

GNG1506 Automne 2016 – Devoir 1

Disponible : le 5 septembre

Échéance: le 25 sept, minuit

Instructions

Vous devez faire ce travail **INDIVIDUELLEMENT** et suivre toutes les instructions décrites dans le document intitulé "*L'utilisation de BlackBoard Learn*" qui décrit comment soumettre votre devoir au Campus virtuel. Les points suivants sont des consignes spécifiques à ce devoir :

- Vous devez soumettre votre devoir électroniquement via BlackBoard Learn. Soumettez le suivant :
 - Un rapport de devoir dans un fichier PDF (ceci vous permet d'utiliser votre traitement de texte favori pour créer le fichier PDF). Pour la question 1, insérez les modèles de programmation pour les parties (a) et (b) remplis conformément aux instructions reçues pour chaque partie. Vous pouvez remplir le modèle de programmation à l'aide des fonctionnalités de dessin de votre logiciel de traitement de texte ou vous pouvez le faire à la main sur du papier qui sera ensuite numérisé et inséré dans votre document. Pour les questions 2 et 3, insérez dans votre rapport de devoir les codes sources (prenez soin de leur apparence), et capturez la sortie du programme pour tous les cas de test. Soumettez les programmes (fichiers de code source C) développés pour les questions 2 et 3.
 - Tous les fichiers devront être placés dans le répertoire D1_ xxxxxxx où xxxxxxx est votre numéro d'étudiant.
 - Comprimer le répertoire dans un fichier zip D1_ xxxxxxx.zip, où xxxxxxx est votre numéro d'étudiant. C'est le fichier zip que vous soumettez via le BlackBoard Learn avant la date et heure d'échéance du devoir.
 - Vos questions sont fournies dans des fichiers Word et PDF. Vous avez l'option d'utiliser le fichier Word afin d'y insérer vos réponses directement. Un fichier RTF est aussi fourni afin de pouvoir l'éditer avec un logiciel traitement de texte autre que Word.
 - Il n'est pas permis d'utiliser des instructions telles que le branchement ou les boucles qui n'ont pas encore été présentées en classe.
- Commencez le devoir bientôt et **n'**attendez **pas** à la dernière minute. Vous serez plus efficace avec plusieurs petits efforts au courant des quelques semaines avant l'échéance que d'un seul grand effort juste avant la limite de soumission.

Barème (total de 35 points)

- Question 1: 10 points
- Question 2: 10 points
- Question 3 : 15 points

Question 1 (10 points)

a) Le modèle de programmation ci-dessous contient le programme C montré dans la mémoire de programme. Vous devez montrer comment la mémoire de travail est utilisée durant l'exécution de ce programme et comment l'UCT évalue l'expression arithmétique qui donne une valeur à la variable z.

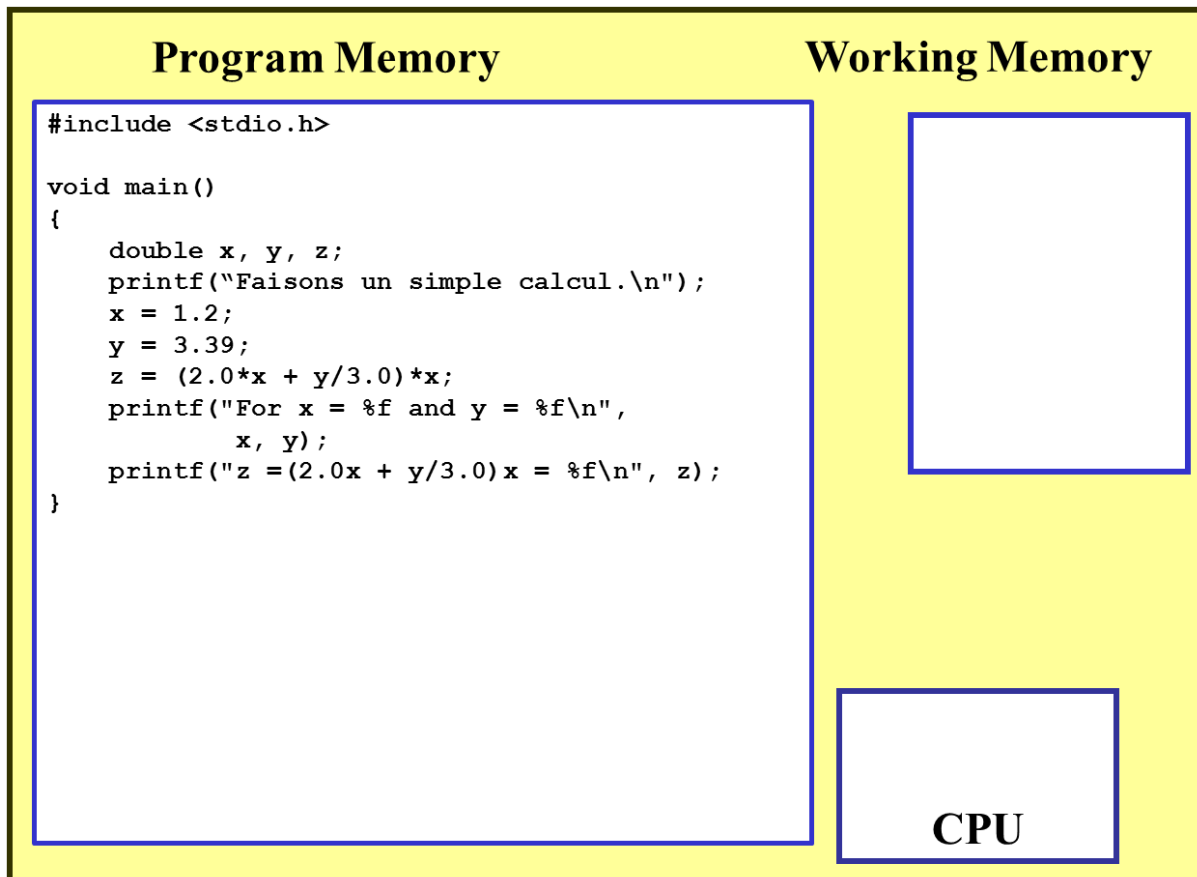
Montrez les variables créées dans la mémoire de travail et comment leurs valeurs changent durant l'exécution du programme.

