

Corrigé type : التصحيح النموذجي

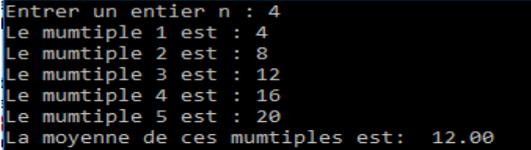
Exercice 1 التمرين 1 : (Barème : 4pts : تنقيط) : a : 1pts - b : 1pts - c : 1pts - d: 1pts.

a. F : Le type <i>double</i> est représenté sur un plus grand nombre de bits que le type <i>float</i> , offrant alors une précision plus élevée et une plage de valeurs plus étendue. E : Type <i>double</i> is represented on much more bits than <i>float</i> , thus offering higher precision and a wider range of values.
b. F : Un algorithme est dit universel lorsqu'il traite tous les cas possibles du problème qu'il résout. E : An algorithm is said to be universal if it treats all cases within the problem it resolves.
c. F : Dans le boucle <i>Tantque</i> (<i>while</i> { } de C), la condition est au début : Nombre minimum de boucles : 0, alors que dans la boucle <i>répéter</i> (<i>do</i> { } <i>while</i> de C), la condition est à la fin : Nombre minimum de boucles : 1. E : In the <i>while</i> loop, the condition is in the beginning: Minimal number of loops: 0, whereas in the <i>repeat</i> (<i>Do</i> { } <i>while</i> of C), the condition is at the end: Minimal number of loops: 1.
d. F : Dans le langage C, <i>typedef</i> permet de créer de nouveaux types dits aussi : types personnalisés. E : In the C language, <i>typedef</i> allows creation of a new types, said also: user defined types.

Exercice 2 التمرين 2 : (Barème : 4pts : تنقيط) : Logique: 1.5pts Exactitude: 1.5pts Syntaxe: 0.5pts structure générale: 0.5pts.

Langage algorithmique	OU Langage C
Algorithme Reduction Var prix : réel // Données nouvprix : réel // Résultats Début Ecrire('Donnez prix :') Lire(prix) Si (prix < 1000) alors nouvprix ← prix * (1 - 0.25) Sinon Si (prix < 5000) alors nouvprix ← prix * (1 - 0.2) Sinon nouvprix ← prix * (1 - 0.1) Finsi fin Ecrire('prix= ', prix, 'remise=', prix - nouvprix, 'nouveau prix=', nouvprix) Fin	<pre>#include <stdio.h> int main(){ float prix, nouvprix, percrem; printf("Prix= "); scanf("%f", &prix); if(prix < 1000) percrem = 0.25; else if(prix<5000) percrem = 0.2; else nouvprix = (1 - 0.1) *prix; nouvprix = (1 - percrem) * prix; printf("\n Prix percentremise: Remise Nouveau Prix "); printf("\n %6.2f %6.2f %6.2f %6.2f ", prix,percrem,prix- nouvprix,nouvprix); return 0; } Exécution : Prix= 750 Prix percentremise: Remise Nouveau Prix 750.00 0.25 187.50 562.50 -----</pre>

عربية	يؤخذ بعين الاعتبار أحد الإثنين : الخوارزمية أو برنامج C
Français	Il est pris en considération une des deux formes : l'algorithme ou le programme C
English	Only one of the two is considered : the algorithm OR the C program

Langage Algorithmique – Algorithmic Language	Langage C – C Language
<p>Algorithme Multiples</p> <p>Var n, m, som, i : entier moy : réel t : tableau [1..5] d'entiers</p> <p>Début Ecrire('Entrer un nombre entier n :'); Lire(n) som ← 0 Pour i de 1 à 5 faire m ← n * i Ecrire('Le multiple est :', m) t[i] ← m som ← som + m FinPour moy ← som / 5 Ecrire('La moyenne de ces multiples est :', moy) Fin</p>	<pre>#include <stdio.h> int n , m ,som ; float moy ; int t[5] ; int main() { int i ; printf("Entrer un entier n : "); scanf ("%d", &n); for (i= 1 ; i<=5; i++){ m = n * i; printf ("Le mumtiple %d est : %d \n", i, m) ; t[i] = m ; som = som + m; } moy = som/5 ; printf ("La moyenne de ces multiples est: %6.2f \n", moy) ; return 0; } </pre>  <pre>Entrer un entier n : 4 Le mumtiple 1 est : 4 Le mumtiple 2 est : 8 Le mumtiple 3 est : 12 Le mumtiple 4 est : 16 Le mumtiple 5 est : 20 La moyenne de ces mumtiples est: 12.00</pre>

عربية	يؤخذ بعين الاعتبار أحد الإثنتين : الـغوريتم أو برنامج C
Français	Il est pris en considération une des deux formes : l'algorithme ou le programme C
English	Only one of the two is considered : the algorithm OR the C program

En langage Algorithmique

Algo FivePoints

Type Point = Enregistrement

x: réel;

y: réel;

FinEnregistrement

Var

PP : Point // Le point particulier

T : tableau [1..5] de Point

i : entier

imin : entier // Le numéro du point le plus proche du point particulier pp

di : réel // La distance entre un point et pp

dmin : réel // La distance minimale

Début

// Lecture des coordonnées du point particulier (coordonnées introduites par l'utilisateur)

Ecrire ('entrer le x du point particulier ')

Lire (pp.x)

Ecrire ('entrer le y du point particulier ')

Lire (pp.y)

