```

Debut (0.5 P)
  Pour i ← 1 jusqu'à 20
    lire (T[i]);
  Fin pour ;
  S_tab ← Somme_tab (T);
  S_premiers ← Somme_nbr_premiers (T);
  S_non_premiers ← S_tab - S_premiers ;
  Ecrire (S_non_premiers) ;
Fin.

```

Question 6 (1.5p)

Fonction Somme_tab_R (T : tableau [1..N] d'entier, N : entier) : entier ;

```

Début
  Si N=1 alors
    Somme_tab_R ← T[1];
  Sinon
    Somme_tab_R ← T[N] + Somme_tab_R (T, N-1);
  Fin si
Fin ;

```

Exercice 2: (5 p)

Question 1 : (1.5 p)

L'algorithme calcule la somme de x et y / $z=x+y$

Question 2 : (1.5 p)

```

#include <stdio.h>
main ()
{
  int x,y,z ;
  int *P1, *P2 ;
  scanf ("%d", &x) ;
  scanf ("%d", &y) ;
  P1 = & x ;
  P2 = & y ;
  z = (*P1) + (*P2) ;
  printf ("%d", z) ;
}

```

Question 3 : (2 p)

Algorithme Exo 2

Var z : entier ;
P1, P2 : *entier ;

Début

Allouer (P1) ;
Allouer (P2) ;
Lire (*P1) ;
Lire (*P2) ;
 $z \leftarrow (*P1) + (*P2)$;
Ecrire (z) ;

Fin.

Exercice 3: (7 points)

Question 1 : (5 p)

Déclaration (1.5 pts), Lecture (1.5 pts), Affichage (2 pts)

Algorithme Ex3 ;

Type Date = enregistrement
 Jour : entier ; * ou bien jour : 1..31 ;
 Mois : entier ; * ou bien jour : 1..12 ;
 Annee : entier ;
Fin

(0.5)

Type Etudiant = enregistrement
 Num_Inscription : entier ;
 Nom : chaine de caractères (20) ;
 Prenom : chaine de caractères (30) ;
 Date_naiss : Date ;
 Specialite : chaine de caractères (20) ;
 Num_G : entier ;
 Moyenne : entier ;

(0.5 pt)

Fin

Variables

T : tableau [1..100] d'Etudiant ;
i, N : entier ;

(0.5 p)

moy_major : entier ;
nom_major : chaine de caractères (20) ;
prenom_major : chaine de caractères (30) ;

Debut

Ecrire ('entrez la valeur de N (N<=100)') ;
Lire (N) ;

 | **Lecture du tableau d'enregistrements (Etudiant)*

Pour i allant de 1 jusqu'à N faire

 | Ecrire ('entrez le numéro d'inscription de l'étudiant n° :', i) ;
 | Lire (T[i].Num_Inscription) ;
 | Ecrire ('entrez le nom et le prenom de l'étudiant n° :', i) ;
 | Lire (T[i].Nom, T[i].Prenom) ;
 | Ecrire ('entrez la date de naissance de l'étudiant n° :', i) ;
 | Ecrire ('entrez le jour :') ; lire (T[i].Date_naiss.Jour) ;
 | Ecrire ('entrez le mois :') ; lire (T[i].Date_naiss.Mois) ;
 | Ecrire ('entrez l'année :') ; lire(T[i].Date_naiss.Anee) ;
 | Ecrire ('entrez la Spécialité de l'étudiant n° :', i) ; Lire (T[i].Specialite) ;
 | Ecrire ('entrez le Num groupe de l'étudiant') ; lire(T[i].Num_G) ;
 | Ecrire ('entrez la Moyenne de l'étudiant') ; lire(T[i]. Moyenne) ;

FinPour

 | **Affichage des informations du major de promo*

 | Ecrire ('les informations du **major de promo**:') ;

 | Moy_major ← T[1].moyenne ;

Pour i allant de 2 jusqu'à N faire

 | Si (T[i]. Moyenne > moy_major) Alors

 | moy_major ← T[i]. Moyenne ;

 | nom_major ← T[i]. Nom ;

 | prenom_major ← T[i]. Prenom ;

 | Finsi

FinPour

 | Ecrire (nom_major) ;

 | Ecrire (prenom_major) ;

 | Ecrire (moy_major) ;

Fin.

(1.5)

(2 p)

Question 2 : (2 p)

- Non, on ne peut pas car la taille du tableau **est fixe** (1 p)
- La solution alternative : l'utilisation des listes chaînées, car la dimension de cette dernière évolue selon les besoins (1 p)