

Exercice 8

OBJECTIF : REVISION UTILISATION DES FONCTIONS

A l'issue de la réalisation de cet exercice, les étudiants doivent être capable de créer des fonctions avec passage des paramètres par valeur, ainsi que d'utiliser les fonctions des librairies, ceci dans le cadre d'une application console avec le Visual Studio 2015.

PREPARATION DU PROJET

L'exercice 2 sert de mode d'emploi pour la création du projet.

- Lancez le "Microsoft Visual Studio 2015
- Créez un nouveau projet :
 - Projets Visual C++
 - Application console Win32 (Dans la fenêtre modèle)
 - Sous emplacement: introduisez votre répertoire d'exercices sous H:
 - Décocher "Créer le répertoire pour la solution"
 - Sous Nom du projet : **Ex8**
- Paramètre de l'application :
 - Application Console
 - Projet Vide
- **Copie du fichier Ex8.c :**

Il faut copier le fichier Ex8.c de
K:\ES\Maitres-Eleves\SLO\Modules\SL121_LOGA\Exercices\Ex8 dans le répertoire
Ex8. (Usage de l'explorateur Windows)
- Ajoutez au projet le fichier Ex8.c :

Depuis le Visual Studio 2015, sélectionnez "**Fichiers sources**", avec un clique-droit obtenez le menu pour Ajouter un élément existant.
- Test de compilation : vous devez obtenir :

Génération : 1 a réussi, 0 a échoué, 0 a été ignore

DONNEES DU PROBLEMES

Le programme traite 3 cas séparés. Le système de sélection des tests est fourni ainsi que la saisie de ValA et ValB, il faut ajouter les déclarations des variables supplémentaires dont vous avez besoins.

Il faut ajouter un affichage unique de "Ex8 Nom Prénom"

Remarque : il faut placer les fonctions avant le programme principal, cela évite la déclaration d'un "forward prototype".

ACTION DU TEST A

Vous obtenez dans la variable **int ValA** la valeur introduite par l'utilisateur.

Ensuite on appelle la fonction **FtestA** en lui passant par valeur le nombre obtenu (ValA). La fonction **FtestA** doit afficher sur une ligne un nombre d'* correspondant à la valeur reçue. En plus, la fonction doit contrôler la valeur reçue et retourner un code d'exécution du type unsigned short.

Code = 1 valeur fournie entre 1 et 15
Code = 2 valeur reçue supérieur à 15 et limitation à 15
Code = 0 valeur reçue <= 0, action non effectuée.

Après l'appel de la fonction **FtestA**, il faut afficher:

TestA : code execution = x (x représente la valeur reçue)

Exemple de résultat :

```
C:\Users\zfpchr\Documents\ETML_ES\etCoursSW\SL121_LOGA\Exercices\ProjExVS2015\SolEx8\...
Ex8 Christian HUBER
Test A, B ou C, Q pour Quitter
a
TestA: entrez un nombre entre 1 et 15
7
*****
TestA : code execution = 1
Test A, B ou C, Q pour Quitter
a
TestA: entrez un nombre entre 1 et 15
17
*****
TestA : code execution = 2
Test A, B ou C, Q pour Quitter
a
TestA: entrez un nombre entre 1 et 15
-3
TestA : code execution = 0
Test A, B ou C, Q pour Quitter
```

ACTION DU TEST B

Vous obtenez dans la variable `int ValB` la valeur introduite par l'utilisateur.

La réponse de l'utilisateur doit être contrôlée, si la valeur fournie est > 99 ou < -99 , alors il faut limiter la valeur à 99 et -99 respectivement et afficher "Valeur limitee a xx (xx représente la valeur limite).

Ensuite on appelle la fonction **FtestB** en lui passant par valeur `ValB`. Ce paramètre représente une valeur d'exposant.

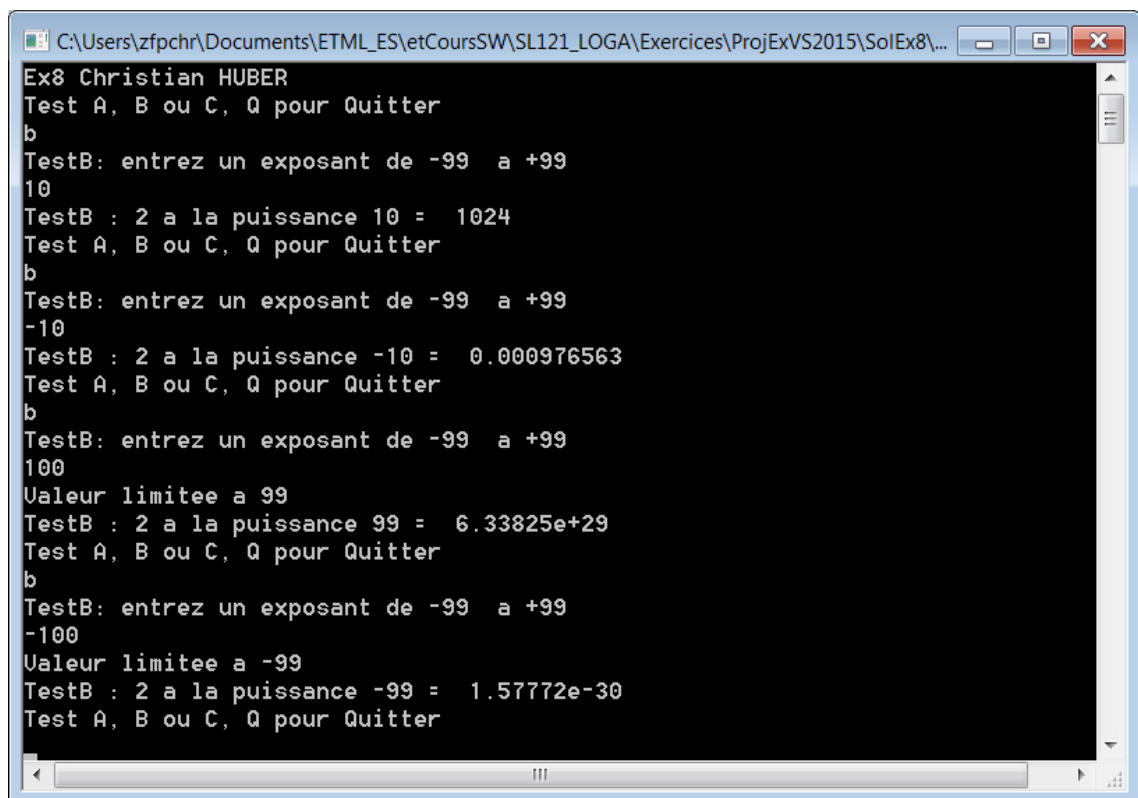
La fonction **FtestB** doit retourner la valeur correspondant à 2^{ValB} . La valeur retournée doit être du type double.

