

Mid-Term Test Practical Portion

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

Examiner: Tashfeen Karamat, PhD

Instructions

- You may begin the practical portion of the exam immediately.
- The practical portion is out of 60 marks.
- The practical portion of this test is open book. You may refer to course notes and you may **NOT** access the Internet.
- Communication with any person other than the invigilators during the practical portion of the test **is forbidden**.
- Ask the invigilator for notepaper if you require it.

Hint: Compile and execute your program **often**. You will lose marks if your program does not compile and execute as submitted, or if it generates compiler warnings. Start small and build from there.

Test de mi-session Partie Pratique

GEF243 Programmation informatique appliquée
17 octobre 2017, 8h00–10h50

Examineur: Capt Adrien Lapointe, CD, MSc

Instructions

- Vous pouvez commencer la partie pratique tout de suite.
- La valeur de la partie pratique est de 60 points.
- La partie pratique de l'examen est à livre ouvert. Vous pouvez faire référence aux notes du cours, mais ne pouvez **PAS** accéder à l'Internet.
- La communication avec toutes personnes autre que les surveillants pendant la partie pratique de l'examen **est interdite**.
- Demandez des feuilles de notes si vous en avez besoins.

Indice : Compilez et exécutez votre code **souvent**. Vous serez pénalisé si votre programme ne compile pas, n'exécute pas ou génère des avertissements de compilateur. Commencez petit et ajoutez des fonctionnalités au fur et à mesure.

Introduction

Open the Midterm project that you created at the beginning of the examination.

Add your name and college number to the comment blocks at the beginnings of the `student.c` and `dterm_2017.c` files.

Ensure that you respect the interfaces that are provided to you in the header files. We will mark the exam using automated tests. Therefore any modifications you make to the function prototypes will break those tests and we will not be able to give you any marks.

Make sure your code compiles.

Requirements

You have been provided with six files: `student.c`, `student.h`, `midterm_2017.c`, `support.c`, `support.h` and `sphere.h`.

In `student.c` you must implement the functions outlined in the four questions.

You **must not modify** `student.h`, `support.c`, `support.h` and `sphere.h`.

You can modify `midterm_2017.c` but it will not be marked. It is where you can run the functions you implemented. You will notice that some code is commented out in that file. You may use that code and modify it for your tests.

Question 1

We provide you with a function, `interface_calc()`, that asks the user to choose what function to execute (addition, difference, multiplication, division) and with what parameters.

Write code in the function `calculator()` so that it returns the proper result based on the user inputs.

Use the following function prototype that is provided in your starting code.

```
float calculator(int choice, float a, float b);
```

Introduction

Ouvrez le projet Midterm que vous avez créé au début de l'examen.

Ajoutez votre nom et numéro de collège au bloc commentaire au début des fichiers `student.c` et `midterm_2017.c`.

Assurez-vous de respecter les interfaces qui vous sont fournies dans les fichiers d'en-tête. Nous corrigerons l'examen en utilisant des tests automatisés. Ainsi, toutes modifications aux prototypes de fonction briseront les tests et nous ne pourrons pas vous donner de points.

Assurez-vous que votre code compile.

Exigences

Six fichiers vous sont fournis : `student.c`, `student.h`, `midterm_2017.c`, `support.c`, `support.h` et `sphere.h`.

Dans `student.c`, vous devez implémenter les fonctions indiquées dans les quatre questions.

Vous **ne devez pas modifier** `student.h`, `support.c`, `support.h` et `sphere.h`.

Vous pouvez modifier `midterm_2017.c`, mais le fichier ne sera pas corrigé. C'est l'endroit où vous pouvez appeler les fonctions que vous avez créées. Vous remarquerez que du code a été mis en commentaires dans ce fichier. Vous pouvez utiliser ce code et le modifier pour vos tests.

10 Question 1

Nous vous donnons une fonction, `interface_calc()`, qui demande à l'utilisateur de choisir quelle fonction exécuter (addition, difference, multiplication, division) et avec quels paramètres.

Écrivez du code dans la fonction `calculator()` afin de retourner le résultat approprié selon les entrées de l'utilisateur.

Utilisez le prototype de fonction suivant qui vous est fourni dans le code de départ.

Question 2

The Taylor Series Expansion of a sine function is given by:

$$\sin(x) = \sum_{i=0}^n \frac{(-1)^i}{(2i+1)!} x^{2i+1}$$

Where x is given in radians.

You are asked to implement the sine function in C for any angle expressed in degrees.

Use 3.14159 as the value for π . Remember that 180 degrees is π radians.

Implement the function to convert from degrees to radians, the power function $x^y = \text{power}(x, y)$, the factorial function $x! = \text{factorial}(x)$. Finally, using the functions you implemented, write the `sinus` function that takes an angle in degrees as a series with n terms as described above.

The functions should have the following prototypes, which are provided in your starting code.

You are not allowed to use the functions in `math.h` for implementing your functions. You may use them to confirm that your results are correct.

Hint: Factorial of 0 is 1. Factorial of 3 is $1 \times 1 \times 2 \times 3$.

- a. `double deg_to_rad(double deg);` (3)
- b. `double power(double x, double y);` (4)
- c. `double factorial(int x);` (4)
- d. `double sinus(double x, int n);` (5)

Question 3

Implement a function that takes a matrix $n \times m$ and returns the sum of each row in a $n \times 1$ matrix.

$$M = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

$$\text{row_adder}(M) = \begin{bmatrix} 6 \\ 15 \\ 24 \end{bmatrix}$$

16 Question 2

Le développement en série de Taylor d'une fonction sinusoïdale est donné par:

Où x est donné en radians.

Vous devez implémenter la fonction sinusoïdale en C pour n'importe quel angle donné en degrés.

