

---

### 3 - Retour vers le Futur

---

#### Répertoire : BASE

#### 1) Boucles et autres branchements conditionnels

- ✎ Écrire un algorithme permettant d'afficher tous les entiers compris entre 50 et 100.
- ✎ Prendre le temps d'exécuter le débogueur sur l'exercice précédent.
- ✎ Écrire un algorithme permettant d'afficher tous les entiers **pairs** compris entre 50 et 100.
- ✎ Proposer un algorithme permettant de calculer et d'afficher la somme des N premiers nombres entiers, N étant saisi par l'utilisateur.

```
Quelle est le nombre limite [1..?] ? : 5
La somme = 15
Appuyez sur une touche pour continuer... _
```

- ✎ Modifier cet algorithme pour qu'il affiche la moyenne des N entiers. **Rappel : la moyenne ne peut être affichée que s'il y a au moins un entier à traiter.**

```
Quelle est le nombre limite [1..?] ? : 0
La somme = 0
La moyenne ne peut pas etre calculee car N = 0
```

```
Quelle est le nombre limite [1..?] ? : 5
La somme = 15
La moyenne = 3
```

- ✎ Proposer un algorithme visant à compter des entiers jusqu'à ce qu'une valeur spécifique soit saisie. La valeur d'arrêt sera fournie par l'utilisateur en tout début du programme.

```
Veillez entrer la valeur d'arrêt : 0
Veillez entrer la liste des valeurs à compter puis terminer par 0
Validez apres chaque valeur saisie.
1
2
3
0
Nombre d'entiers lus = 3
Appuyez sur une touche pour continuer...
```

#### 2) Du dessin "en mode texte"

- ✎ Proposer un algorithme permettant d'afficher dans la console une ligne d'étoiles ayant une longueur spécifiée par l'utilisateur (comprise entre 1 et 50). Le test de validité de la valeur saisie n'est pas demandé.


Pour tracer la ligne d'étoiles, nous utiliserons le caractère "\*". C'est-à-dire que pour une ligne de longueur 20 vous devez obtenir, sur la console, le résultat suivant.

```
Donner la longueur de la ligne d'etoiles : 20
*****
Appuyez sur une touche pour continuer... _
```

✍ Faites évoluer cet algorithme pour vérifier la saisie de l'utilisateur. S'il saisit une valeur incorrecte, l'utilisateur devra saisir à nouveau la valeur jusqu'à ce que celle-ci soit correcte. La figure ci-dessous vous présente un résultat possible.

```
Donner la longueur de la ligne d'etoiles : 55
Erreur [1; 50] - Donner la longueur de la ligne d'etoiles : 50
*****
Appuyez sur une touche pour continuer... _
```

### 3) Tracer un sapin (sans création de sous-programmes)

 Proposer un algorithme permettant d'afficher un sapin pointe en bas composé de lignes d'étoiles. L'utilisateur devra pouvoir configurer le sapin souhaité en fournissant :

- le nombre de blanc ( $nbb$ ) précédant la plus longue ligne d'étoiles avec  $nbb \in [1;10]$ .
- le nombre d'étoiles ( $nbe$ ) sur la plus longue ligne d'étoiles avec  $nbe \in [1;20]$  et  $nbe$  impair.

Dans cet exercice, on vérifiera que les valeurs saisies soient correctes.

La fenêtre ci-dessous présente un sapin ayant comme valeurs  $nbb = 5$  et  $nbe = 15$ .



---

## 4 - « Avancement »

---

### Répertoire : BASE

#### 1) *Le compte est bon*

✎ Écrire un algorithme permettant de saisir des factures. L'utilisateur donne tous les renseignements relatifs aux factures. Pour chaque commande, l'utilisateur saisit tout d'abord le code de facture et le nombre d'articles. Le prix unitaire ainsi que la quantité commandée sont ensuite saisis pour chaque article.

La réduction applicable au montant total de la facture doit également être calculée.

La réduction se calcule de la manière suivante :

**si** montant de la facture > 50 **alors**

réduction de 10% sur la totalité de la facture ;

**si** montant de la facture [150, 300] **alors**

réduction de 10 % sur la tranche du montant [1 ; 50] et 5% au dessus de 50 ;

**si** montant de la facture > 300 **alors**

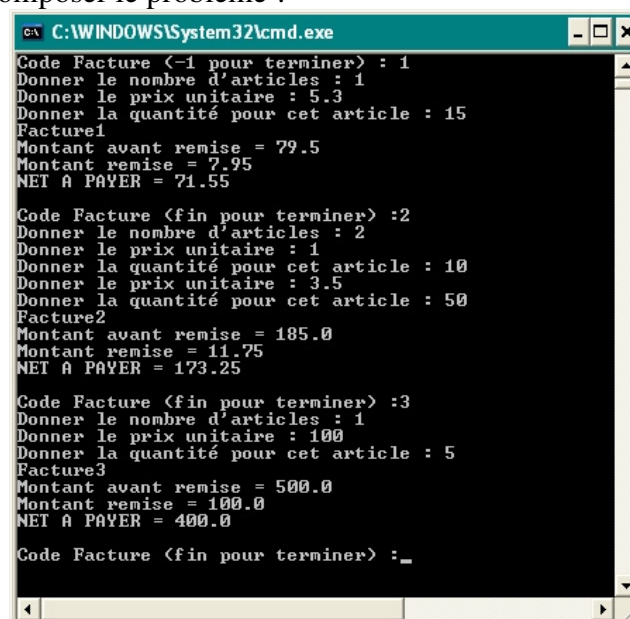
réduction de 20% sur la totalité de la facture ;

L'algorithme affichera, pour chaque facture, le montant de la facture avant remise, le montant de la remise et le montant net de la facture. On rappellera lors de ces affichages le numéro de facture correspondant.