Après l'appel de la fonction **FtestB**, il faut afficher :

TestB : 2 a la puissance x = y (x représente l'exposant et y le résultat)

Remarque : pour un affichage qui s'adapte à la valeur qui peut être très grande ou très petite, utilisez **%g** pour l'affichage du résultat (type double).

Exemple de résultat :



```
Ex8 Christian HUBER
Test A, B ou C, Q pour Quitter
b
TestB: entrez un exposant de -99 a +99
10
TestB : 2 a la puissance 10 = 1024
Test A, B ou C, Q pour Quitter
b
TestB: entrez un exposant de -99 a +99
-10
TestB : 2 a la puissance -10 = 0.000976563
Test A, B ou C, Q pour Quitter
b
TestB: entrez un exposant de -99 a +99
100
Valeur limitee a 99
TestB : 2 a la puissance 99 = 6.33825e+29
Test A, B ou C, Q pour Quitter
b
TestB: entrez un exposant de -99 a +99
-100
Valeur limitee a -99
TestB : 2 a la puissance -99 = 1.57772e-30
Test A, B ou C, Q pour Quitter
```

ACTION DU TEST C

Il faut réaliser la saisie de 2 valeurs de type int, la première est le dividende, la deuxième le diviseur.

Si le diviseur = 0, il ne faut pas appeler les fonctions, par contre il faut afficher :
TestC : erreur division par 0 !

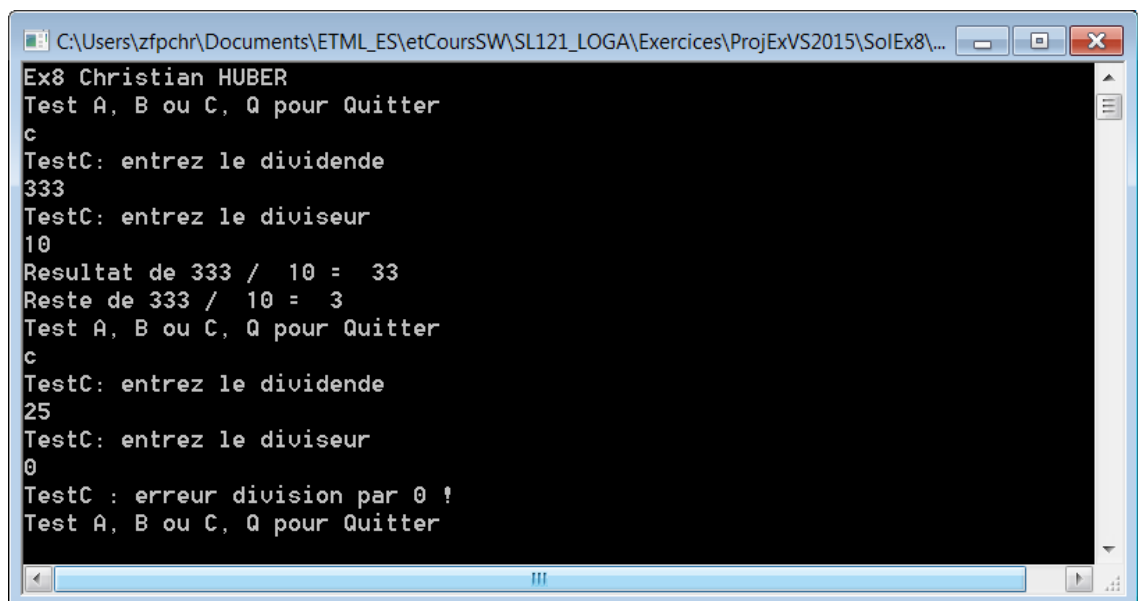
Il faut appeler deux fonctions :

fDivision : comme paramètre le dividende et le diviseur de type int. La fonction retourne le résultat de la division entière.

fReste : comme paramètre le dividende et le diviseur de type int. La fonction retourne le reste de la division entière.

Après l'appel des fonctions **fDivision**, **fReste**, il faut afficher sur 2 lignes:
Resultat de a / b = y (a représente le dividende, b le diviseur et y le résultat)
Reste de a / b = y (a représente le dividende, b le diviseur et y le reste)

Exemple de résultat :



```
Ex8 Christian HUBER
Test A, B ou C, Q pour Quitter
c
TestC: entrez le dividende
333
TestC: entrez le diviseur
10
Resultat de 333 / 10 = 33
Reste de 333 / 10 = 3
Test A, B ou C, Q pour Quitter
c
TestC: entrez le dividende
25
TestC: entrez le diviseur
0
TestC : erreur division par 0 !
Test A, B ou C, Q pour Quitter
```