



Remarques : Les Documents, calculatrices, téléphone portable sont interdits.

Veuillez rendre une copie propre et claire. Si la syntaxe d'une instruction est fautive alors la note est 0. La qualité de l'écriture et de la présentation sera prise en compte dans la note finale.

Questions de réflexion (3 points)

1. Définissez le rôle d'un compilateur ? (1pt)

Un compilateur permet de :

a. Vérifier la syntaxe du programme.

b. Traduire le programme en langage machine.

2. Quelle est la sortie de ce code C ? (2pt)

```
#include <stdio.h>
void main()
{int i = 9;
 while(i++ < 10){
 switch(i){
 case 9 : puts("start");
 case 10 : puts("end");
```

```
default : puts("Good By");
}
}
printf("%d\n", i);}
```

end (0.5pt)

Good By (0,5pt)

11 (1pt)

Exercice 1 (4 points) Exécution

Soit l'algorithme suivant :

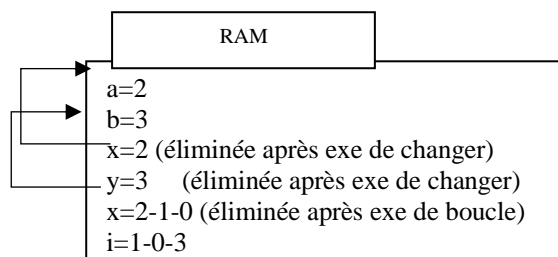
```
Algorithme Exo
a,b : entier ;
Début
a ← 2;
changer(a,b);
afficher("b= ", b , "a= ", a );
```

Fin

```
Procédure changer(a/r x : entier, r y :entier )
Début
y ← boucle(x);
afficher(x);
Fin
```

```
Fonction boucle(x : entier) : entier
i :entier
Début
i ← 1 ;
repete
si (x mod 2 =0) alors
i ← i-1;
sinon
i ← i+3 ;
FinSi
x ← x-1 ;
jusqu'à (x ≤ 0) ;
retourne i ;
Fin
```

1. Exécuter le et donner la séquence des affichages produits (1,5pt)



Ecran

2 (0.5pt)
b=3 a=2 (1pt)

2. Traduire la fonction **boucle** en langage C (2,5pt)

```
int boucle (int x){ (0,25pt)
int i =1; (0,25pt)
do{ (0,25pt)
if(x%2==0) i=i-1; (0,5pt)
else i=i+3; (0,5pt)
```

```

x=x-1;      (0,25pt)
}while(x>0) ; (0,25pt)
return i; (0,25pt)
}

```

Exercice 2 (3 points) Chaîne de caractères

Finir le programme suivant pour compter le nombre total de mots dans une chaîne de caractère

```

#include <stdio.h>
#include <string.h>

void main()
{
    char str[100];
    int i, count = 0;
    printf("Entrer une chaîne de caractère : ");
    gets(str);
    for(i = 0; str[i] != '\0'; i++) (1pt)
    {
        if(str[i] == ' ' ) (1pt)
        {
            count++; (1pt)
        }
    }
    printf("Le nombre total de mots dans cette chaîne est = %d", count);
}

```

Exercice 3 (10 points)

On imagine une visite médicale pour des patients où on enregistre simultanément leur taille et leur poids dans deux tableaux différents.

Nous allons demander le nombre total des patients d'abord, pour enregistrer simultanément dans un premier tableau nommé « Tailles » la taille ($0 \leq t \leq 3,5$) en mètres et dans un deuxième tableau nommé « Poids » le poids ($0 \leq p \leq 635,0$) en kilogrammes, de tous les patients qui se présentent.

Nous allons ensuite calculer et afficher la taille moyenne et le poids moyen des patients.

Finalement, pour chaque patient, nous allons calculer l'indice de masse corporelle : $IMC = p/t^2$ et afficher:

- "insuffisance pondérale" si $IMC < 18.5$,
- "en surpoids" si $IMC > 25$,
- ou "normal" sinon.

