

NAME/NOM	COLLEGE NUMBER NUMÉRO DE COLLÈGE
----------	-------------------------------------

Mid-Term Test Theoretical Part

EEE243 Applied Computer Programming
17 October 2017, 0800 – 1050 hrs

Examiner: Dr. Tashfeen Karamat, PhD

Instructions:

- **Do not turn this page until instructed to do so.**
- You have 170 minutes to complete both parts of the exam.
- This exam has two parts.
 - The *theoretical* portion of the test is closed-book, out of a total of 30 marks.
 - The *practical* portion of the test is open-book (you may have course notes and textbook but you may **NOT** access the internet), and is worth 60 marks.
- There are 20 multiple choice questions, 20 True/False questions each worth 0.5 marks
- The 3 short answer questions are worth the points indicated in the center column.
- You must hand in the *theoretical* exam before you can begin the *practical* portion of the exam.
- Answer all the questions directly on the questionnaire.
- Immediately fill out your name, college number and the number of your workstation on the information card.
- Write your name on the questionnaire
- If a question seems unclear, make a *reasonable* assumption, document it, and answer the question as though the assumption were correct. *The examiners will not clarify the meaning of questions during the test.*
- Good luck!

Test de mi-session Partie théorique

GEF243 Programmation informatique appliquée
17 octobre 2017, 08h00–10h50

Examineur: Capt Adrien Lapointe, CD, MSc

Instructions:

- **Ne tournez pas cette page avant l'instruction de l'examineur.**
- Vous avez 170 minutes pour compléter les deux parties de l'examen.
- Cet examen a deux parties.
 - La partie *théorique* est à livre fermé et compte pour 30 points.
 - La partie *pratique* est à livre ouvert (vous avez droit à vos notes, à votre livre mais vous ne pouvez **PAS** accéder à l'Internet), et compte pour un total de 60 points.
- Il y a 20 questions à choix de réponse et 20 questions Vrai/Faux valant chacune 0,5 points.
- Les 3 questions à réponse courtes ont les valeurs indiquées dans la colonne centrale.
- Vous devez remettre l'examen *théorique* avant de pouvoir commencer la partie *pratique*.
- Répondez à toutes les questions directement sur le questionnaire d'examen.
- Écrivez immédiatement votre nom, numéro de collège et numéro de station de travail sur la carte d'information.
- Écrivez votre nom sur le questionnaire.
- Si une question ne vous semble pas claire, faites des suppositions raisonnables, documentez-les et répondez à la question en tenant compte des suppositions. *Les examinateurs ne clarifieront pas le sens des questions pendant le test.*
- Bonne chance!

MULTIPLE CHOICE QUESTIONS / QUESTION À CHOIX DE RÉPONSE

1. Which of the following statements is NOT a valid declaration?
 - a. `int x=10;`
 - b. `int x, int y;`
 - c. `float a_float;`
 - d. All choices are correct.
2. The statement `x++` increments variable `x` by:
 - a. 1
 - b. 2
 - c. 3
 - d. None of the choices is correct.
3. Infinite for loop can be written as:
 - a. `for(1)`
 - b. `for(i=0;i<∞;i++)`
 - c. `for(; ;)`
 - d. `for(infinite)`
4. Which of the following is the right way to get input from user:
 - a. `scanf("%d",num);`
 - b. `scanf("num",%d);`
 - c. `scanf("%d",&num);`
 - d. `scanf("&d", %num);`
5. Which of the following is a valid declaration of a function which takes a two dimensional array as an argument:
 - a. `void hello(int table[][5]);`
 - b. `void hello(int table[][]);`
 - c. `void hello(int table[]);`
 - d. `void hello(int table[5]);`
1. Lequel des énoncés suivants n'est pas une déclaration valide ?
 - a. `int x=10;`
 - b. `int x, int y;`
 - c. `float a_float;`
 - d. Tous les choix sont corrects.
2. L'énoncé `x++` augmente la valeur de la variable `x` de :
 - a. 1
 - b. 2
 - c. 3
 - d. Aucun des choix n'est correct.
3. Les boucles sans fin avec `for` sont écrites comme :
 - a. `for(1)`
 - b. `for(i=0;i<∞;i++)`
 - c. `for(; ;)`
 - d. `for(infinite)`
4. Lequel des énoncés suivant est la bonne façon de demander l'entrée de l'utilisateur :
 - a. `scanf("%d",num);`
 - b. `scanf("num",%d);`
 - c. `scanf("%d",&num);`
 - d. `scanf("&d", %num);`
5. Lequel des énoncés suivants est une déclaration valide pour une fonction qui prend en paramètre un tableau à deux dimensions :
 - a. `void hello(int table[][5]);`
 - b. `void hello(int table[][]);`
 - c. `void hello(int table[]);`
 - d. `void hello(int table[5]);`

