

PROGRAMMES SUR MACHINE

Programme 1

Construire un programme qui affiche le signe du produit de deux nombres entiers a et b sans faire la multiplication (sans utiliser *)

Programme 2

a) Construire les programmes correspondants aux traitements suivants:

- multiplication entière sans utiliser * ni /
- division (affichage du quotient et du reste) sans utiliser * ni /

b) Le C autorise l'utilisation de "bibliothèques extérieures", c'est-à-dire de fonctionnalités construites par d'autres programmeurs. A titre d'exemple, il existe une bibliothèque nommée conio (uniquement sous Windows), permettant des traitements d'écran comme:

- le choix d'une couleur de tracé : `textcolor(BLUE);`
- le positionnement du curseur (sur un intervalle en ligne de 1 à 24, et en colonne de 1 à 80) : `gotoxy(12,24);`
- le choix d'une couleur pour le fond d'écran (`textbackground(BLACK);`)
- et bien d'autres choses encore.

Pour l'utiliser :

- charger le zip fourni sur l'extranet et le décompresser, (les fichiers .h dans le répertoire include de MinGW, les .a dans le répertoire lib)
- cliquer avec le bouton droit de la souris sur le nom de votre projet,
- choisir l'option "Build Options",
- cliquer sur l'onglet "Linker Settings", puis dans la boîte "Link Libraries", cliquer sur Add,
- aller chercher le fichier "libconio.a",
- valider autant de fois que nécessaire,
- dans le fichier source, rajouter : `#include <conio2.h>`

C'est tout bon !

Reprendre le dernier exercice, et le présenter joliment ! (fond bleu, texte jaune, positionnement etc ...)

c) Calculer la factorielle $N! = 1*2*3*...*(N-1)*N$ d'un entier naturel N en respectant $0!=1$.

Programme 3

Vista Print vend des cartes de visites par paquets de 250 et propose un outil de calcul de facture, à partir du nombre n de paquets commandés par un client et des frais de transport. Le prix du paquet est donné par les règles suivantes :

- 10,00€ HT pour $N \leq 10$
- 8,00€ HT pour $10 < N \leq 30$
- 7,00€ HT pour $30 < N \leq 50$
- 6,00€ HT pour $50 < N \leq 100$
- 5,00€ HT pour $100 < N$

Par exemple, si un client commande 12 paquets de cartes, il les paiera $12*8,00$ € soit 96,00 € HT.

Les frais de transports sont de 8,00 € HT si le nombre de paquets commandés est inférieur à 25 et de 32,00€ HT sinon. Enfin le taux de TVA de 20,0%.

Construire un programme permettant de calculer et d'afficher le montant TTC de la facture d'un client quelconque.

b) Et pourquoi ne pas programmer les forme suivantes ?

* * * * *	* * * * *	*	*	* *
* * * * *	* *	* *	* *	* *
* * * * *	* *	* * *	* *	* *
* * * * *	* *	* * * *	* *	* *
* * * * *	* * * * *	* * * * *	* * * * *	* *

Programme 8

Concevoir un programme qui utilise switch pour prendre un nombre entier et convertir ce nombre en une série de mots. Par exemple, le nombre "529" deviendra la chaîne "cinq deux neuf" ou "neuf deux cinq" au choix. Pour les nombres négatifs, ajouter le mot moins avant le reste des mots.

Programme 9

Construire le tableau d'amortissement d'un emprunt à annuités constantes, en fonction d'un capital, d'une durée et d'un taux variables.

$$Annuité = \frac{capital * taux}{1 - (1 + taux)^{-durée}}$$

Le tableau devra donner le n° de l'année de remboursement, le capital dû en début de période, l'annuité de remboursement, les intérêts de la période, l'amortissement du capital et le capital dû en fin de période.

NB:

- si la durée est supérieure à 15 ans, effacer l'écran pour afficher la suite du tableau
- utiliser les fonctions de conio :
 1. gotoxy(x,y) = position du curseur à l'écran
 2. printf (idem printf mais utilisable avec gotoxy)

Programme 10

a) Construire un programme permettant de convertir un entier d'une base dans une autre (les deux bases doivent être comprises entre 1 et 10).

b) Construire un programme permettant de calculer la somme des chiffres formant un entier et d'en compter le nombre de ceux qui sont impairs.

Bonus :

Construire un programme permettant de déterminer la liste des nombres K (k<100000) formés de N chiffres vérifiant le critère suivant : en partant des nombres composés chacun d'un des N chiffres de K, on compose une suite en calculant la somme des n derniers nombres de la suite pour déterminer le suivant. Si cette suite fournit à un moment le nombre K, la propriété est vérifiée.

Exemple: K=197

1,9,7=>17(=1+9+7),33(=9+7+17),57(=7+17+33),107(=17+33+57),197(=33+57+107)

Autres exemples : 1537, 2208