/*

On peut initialiser directement les coordonnées du point particulier lors de sa déclaration comme suit :

pp : Point = {15 , 12}

Ou bien champ par champ (par des simples affectations dans le programme principal) comme suit :

pp.x ← 15

pp.y ← 12

*/

// Lecture des coordonnées des 5 points

Pour (i de 1 à 5) faire

Ecrire ('entrer le x du point numéro ', i)

Lire (T[i].x)

Ecrire ('entrer le y du point numéro ', i)

Lire (T[i].y)

FinPour

/*

On peut initialiser directement les coordonnées des 5 points lors de la déclaration du tableau comme suit :

T : tableau [1..5] de Point = {2.5, 4.5, -4, 13, 11, -5, 8.4, 9, 6.7, -2.9}

Ou bien une par une (dans le programme principal) comme suit

T[1].x ← 2.5 ; T[1].y ← 4.5 ; T[2].x ← -4 ; T[2].y ← 13 ; T[3].x ← 11 ; T[3].y ← -5 ;

T[4].x ← 8.4 ; T[4].y ← 9 ; T[5].x ← 6.7 ; T[5].y ← -2.9 ;

*/

// calcul de la distance minimale

// Au début, on suppose que la distance minimale est celle du premier point (le premier point est le plus proche)

dmin ← sqrt ((T[0].x - pp.x) * (T[0].x - pp.x) + (T[0].y - pp.y) * (T[0].y - pp.y))

imin ← 0

Pour (i de 2 à 5) faire

di = sqrt ((T[i].x - pp.x) * (T[i].x - pp.x) + (T[i].y - pp.y) * (T[i].y - pp.y))

Si (di < dmin) alors

dmin ← di // changer la distance minimale

imin ← i // changer l'indice du point le plus proche.

FinSi

FinPour

Ecrire ('le point la plus proche est le point numéro ', imin , ' situé en (', T[imin].x , ', ', T[imin].y , ', ')

Ecrire (' , avec une distance de ', dmin)

Fin

C language

```
#include<stdio.h>
#include<math.h>
int main(){
typedef struct{
    float x;
    float y;
}Point;
int i ;
int imin ;
float di ;
float dmin ;
Point pp ;
Point T[5] ;
// introduire les coordonnés du point particulier pp
printf ("entrer le x du point paticulier ") ;
scanf ("%f" , &pp.x) ;
printf ("entrer le y du point paticulier ") ;
scanf ("%f" , &pp.y) ;
/*
On peut initialiser directement les coordonnés du point particulier lors de sa déclaration comme suit :
Point pp = {15, 12} ;
Ou bien champ par champ comme suit :
pp.x = 15 ;
pp.y = 12;
*/
// introduire les coordonnés des 5 points
for(i=0 ; i<5; i++) {
    printf ("entrer le x du point numero %d " , i ) ;
    scanf ("%f" , &T[i].x) ;
    printf ("entrer le y du point numero %d " , i ) ;
    scanf ("%f" , &T[i].y) ;
}
/*
On peut initialiser directement les coordonnés des 5 points lors de la déclaration du tableau comme suit :
Point T[5]= {2.5, 4.5,-4, 13, 11, -5, 8.4, 9, 6.7,-2.9};
Ou bien une par une comme suit :
T[0].x = 2.5 ; T[0].y = 4.5 ; T[1].x = -4; T[1].y = 13 ; T[2].x = 11 ; T[2].y = -5 ;
T[3].x = 8.4 ; T[3].y = 9 ; T[4].x = 6.7 ; T[4].y = -2.9 ;
*/
// calculer la distance minimale
// au début, on suppose que le point le plus proche est le premier point
dmin = sqrt((T[0].x - pp.x) * (T[0].x - pp.x) + (T[0].y - pp.y) * (T[0].y - pp.y));
imin = 0 ;
for (i=1; i<5; i++){
    di=sqrt((T[i].x - pp.x) * (T[i].x - pp.x) + (T[i].y - pp.y) * (T[i].y - pp.y));
    if(di<dmin){
        dmin=di;
        imin=i;
    }
}
printf("\n i= %d di=%f dmin= %f imin=%d ",i,di, dmin,imin);
}
printf("\n le point le plus proche est le point numero %d situé en (%.2f , %.2f ) avec une distance de %.2f " , imin, T[imin].x ,
T[imin].y , dmin);
}
// Exécution page suivante :
```

Exécution :

```

entrer le x du point paticulier 12.6
entrer le y du point paticulier -11.8
entrer le x du point numero 0 20.5
entrer le y du point numero 0 31.4
entrer le x du point numero 1 -42.7
entrer le y du point numero 1 78.9
entrer le x du point numero 2 17.2
entrer le y du point numero 2 15.4
entrer le x du point numero 3 10.0
entrer le y du point numero 3 -9.9
entrer le x du point numero 4 19.0
entrer le y du point numero 4 -4.4

i= 1 di=106.228912 dmin= 43.916401 imin=0
i= 2 di=27.586229 dmin= 27.586229 imin=2
i= 3 di=3.220249 dmin= 3.220249 imin=3
i= 4 di=9.783660 dmin= 3.220249 imin=3
le point le plus proche est le point numero 3 situo en (10.00 , -9.90 ) avec une distance de 3.22
    
```

عربية	يؤخذ بعين الإعتبار أحد الإثنيتين : الغوربتم أو برنامج C
Français	Il est pris en considération une des deux formes : l'algorithme ou Le programme C
English	Only one of the two is considered : the algorithm OR the C program