6. If function is declared as

```
float square(float num);
```

What is the correct way to call it in main function:

- a. square (102);
- b. float square(num);
- c. square(float num);
- d. float square(float num);

7. Which of the following is not a valid declaration of an array in C:

- a. int scores[]={3, 4, 44, 54};
- b. char first_names[25];
- c. int grades[];
- d. float averages[12];

8. The compiler will check for out of bound problems only during:

- a. initialization of the array
- b. declaration of the array
- c. first use of the array
- d. printing of an array

9. When passing an array to a function:

- a. the values of the arrays are passed
- b. address of the values of the all array elements are passed
- c. address of first element of the array is passed
- d. none of the choices is correct

10. Which of the following is not a valid way to initialize a two dimensional array

- a. int num_table[2][3] = {2,5,8,1,9,11};
- b. int num_table[2][3] = {
 {2,5,8},
 {1,9,11}
 };
- c. int num_table[][3] = {
 {2,5,8},
 {1,9,11}
 };
- d. All choices are correct

6. Si une fonction est déclarée comme

```
float square(float num);
```

Quelle est la façon correcte de l'appeler dans la fonction main :

- a. square (102);
- b. float square(num);
- c. square(float num);
- d. float square(float num);

7. Laquelle des déclarations de tableau suivantes n'est pas valide en C :

- a. int scores[]={3, 4, 44, 54};
- b. char first_names[25];
- c. int grades[];
- d. float averages[12];

8. Le compilateur vérifie les limites d'un tableau seulement à :

- a. l'initialisation du tableau
- b. la déclaration du tableau
- c. la première utilisation du tableau
- d. l'affichage du tableau

9. Lorsqu'on passe un tableau à une fonction :

- a. les valeurs contenues dans les tableaux sont aussi passées
- b. Les adresses des valeurs de tous les éléments du tableau sont passées
- c. L'adresse du premier élément du tableau est passée
- d. Aucun des choix n'est correct.

10. Lequel des énoncés suivants n'est pas une façon valide d'initialiser un tableau à deux dimensions :

- a. int num_table[2][3]= {2,5,8,1,9,11};
- b. int num_table[2][3] = {
 {2,5,8},
 {1,9,11}
 };
- c. int num_table[][3] = {
 {2,5,8},
 {1,9,11}
 };
- d. Tous les choix sont corrects

11. Which of the following is not a correct way of initializing strings:

- a. `char str[9] = "Good Day";`
- b. `char str[9] = {'G','o','o','d',' ',' ','D','a','y','\0'};`
- c. `char str{}="hello";`
- d. `char month[] = "January";`

12. Which one of the following is a valid way of printing a string contained in an array called `str`:

- a. `printf("Message: %s\n", str);`
- b. `printf("Message: %c\n", str);`
- c. `printf("Message: %s\n", str[]);`
- d. `printf("Message: str\n", %c);`

13. Which of the following creates the final executable file:

- a. compiler
- b. pre-processor
- c. linker
- d. assembler

14. After execution of the following code, what value will variable `b` contain

```
-----  
int a = 6;  
int b = 10;  
int *p = NULL;  
int *r = NULL;  
p = &a;  
r = p;  
b = *r;  
-----
```