Utilisez la valeur 3,14159 pour π . Souvenez-vous que 180 degrés équivaut à π radians.

Implémentez la fonction pour convertir les degrés en radians, la fonction de puissance $x^y = \text{power}(x, y)$, la fonction factoriel $x! = \text{factorial}(x)$. Enfin, en utilisant les fonctions que vous avez implémentées, écrivez la fonction `sinus` prenant un angle en degrés comme une série à n termes tel que décrit ci-dessus.

Les fonctions devraient avoir les prototypes suivants qui vous sont fournis dans le code de départ.

Vous ne pouvez pas utiliser les fonctions de `math.h` pour implémenter vos fonctions. Vous pouvez toutefois les utiliser pour confirmer vos résultats.

Indice : La factoriel de 0 est 1. La factoriel de 3 est $1 \times 1 \times 2 \times 3$.

- a. `double deg_to_rad(double deg);`
- b. `double power(double x, double y);`
- c. `double factorial(int x);`
- d. `double sinus(double x, int n);`

15 Question 3

Implémentez une fonction qui prend une matrice $n \times m$ et retourne la somme de chacune de ses rangées dans une matrice $n \times 1$.

Your function must have the following prototype that is provided in your starting code.

```
int *row_adder(int matrix[][WIDTH], int height);
```

Note that the width of the matrix has to be specified. While the current value is set to 3, we expect your function to be general enough to handle different widths.

You may use the function `display_row_matrix` that is provided to you in `support.c`.

Hint: You need to use `malloc()` in your code.

Question 4

We give you the representation of an image as an array of integers in `sphere.h`.

If you were to display this array in the console, you would have a hard time to see the sphere that it represents.

Early computers used special characters to represent images in the console. We will use such a strategy here using ASCII characters to display the image in the console. This is often referred to as ASCII art.

The image is in greyscale where 0 represents completely black and 255 completely white with values in between representing various levels of grey.

The string `GREYSCALE` defined in `student.h` gives you 10 characters that can be used to represent those levels of grey. Using this ASCII scale, the space character ' ' is white and the '@' character is black.

- a. Write a function to convert the image into an array of characters that uses the provided scale. Use the following function prototype that is provided in your starting code.

The mapping should be such that each range on the 0 to 255 scale is pretty much divided equally. For instance, if you picked a range size of 12 (don't), the space character would be mapped to the values 243 to 255.

```
void convert_to_ascii(int image[][IMG_SIZE], char ascii[][IMG_SIZE]);
```

Utilisez le prototype de fonction suivant qui vous est fourni dans le code de départ.

Notez que la largeur de votre matrice doit être spécifiée. Bien que la valeur soit présentement de 3, votre fonction devrait être assez générale pour prendre d'autres largeurs.

Vous pouvez utiliser la fonction `display_row_matrix` qui se trouve dans `support.c`.

Indice : vous devez utiliser `malloc()` dans votre code.

19 Question 4

Nous vous donnons la représentation d'une image comme un tableau d'entiers dans `sphere.h`.

Si vous affichiez le tableau sur la console, vous auriez de la difficulté à voir la sphère qu'il représente.

Les ordinateurs des débuts de l'informatique utilisaient des caractères spéciaux pour afficher des images sur la console. Nous utiliserons une telle stratégie avec les caractères ASCII. Cela est souvent appelé de l'art ASCII.

L'image est en tons de gris où 0 représente complètement noir et 255 représente complètement blanc avec les valeurs intermédiaires pour les différents niveaux de gris.

La chaîne de caractères `GREYSCALE` définie dans `student.h` donne 10 caractères pour représenter ces tons de gris. En utilisant cette échelle ASCII, le caractère ' ' est blanc et le caractère '@' est noir

- (12) a. Écrivez une fonction pour convertir l'image en un tableau de caractères utilisant l'échelle fournie. Utilisez le prototype de fonction suivant qui vous est fourni dans le code de départ.

Le mappage devrait faire en sorte que chaque intervalle de l'échelle de 0 à 255 est presque égal. Par exemple, si vous choissiez une taille d'intervalle de 12 (mauvaise idée), le caractère d'espace serait mappé aux valeurs de 243 à 255.

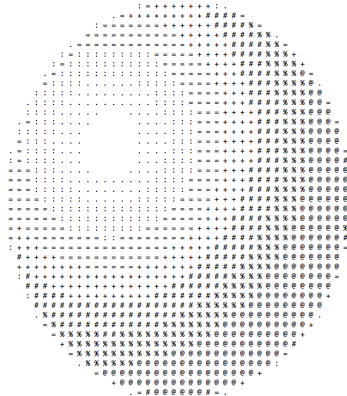
- b. Write a function to display the array of character (7) to the console. Use the following function prototype that is provided in your starting code.

```
void display_image(char ascii[][IMG_SIZE]);
```

- b. Écrivez une fonction pour afficher le tableau de caractères à la console. Utilisez le prototype de fonction suivant qui vous est fourni dans le code de départ.

The output to the console should look something like this

La sortie à la console devrait ressembler à ceci



When you are finished

Ensure your name is in the header comment of the `student.c` and `midterm_2017.c` files and on any notes pages. Leave the computer logged in and your project open in Eclipse, and leave your notes pages on the keyboard. Leave quietly by the closest door to minimize disturbance to others.

Lorsque vous avez terminé

Assurez-vous que votre nom est dans le bloc commentaire des fichiers `student.c` et `midterm_2017.c`, ainsi que sur vos pages de notes. Laissez l'ordinateur connecté, votre projet ouvert dans Eclipse et laissez vos feuilles de notes sur le clavier. Sortez discrètement par la porte la plus proche pour ne pas déranger les autres.

End of the examination

Fin de l'examen