- Montrez comment les valeurs sont assignées aux variables. Soyez certain de montrer toutes les valeurs qui sont assignées et remplacées. Représenter les affectations successives des valeurs comme suit :

Variable

$\cancel{x}, \cancel{y}, \cancel{z}, \cancel{A}, 10$

- Pour l'opération arithmétique, montrez comment le contenu de la mémoire de travail est déplacé vers l'UCT pour calculer la valeur affectée à la variable z. Montrez **toutes** les opérations effectuées dans l'UCT, c.-à-d. une ligne par opération.



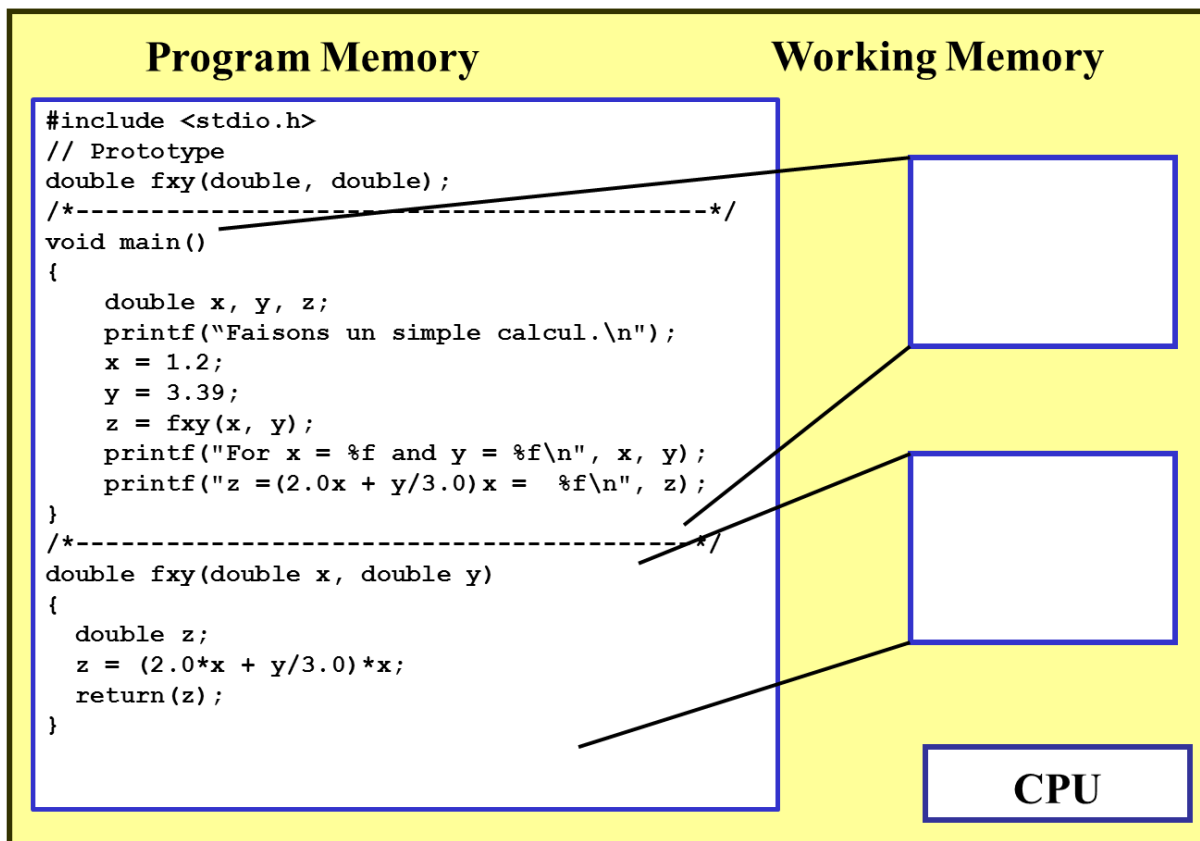
b) Le modèle de programmation ci-dessous contient dans la mémoire de programme le programme C avec 2 fonctions. Vous devez montrer comment la mémoire de travail est utilisée durant l'exécution des deux fonctions de ce programme. Chaque portion de mémoire de travail est associée à une fonction avec une paire de lignes. (Note : la première paire de lignes indique la portion de mémoire de travail allouée à la fonction **main** et la deuxième paire indiquent celle allouée à la fonction **fxxy**).