- a. 6
- b. 10
- c. Address of `r`
- d. Address of `p`

15. We can pass parameters by reference to function using:

- a. Arrays
- b. Pointers
- c. `Malloc()`
- d. Both a and b

11. Lequel des énoncés suivants n'est pas une façon correcte d'initialiser une chaîne de caractères :

- a. `char str[9] = "Good Day";`
- b. `char str[9] = {'G','o','o','d',' ',' ','D','a','y','\0'};`
- c. `char str{}="hello";`
- d. `char month[] = "January";`

12. Lequel des énoncés suivants est une façon correcte d'afficher la chaîne de caractères contenue dans le tableau `str` :

- a. `printf("Message: %s\n", str);`
- b. `printf("Message: %c\n", str);`
- c. `printf("Message: %s\n", str[]);`
- d. `printf("Message: str\n", %c);`

13. Lequel des énoncés suivants crée le fichier exécutable final :

- a. compilateur
- b. pré-compilateur
- c. relieur
- d. assembleur

14. Après l'exécution du code suivant, quelle sera la valeur contenue dans `b` :

```
-----  
int a = 6;  
int b = 10;  
int *p = NULL;  
int *r = NULL;  
p = &a;  
r = p;  
b = *r;  
-----
```

- a. 6
- b. 10
- c. Adresse de `r`
- d. Adresse de `p`

15. Nous pouvons passer un paramètre par référence en utilisant :

- a. Tableaux
- b. Pointeurs
- c. `Malloc()`
- d. a et b

16. Following code moves ahead the pointer p by

```
-----  
char x[] = "hello exam";  
char *p = x;  
p++;  
-----
```

- a. One byte
- b. Two bytes
- c. Three bytes
- d. Four bytes

17. Given the following code, which of the following statements is meaningless:

```
-----  
int *p1;  
char *p2;  
int *p3;  
-----
```

- a. p1--;
- b. p2+5;
- c. p1-p2;
- d. p1-p3;

18. Global variables reside in following segment of memory:

- a. Stack
- b. Heap
- c. Data
- d. Function

19. Which of the following statement is valid about the malloc() function:

- a. It allocates the memory in 'data' segment
- b. It allocates the memory in 'stack' segment
- c. It allocates the memory in 'code' segment
- d. It allocates memory in 'heap' segment

20. Which of the following is NOT a valid category of constant in C:

- a. Literal constant
- b. Universal constant
- c. Defined constant
- d. Memory constant

16. Le code suivant déplace le pointeur p de

```
-----  
char x[] = "hello exam";  
char *p = x;  
p++;  
-----
```

- a. Un octet
- b. Deux octets
- c. Trois octets
- d. Quatre octets

17. Étant donné le code suivant, lequel des énoncés est incongru :

```
-----  
int *p1;  
char *p2;  
int *p3;  
-----
```

- a. p1--;
- b. p2+5;
- c. p1-p2;
- d. p1-p3;

18. Les variables globales résident dans le segment de mémoire suivant :

- a. Pile
- b. Tas
- c. Données
- d. Fonction

19. Lequel des énoncés suivants au sujet de malloc() est correct :

- a. Il alloue la mémoire dans le segment des données
- b. Il alloue la mémoire dans le segment de la pile
- c. Il alloue la mémoire dans le segment du code
- d. Il alloue la mémoire dans le segment du tas

20. Laquelle des catégories de constantes n'est PAS valide en C :

- a. Constante littérale
- b. Constante universelle
- c. Constante définie
- d. Constante mémoire