L'algorithme, commenté par la suite, est le suivant :

Algorithme IMC

variables

moyTaille, moyPoids, Tailles[100], Poids[100] : réel ;
n: entier ;

Début

init (Tailles, Poids, n) ; {procédure init décrite ci-dessous}

moyTaille ← calcul_moy(Tailles, n) ; {fonction calcul_moy(t,n) : $\frac{\sum_{i=0}^{n-1} t[i]}{n}$ }

moyPoids ← calcul_moy(Poids, n) ; {fonction calcul_moy(t,n) : $\frac{\sum_{i=0}^{n-1} t[i]}{n}$ }

afficher("taille moyenne patient= " , moyTaille, " poids moyen patients= " , moyPoids) ;

CalculerIMC(Tailles, Poids, n) ; {procédure CalculerIMC décrite ci-dessous}

Fin

1. Définir une procédure **init()** en algorithmique qui demande à l'utilisateur de saisir la valeur d'un entier ***n* strictement** positif (cette valeur ne doit pas aussi dépasser la taille maximum du tableau, vérifier que la valeur saisie est bonne et redemander si nécessaire), et qui ensuite saisit les ***n*** éléments des deux tableaux de réels. Les valeurs des tailles et des poids acceptées sont ($0 \leq \text{taille} \leq 3,5$) et ($0 \leq \text{poids} \leq 635,0$)

(2pt)

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

i : entier ;

Debut

Répéter (0,5pt)

Afficher(« donner un nombre positif inférieur à 100 ») ;

Entrer(n) ;

Jusqu'à (n>=0)et (n<=100) ;

Pour(i de 0 à n-1) **faire**

Répéter (0,5pt)

Afficher(« donner taille <=3,5 ») ;

Entrer(t[i]) ;

Jusqu'à (t[i]>=0)et (t[i]<= 3,5) ;

Répéter (0,5pt)

Afficher(« donner poids inférieur à 635,0 ») ;

Entrer(p[i]) ;

Jusqu'à (p[i]>=0)et (p[i]<=635,0) ;

FINPOUR ;

fin

2. Définir la fonction **calcul_moy ()** en algorithmique qui calcule la moyenne arithmétique des éléments d'un tableau de réels : $\frac{\sum_{i=0}^{n-1} t[i]}{n}$

et qui renvoie la moyenne calculée. Cette fonction prend en paramètre le tableau et sa taille.

(2pt)

fonction **calcul_moy**(t[] :réel, n :entier) **:réel** (0,25pt)

s :réel ;

i :entier ; (0,25pt)

début

s←0,0; (0,25pt)

Pour (i de 0 à n-1) faire (0,25pt)

s←s+t[i] ; (0,25pt)

FinPour ;

Si(n<>0) retourne (s div n); (0,5pt)

Sinon retourne 0.0 ; (0,25pt)

Fin SI

Fin

3. Définir une procédure **CalculerIMC ()** en algorithmique qui consiste à calculer pour chaque patient l'indice de masse corporelle : $IMC = p/t^2$ et afficher:

- "patient " i " insuffisance pondérale" si $IMC < 18.5$,
- "patient " i " en surpoids" si $IMC > 25$,
- ou "patient " i " normal", sinon.

(3pt)

Procédure **calculerIMC**(t[] : réel, p[] :réel, n : entier) (0,25pt)

i :entier ;

Imc : réel ; (0,25pt)

Début

Pour (i de 0 à n-1) Faire (0,25pt)

imc \leftarrow p[i] / (t[i] * t[i]); (0,25pt)

si (imc < 18.5) alors (0,25pt)

afficher("patient " ,i+1 " insuffisance pondérale") ; (0,25pt)

Sinon (0,25pt)

si(imc > 25) alors (0,25pt)

afficher("patient " ,i+1 " en surpoids") ; (0,25pt)

Sinon (0,25pt)

afficher("patient " ,i+1 " normal") ; (0,25pt)

Finsi

Finsi ; (0,25pt)

FinPour ;

FIN ;

3. Traduire la fonction **calcul_moy** () en langage C.

(3pt)

float calcul_moy(float t[], int n) (0,5pt)

{

float s=0.; int i; (0,5pt)

for(i=0;i<n;i++) (0,5pt)

{

s+=t[i]; (0,5pt)

}

if(n!=0) return (s/n); (0,5pt)

else return 0.0; (0,5pt)

}

Bon travail