Montrez les variables (et paramètres) créées dans chaque parcelle de mémoire durant l'exécution des fonctions. Utilisez des flèches pour montrer comment les valeurs sont copiées entre la mémoire de travail allouée à la fonction **main** et la mémoire de travail allouée à la fonction **fxxy**. Il n'est pas nécessaire de montrer comment l'UCT évalue les expressions dans les différentes instructions de ce programme comme vous l'avez fait dans la partie (a).

- Montrez comment les valeurs sont assignées aux variables. Soyez certain de montrer toutes les valeurs qui sont assignées et remplacées. Représenter les affectations successives des valeurs comme suit :

Variable z, y, x, 10

- Utilisez des flèches pour montrer comment les valeurs sont copiées entre la mémoire de travail allouée à la fonction **main** et la mémoire de travail allouée à la fonction **fxxy**.



Question 2 (10 points)

Les équations suivantes décrivent les relations qui existent entre l'accélération a qui est constante, la vitesse v , et le déplacement (distance parcourue) x d'un objet en un mouvement rectiligne uniformément accéléré.

$$v = v_0 + at$$

$$x = v_0t + \frac{1}{2}at^2$$

- où
- a représente l'accélération en m/s^2
 - v représente la vitesse en m/s .
 - x représente la distance parcourue en m .
 - t représente le temps en secondes.
 - v_0 représente la vitesse initiale (c'est-à-dire la vitesse au moment $t = 0$ sec).

Vous devez écrire un programme en C qui demande à l'utilisateur d'entrer la vitesse initiale v_0 et l'accélération a d'un objet un mouvement rectiligne ainsi que le temps t (plus grand que zéro). À partir de ces valeurs le programme doit calculer la distance parcourue x par l'objet après le temps t .

Testez votre programme en utilisant les cas indiqués dans le tableau suivant :

Temps t (secondes)	Vitesse initiale v_0 (m/s)	Accélération (m/s^2)	Vitesse (m/s)	Déplacement x (m)
10	1	0	1.00	10.00
0.5	0	250	125.00	31.25
5.2	10.2	0.5	12.80	59.80
120	60	1.2	204.00	15,840.00
0	60	1.2	60.00	0.00

Lignes directrices:

Logique/Stratégies :

- Dans la fonction **main**,
 - Pensez à utiliser les variables locales suivantes :
 - t : pour stocker le temps entré par l'utilisateur.
 - v_0 : pour stocker la vitesse initiale entrée par l'utilisateur.
 - a : pour stocker l'accélération entrée par l'utilisateur.
 - x : pour stocker la distance parcourue.
 - Demandez à l'utilisateur d'entrer les valeurs de temps, vitesse initiale, et d'accélération. Utilisez des messages courts et appropriés pour le faire. Et enfin, lisez les valeurs à partir du clavier et attribuez les respectivement aux variables t , v_0 , et a .
 - Faites un appel à une fonction, par exemple **calculeDistance**, pour calculer la valeur de la distance parcourue par un objet en mouvement rectiligne uniformément accéléré après un temps t . Stockez le résultat retourné par la fonction **calculeDistance** dans la variable x . Vous aurez besoin de définir au préalable cette fonction dans votre programme.
 - Affichez les résultats de la façon suivante :
"Après un temps $t = 10.2$ secondes, le déplacement $x = 59.80$ m"