TRUE/FALSE QUESTIONS / QUESTIONS VRAI OU FAUX

- | | | |
|---|--------|--|
| 21. You cannot initialize variables at the time of declaration. | T
F | 21. Vous ne pouvez pas initialiser les variables lors de leur déclaration. |
| 22. Global and static variables have same scope and extent. | T
F | 22. Les variables globales et statiques ont la même portée et durée. |
| 23. In C, A=B verifies that A is equal to B. | T
F | 23. En C, A=B vérifie que A est égal à B. |
| 24. & is a bitwise operator. | T
F | 24. & est un opérateur bit par bit |
| 25. You can declare loop variable inside the loop or outside the loop. | T
F | 25. Vous pouvez déclarer les variables de boucle à l'intérieur ou à l'extérieur de la boucle. |
| 26. You could technically write all your code in the main function. | T
F | 26. Techniquement, vous pourriez écrire tout votre code dans la fonction main. |
| 27. The name of the array is a symbolic reference to the first byte in the array. | T
F | 27. Le nom d'un tableau est une référence symbolique au premier octet du tableau. |
| 28. The division 4/5 will give you 0.8. | T
F | 28. La division 4/5 donne 0,8. |
| 29. The compiler will complain (with a warning or error) if you use an <i>out of bounds</i> index for your array. | T
F | 29. Le compilateur donne un avertissement ou une erreur pour les indices <i>hors limites</i> d'un tableau. |
| 30. When passing a two-dimensional array to a function, you can omit the first dimension. | T
F | 30. Lorsque vous passez un tableau à deux dimensions à une fonction, vous pouvez omettre la première dimension. |
| 31. Strings can be declared as an array or a pointer. | T
F | 31. Les chaînes de caractères peuvent être déclarées comme un tableau ou comme un pointeur. |
| 32. \0 is used as a logical marker in C to tell our functions the START of the string. | T
F | 32. \0 est un marqueur logique utilisé en C pour indiquer à nos fonctions le DÉBUT d'une chaîne de caractères. |
| 33. The size of logical storage of the strings could be different than the physical storage size. | T
F | 33. La taille du stockage logique d'une chaîne de caractères peut être différente de celle du stockage physique. |
| 34. Before a code is compiled, it goes through a 'Preprocessor' | T
F | 34. Avant que le code ne soit compilé, il passe dans un « pré-compilateur » |

- | | | |
|---|----------------------|--|
| 35. Name of an array is a constant pointer. | T
F | 35. Le nom d'un tableau est un pointeur constant. |
| 36. We cannot specify a size when declaring a string as pointer | T
F | 36. Nous ne pouvons pas spécifier la taille lorsque nous déclarons une chaîne de caractères comme un pointeur. |
| 37. The <code>malloc()</code> function returns any types of pointer as specified in its parameters. | T
F | 37. La fonction <code>malloc()</code> retourne n'importe quel type spécifié dans ses paramètres. |
| 38. It is not possible to have a pointer to a pointer. | T
F | 38. Il n'est pas possible d'avoir un pointeur à un pointeur. |
| 39. When using <code>malloc()</code> for memory allocation, the compiler will warn you if there is not enough memory. | T
F | 39. Lorsque <code>malloc()</code> est utilisé pour allouer de la mémoire, le compilateur nous averti d'un manque de mémoire. |
| 40. When a dynamically allocated memory is not needed, it is freed automatically | T
F | 40. Lorsque la mémoire alloué dynamiquement n'est plus nécessaire, elle est libérée automatiquement. |

SHORT ANSWER QUESTIONS / QUESTIONS À RÉPONSE COURTE

Refer to the following code for **Questions 41 through 43.**

Référez-vous au code suivant pour les **questions 41 à 43.**

```
#include <stdio.h>

void something(float *a, float *b, float c) {
    c = *b;
    *a = *a * c;
    *b = *a * c;
}

int main(void) {
    float a = 0.1;
    float b = 4.0;
    float c = 2.5;
    something(&a, &b, c);
    printf("a = %0.2f, b = %0.3f, c = %0.2f\n", a, b, c);
}
```

- | | | |
|---|----------|--|
| 41. Some of the parameters to the function <code>something</code> are passed <i>by value</i> and some are passed <i>by reference</i> . Briefly explain what these terms mean. | 4 | 41. Certains paramètres de la fonction <code>something</code> sont passés <i>par valeur</i> alors que d'autres le sont <i>par référence</i> . Expliquez brièvement ce que ces termes signifient. |
|---|----------|--|

42. Indicate which parameters are being passed by each method.

3 42. Indiquez par quelle méthode les paramètres sont passés.

43. What would the `main` function print?

3 43. Qu'imprimerait la fonction `main` ?

End of THEORETICAL Portion of Test

Fin de la partie THÉORIQUE du test