

U.E Informatique III INF 212
Examen Janvier 2021: Durée 2 heures
(Documents et calculatrices interdits)

Exercice I : Programmes en C (7pts)

Donner en illustrant les résultats des programmes suivants :

<p>1. 0.5pt</p> <pre>main () { unsigned int y = 12; int x = -2; if (x > y) printf("R1: %d\n", x); else printf("R1: %d\n", y); }</pre>	<p>2. 0.5pt</p> <pre>main () { int i; printf("R2: "); for (i = 1; i <= 3; i++) { printf("%d\t", i); } }</pre>
<p>3. 0.5pt</p> <pre>main () { float h = 10.0; printf("R3: %f\n", 1/2*h); }</pre>	<p>4. 8*0.5pts</p> <pre>#include <stdlib.h> int a = 2; int b = 3; int f1 (int i) {return (++i);} int f2 (int i) {return (i++);} int f3 (int *i) {return ((*i)++);} int f4 (int a) {return (a ++ + ++b);} int f5 (int *a, int c) { int d = 1; int e = 0; return ((*a) ++ + c + d ++); }</pre>
<p>5. 0.5pt</p> <pre>main () { int i = 2, j; j = i++ > 1 ++i > 2; printf("R5: %d\n", i); }</pre>	<p>6. 1pt</p> <pre>int rec (int n, int p) { if (!p) return 1; else if (!(p%2)) return (rec(n*n, p/2)); else return (n*rec(n*n, (p-1)/2)); } main () { printf("R6: %d\n", rec(2, 8)); }</pre>

Exercice 3 : (13 Points)

On considère les programmes (a), (b) et (c) ci-dessous.

1. Expliquer de façon concise et précise ce que fait le programme (a).
2. Expliquer de façon concise et précise ce que fait le programme (b).
3. En tant que programmeur, donner une différence entre les programmes (a) et (b).

2pts
1pts
1pt

4. Expliquer de façon concise et précise ce que fait le programme (c). 2pt
5. Le programme (c) fait intervenir une notion inexistante dans le programme (b), laquelle ? 1pt
6. Recopier le programme (b) sur votre copie en ajoutant un commentaire claire à chaque ligne du programme principal. (Un bon commentaire se doit d'être concis et précis) 0,5pt * 10
7. Donner un exemple de résultats que peut renvoyer le programme (c). 1pt

Programme (a)

```
#include <stdio.h>
#define NVAL 5 /* NVAL est donc une constante égale à 5 */
main()
{ int i, min, max ;
  int t[NVAL] ;
  printf ("donnez %d valeurs\n", NVAL) ;
  for (i=0 ; i<NVAL ; i++){ scanf ("%d", &t[i]) ;}
  max = min = t[0] ;
  for (i=1 ; i<NVAL ; i++)
    { if (t[i] > max) max = t[i] ;
      if (t[i] < min) min = t[i] ;
    }
  printf ("valeur max : %d\n", max) ;
  printf ("valeur min : %d\n", min) ;
}
```

Programme (b)

```
#include <stdio.h>
#define NVAL 5
main()
{ int i, min, max ;
  int t[NVAL] ;
  printf ("donnez %d valeurs\n", NVAL) ;
  for (i=0 ; i<NVAL ; i++) {scanf ("%d", &t[i]) ;}
  max = min = *t ;
  for (i=1 ; i<NVAL ; i++)
    { if (*(t+i) > max) max = *(t+i) ;
      if (*(t+i) < min) min = *(t+i) ;
    }
  printf ("valeur max : %d\n", max) ;
  printf ("valeur min : %d\n", min) ;
}
```

Programme (c) :

```
#include <stdio.h>
#define NPOINTS 3
main()
{ struct point { int num ;
                float x ;
                float y ;
              } ;
  struct point courbe[NPOINTS] ;
  int i ;
  for (i=0 ; i<NPOINTS ; i++)
    { printf ("numéro : ") ; scanf ("%d", &courbe[i].num) ;
      printf ("x : ") ; scanf ("%f", &courbe[i].x) ;
      printf ("y : ") ; scanf ("%f", &courbe[i].y) ;
    }
  printf (" **** structure fournie ****\n") ;
  for (i=0 ; i<NPOINTS ; i++){
    printf ("numéro : %d  x : %f  y : %f\n", courbe[i].num, courbe[i].x, courbe[i].y) ;
  }
}
```

SN Informatique

Exercice 1

Résultats des programmes:

- 1- R1: 12 car $12 > -2$
- 2- R2: 1 2 3 car i prend les valeurs 1, 2 et 3 dans "for".
- 3- R3: 5.0 car $1/2 * 10.0 = 0.5 * 10.0 = 5.0$ (* et / ont la même priorité)
- 4- R4.1: 1 car $f_1(e) = ++e = ++0 = 1$ (la valeur de e reste 0)
- R4.2: 0 car $f_2(e) = e++ = 0++$ ($f_2(e)$ prend la valeur de e et e est incrémentée)
- R4.3: 0 car $f_3(e) = *(&e)++ = e++$ ($f_3(e)$ prend la valeur de e et e est incrémentée)
- R4.4: 1 car $f_3(e) = *(&e)++ = e++$ ($f_3(e)$ prend la valeur de e et e est incrémentée)
- R4.5: 4 car $f_4(e) = e++ + ++b$
 $= 0 + 4$
 $= 4$ ($f_4(e)$ prend la valeur de e + b + 1 et e est incrémentée)
- R4.6: 5 car $f_4(e) = e++ + ++b$
 $= 1 + 4$
 $= 5$ ($f_4(e)$ prend la valeur de e + b + 1 et e est incrémentée)
- R4.7: 3 car $f_5(&a, e) = a++ + e + d++$
 $= 2 + 0 + 1 = 3$ ($f_5(&a, e)$ prend la valeur de a + e + d et a et d sont incrémentées)
- R4.8: 4 car $f_5(&e, a) = e++ + a + d++$
 $= 0 + 3 + 1 = 4$ ($f_5(&e, a)$ prend la valeur de e + a + d (d valant 1) et e et d sont incrémentées)

5- R5: 3 car;

- On teste d'abord $i > 1$ (vrai) et on incrémente i à 3.
- On ne se préoccupe pas de $++i > 2$ à cause de l'opérateur $||$ (ou) (q vaut 1)
- i vaut donc 3

6- R6: 256 car;

- $\text{rec}(2, 8) = \text{rec}(2 \times 2, 8/2)$ car $!(8 \% 2) = !0 = 1$
 $= \text{rec}(4, 4)$
 $= \text{rec}(4 \times 4, 4/2)$ car $!(4 \% 2) = !0 = 1$
 $= \text{rec}(16, 2)$
 $= \text{rec}(16 \times 16, 2/2)$ car $!(2 \% 2) = !0 = 1$
 $= \text{rec}(256, 1)$
 ~~$= \text{rec}(256, 1)$~~
 $= 256 \times \text{rec}(256 \times 256, (1-1)/2)$ car $!(1 \% 2) = !1 = 0$
 $= 256 \times \text{rec}(65536, 0)$
 $= 256 \times 1$ car $!0 = 1$
 $= 256.$

Exercice 2

1- Le programme (a) permet de remplir un tableau d'entiers de longueur $NVAL (= 5)$, de déterminer et d'afficher sa plus grande et sa plus petite valeur.

2- Le programme (b) permet de remplir un tableau d'entiers de longueur $NVAL (= 5)$ à l'aide d'un pointeur, de déterminer et d'afficher sa plus grande et sa plus petite valeur.

3- Le programme (b) contrairement au programme (a) utilise un pointeur pour remplir le tableau ce qui permet de créer une adresse pour un élément du tableau au fur et à mesure qu'il est rempli au lieu de donner des emplacements prédéfinis (meilleure gestion de l'espace mémoire)

4- Le programme (c) permet de remplir progressivement le tableau courbe de type point et de longueur NPOINTS et ensuite d'afficher son contenu.

5- Il fait intervenir la notion d'enregistrement ou structure (le type point)

6-

Lignes	Contenu	Commentaire
1	int i, min, max;	Déclaration des entiers i, min, max
2	int t[NVAL]	Déclare le tableau t de longueur NVAL
3	printf("donnez..."); NVAL);	// Demande de remplir le tableau t
4	for(...) { scanf(...); }	// lit les valeurs progressivement en-tiers.
5	max = min = *t	// initialise max et min à t[0]
6	for(i=1; i<NVAL; i++)	// parcourt le tableau
7	{ if(*t+i > max) max = *(t+i);	// détermine le plus grand élément de t
8	if(*t+i < min) min = *(t+i);	// détermine le plus petit élément de t.
9	printf("Valeur max...");	// affiche la plus grande valeur
10	printf("Valeur min...");	// affiche la plus petite valeur

7- Exemple : Donnez des valeurs arbitraires aux champs des point