- Pour la fonction **calculDistance**,
 - Pensez à utiliser les paramètres suivants :
 - **t** : qui est le temps pendant lequel l'objet s'est déplacé.
 - **v0** : qui est la vitesse initiale de l'objet.
 - **a** : qui est l'accélération de l'objet.
 - Pensez à utiliser la variable locale suivante :
 - **x** : pour stocker la distance calculée (noter que cette variable contiendra la valeur retournée par la fonction).
 - La fonction calcule la valeur de la distance parcourue (voir l'équation donnée au début de la question), la stocke dans la variable x et retourne la valeur stockée dans x.

Testez le programme en utilisant les valeurs fournies dans le tableau présenté plus haut. Dans votre rapport de devoir, présenter la sortie de votre programme pour tous les cas tests.

La réponse à cette question doit fournir :

- 1) Le code source de votre programme (insérer également le code source dans le rapport de devoir).
- 2) Les sorties montrant les résultats de tous les cas de test; insérer la sortie dans le rapport de devoir. Ce qui suit est un exemple de la sortie pour le premier cas de test.

```
"C:\Users\Dave Bosasi Wabali\Downloads\Devoir1_question2.exe"
veuillez entrer le temps <sec.>: 10
veuillez entrer la vitesse initiale <m/s>: 1
veuillez entrer l'acceleration <m/s^2>: 0

Resultats:
Apres un temps t = 10.00 s, le deplacement x = 10.00 (m)

Process returned 70 (0x46)   execution time : 24.505 s
Press any key to continue.
```

Question 3 (15 points)

La loi des gaz parfaits, mentionnée pour la première fois par Émile Clapeyron en 1834 montre la relation simple qui existe entre la pression, la température et la masse d'un gaz.

$$PV = \frac{m}{M} RT$$

- où
- P représente la pression du gaz, exprimé en Kilo pascals (kPa),
 - V représente le volume du gaz, exprimé en m^3 ,
 - T représente la température en Kelvin (K), notez bien que $0^{\circ}C = 273.15K$,
 - M représente la masse moléculaire du gaz, exprimé en (kg/kmole)
(pour l'air $M = 28.97$ kg/kmole),
 - m représente la masse du gaz, exprimé en kilogrammes (kg),
 - R représente la constante des gaz parfaits, exprimée en $kPa\ m^3 / (kmole\ K)$,
 $R = 8.314\ kPa\ m^3 / kmole\ K$.

En utilisant la loi des gaz parfaits, développez un programme en C capable de déterminer le volume d'un gaz en supposant une pression d'une atmosphère ($1\ atm = 101.325\ kPascals$) et que les données suivantes sont obtenues de l'utilisateur : la masse du gaz en kg, la masse moléculaire du gaz en kg/kmole et la température en Celsius. (Indice : portez attention aux unités.)

Suivent les consignes suivantes pour répondre à cette question :

1) Tout d'abord, élaborer un ensemble de cas de test (Excel est un logiciel de pratique qui vous permet de créer des cas de test). Assurez-vous que les cas de test couvrent des larges gammes des données (petite et grande masse, basse et haute température, etc.) et que le gaz sélectionné sera dans l'état de gaz (c.-à-d. utilisez le point d'ébullition du gaz pour choisir une température appropriée). Fournir au moins 5 cas de test (un cas pour chaque gaz donné dans le tableau ci-dessous). Par exemple, 0.1 kg d'argon à -50 Celsius et à une pression de 1 atm ($101.325\ kPascals$) occupera un volume de $0.045629\ m^3$. Le tableau suivant donne des exemples pour divers gaz.

Gaz	Masse moléculaire (kg/kmole)	Point ébullition (degrés Celsius)
Argon	39.948	-185,8
Benzène	78.114	80.4
Hydrogène	2.016	-253
Nitrogène	28.0134	-196
R-114 (réfrigérant)	170.93	3.59

2) Développer votre programme en utilisant le gabarit C GNG1506 (GNG1506gabarit.c), c'est-à-dire, votre programme doit contenir une fonction `main` et une fonction qui calcule le volume du gaz. La fonction `main` contient des instructions pour obtenir les données de l'utilisateur, elle appelle la fonction pour obtenir le volume du gaz, et affiche les résultats à l'écran.

3) Inclure dans votre rapport de devoir, un tableau contenant tous vos cas de test, votre code source, et les sorties montrant les résultats de tous les cas de test. Soumettez également votre fichier de code source.