

Correction examen Algorithmique et programmation C 1 (2017/2018) 1ère année

—Enseignante : Mme Aïcha El Golli Jabbes—

1h30 – Documents et calculettes interdits

► Exercice 1. Appel Fonction et procédure (5 Points)

Soit le programme suivant :

Procédure Deux(d/r A :entier)

Début

A \leftarrow A + 1 ;

afficher(A) ;

Fin

Procédure Un(B :entier)

Début

B \leftarrow B +1 ;

Deux (B) ;

afficher(" B= ", B) ;

Fin

Fonction Trois(C :entier) :entier

Début

Retourne(C+2) ;

Fin

Algorithme Essai

a, b, c : entier ;

Début

a \leftarrow 10 ;

Deux(a) ;

Un(a) ;

b \leftarrow 10 ;

Un(b) ;

Deux(b) ;

c \leftarrow 10 ;

c \leftarrow Trois(c) ;

afficher("a= ", a) ;

afficher("b= ", b) ;

afficher("c= ", c) ;

Fin

1. Exécutez le et donnez la séquence des affichages produits
2. Traduire la **procédure** Un() ainsi que la **fonction** Trois() en langage C



11 (0.25pt)
13 (0.25pt)
B= 13 (0.25pt)
12 (0.25pt)
B= 12 (0.25pt)

11 (0.25pt)
a=11 (0.5pt)
b=11 (0.25pt)
c=12 (0.25pt)

```
void Un (int B)//(0.25pt)
{B = B +1 ;//(0.25pt)
  Deux (B) ;//(0.5pt)
  printf("B= %d \n", B) ;//(0.5pt)
}
```

```
int Trois(int C)//(0.5pt)
{return(C+2) ;}//(0.5pt)
```



► **Exercice 2. Langage C (6 Points)**

1. Donnez le nombre de boucles pour chacune des instructions suivantes :

a.	<pre>int i=1; while(--i){printf("hello\n"); i++;}</pre>	✂ 0 boucle (1pt) ✂
b.	<pre>int i=1; while(++i<10){i==2?i=10:i--; printf("hello\n"); }</pre>	✂ 1 boucle (1pt) ✂
c.	<pre>int i=10; do printf("hello\n"); while (++i< 10);</pre>	✂ 1 boucle (1pt) ✂
d.	<pre>for(i=1,j=3;i<j;i++,j--) {printf("hello\n");}</pre>	✂ 1 boucle (1pt) ✂

2. Quelles sont les valeurs des variables i, j et k après exécution de ces instructions :

```
int i, j = 5, k=7;
i= k++;
k= --j;
j= i++;
```

✂
i=8, j=7, K=4 (1pt)
..... ✂

3. Qu'affiche le code C suivant :

```
char ch1 [] = "Bonjour";
char ch2 [] = "tout va bien?";
int i = 0, j =0;
while ( ch1 [i++] != 0)
{if(i!=2)  ch2[j]=ch1[i];
j++;}
puts(ch2);
```



oojour (2pt)

**► Exercice 3. (9 Points)**

On désire écrire un algorithme qui construit un tableau de notes et qui permet ensuite de faire certains traitements sur le tableau. L'algorithme, commenté par la suite, est le suivant :

Algorithme tableauNotes**constantes**

max :entier=100;

variables

tabnote [max] : réel;

nbVal : entier;

Début**init** (tabnote, nbVal);afficher("la moyenne des element= ", **calculmoyenne** (tabnote, nbVal));afficher("le plus grand écart dans le tableau= ", **grandecart** (tabnote, nbVal));**supprimer**(tabnote, nbVal);**Fin**

1. Définir une procédure **init()** qui demande à l'utilisateur de saisir la taille du tableau, entier n positif ou nul (cette valeur ne doit pas aussi dépasser la taille maximum du tableau, redemander si nécessaire), et qui ensuite saisit les n notes comprises entre 0 et 20.0 d'un tableau de réels. Ce tableau ainsi que sa taille doivent être renvoyés par le biais des paramètres.

**Procédure** **init**(d/r t[] : reel, d/r n : entier) (0,5pt)

i : entier; (0,25pt)

Début**Répéter**

Afficher("Entrer la taille du tableau : ");

Lire(n); (0,25pt)

Jusqu'à (n>=0) et (n<=100); (0,25pt)

Afficher("Donner les éléments du tableau : ");

Pour i \leftarrow 0 à n-1 (0,25pt)**répéter**

Lire(t[i]); (0,25pt)

jusqu'à (t[i] >= 0.0) et (t[i] <= 20.0); (0,25pt)**FPour; Fin**

2. Définir une fonction **calculmoyenne** () qui retourne le calcul de la moyenne des éléments du tableau.

**Fonction** **calculmoyenne**(t[] : reel, n : entier) : réel (1pt)

som : reel;

```

i : entier ;
Début
som ← 0.0 ;
Pour i ← 0 à n-1
    som ← som + t[i] ;
Finpour ;
si (n=0) alors retourne 0.0 ;
sinon retourne som/n ;
Finsi ; Fin ;

```

3. Définir une fonction **grandecart()** qui permet de retourner le plus grand écart au sein du tableau. Traduire cette fonction en langage C.

<pre> Fonction grandecart(t[] : reel, n : entier) : réel min, max : reel ; i : entier ; (0,25pt) Début si (n!=0) alors min ← t[0] ; max ← t[0] ; (0,5pt) Pour i ← 1 à n-1 (0,25pt) si (min > t[i]) alors min ← t[i] ; Fsi ; (0,25pt) si (max < t[i]) alors max ← t[i] ; Fsi ; (0,25pt) Fpour ; retourne (max-min) ; (0,25pt) sinon retourne (0.0) ; (0,25pt) </pre>	<pre> Finsi ; Fin ; float grandecart(float t[] , int n) { float min, max ; int i ; (0,25pt) if (n!=0) { min = t[0] ; (0,25pt) max = t[0] ; (0,25pt) for (i = 1 ; i < n ; i++) { (0,25pt) if (min > t[i]) min = t[i] ; (0,25pt) if (max < t[i]) max = t[i] ; (0,25pt) } return (max-min) ; } (0,25pt) else return (0.0) ; (0,25pt) } </pre>
--	--

4. Définir une procédure **supprimer()** qui permet de supprimer un élément dans un tableau. Cette procédure demandera à l'utilisateur l'indice de l'élément à supprimer si cet indice existe l'élément sera supprimé, sinon rien ne sera effectué.

```

Procédure supprimer(d/r t[] : reel, d/r n : entier) (0,5pt)
i, pos : entier ; (0,25pt)
Début
lire(pos) ; (0,25pt)
si pos >= 0 et pos < n alors (0,25pt)
    Pour i ← pos à n-2 (0,25pt)
        t[i] ← t[i+1] ; (0,25pt) Fpour ;
    n ← n-1 ; (0,25pt)
Finsi ; Fin ;

```

Bon travail