Note : La fin de la saisie des factures sera signalée par le code de facture spécifique (valeur d'arrêt) -1.

Note 2 : Les informations relatives aux factures ne sont pas sauvegardées. Un simple affichage est demandé. **Nous ne souhaitons donc pas utiliser des tableaux.**

Note 3 : Pensez à décomposer le problème !



```
C:\WINDOWS\System32\cmd.exe
Code Facture <-1 pour terminer> : 1
Donner le nombre d'articles : 1
Donner le prix unitaire : 5.3
Donner la quantité pour cet article : 15
Facture1
Montant avant remise = 79.5
Montant remise = 7.95
NET A PAYER = 71.55

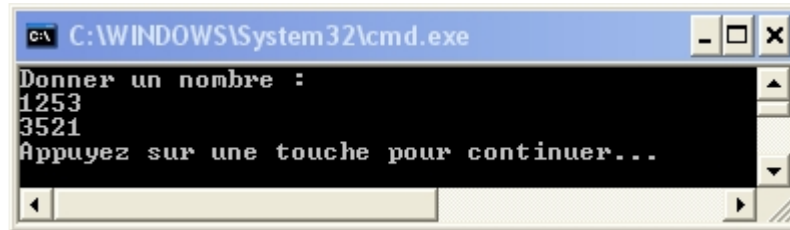
Code Facture <fin pour terminer> :2
Donner le nombre d'articles : 2
Donner le prix unitaire : 1
Donner la quantité pour cet article : 10
Donner le prix unitaire : 3.5
Donner la quantité pour cet article : 50
Facture2
Montant avant remise = 185.0
Montant remise = 11.75
NET A PAYER = 173.25

Code Facture <fin pour terminer> :3
Donner le nombre d'articles : 1
Donner le prix unitaire : 100
Donner la quantité pour cet article : 5
Facture3
Montant avant remise = 500.0
Montant remise = 100.0
NET A PAYER = 400.0

Code Facture <fin pour terminer> :_
```

✎ À partir d'un nombre saisi au clavier, écrire un algorithme permettant d'afficher successivement les chiffres du nombre dans le sens des unités vers les centaines (utiliser la division entière `div` et le modulo `mod`). **Attention : on ne souhaite pas modifier la valeur saisie.**

Exemple :



✎ Parmi tous les entiers  $> 1$ , seuls 4 peuvent être représentés par la somme des cubes de leurs chiffres. Ecrire un programme en langage algorithmique permettant de trouver ces nombres.

Ainsi par exemple,  $153 = 1^3 + 5^3 + 3^3$

Remarque : les quatre nombres sont compris entre 150 et 410.

## 2) Bizarries mathématiques

La table de multiplication de 9 recèle de différents « secrets ». Nous vous proposons de les identifier.

✎ En fait le premier résultat est égal à 09 ( $9 \times 1$ ). Pour passer d'une ligne à la suivante, il suffit de d'incrémenter le premier chiffre et décrémenter le deuxième... Vérifier le en affichant la table de multiplication de 9 sans la calculer !

✎ Cet exercice repose sur le résultat vu dans l'exercice précédent. Dans la table de multiplication de 9, pour chaque résultat, la somme des chiffres composant le résultat fait **TOUJOURS 9**.... Vérifier le. Vous devez obtenir un résultat proche de celui-ci :

```

9x1=9   [9 = 9]
9x2=18  [8 + 1 = 9]
9x3=27  [7 + 2 = 9]
9x4=36  [6 + 3 = 9]
9x5=45  [5 + 4 = 9]
9x6=54  [4 + 5 = 9]
9x7=63  [3 + 6 = 9]
9x8=72  [2 + 7 = 9]
9x9=81  [1 + 8 = 9]
9x10=90 [0 + 9 = 9]

```

## Pour aller plus loin....

### 3) Tracer un sapin

✎ Écrire un nouvel algorithme à partir du précédent permettant d'afficher un sapin tête en haut. Utilisez si possible une démarche différente par rapport à l'exercice précédent (calculer

par exemple le nombre d'espaces nécessaires pour chaque ligne en fonction du nombre d'étoiles à afficher).

A screenshot of a Windows Command Prompt window titled "C:\WINDOWS\System32\cmd.exe". The window has standard Windows XP-style title bar controls (minimize, maximize, close). The command prompt shows the following text:  

```
Donner le nombre de blancs : 5  
Donner le nombre d'etoiles : 15  
  
      *  
    ***  
  *****  
*****  
*****  
*****  
*****  
*****  
*****
```

  
At the bottom, it says "Appuyez sur une touche pour continuer..." followed by a cursor. Below the command prompt area is a scroll bar.

- 📄 Reprendre le dessin du sapin en mode texte en introduisant des sous-programmes (affichage d'une ligne d'étoiles...). Tentez la récursivité !

#### 4) Etude d'une fonction

Soit la fonction suivante :

$$f(x) = 3x^2 + 4x - 1$$

- 📌 Proposer un algorithme permettant de calculer une moyenne des valeurs de cette fonction échantillonnée entre deux bornes. Le pas d'échantillonnage ainsi que les deux bornes de calcul seront fournis par l'utilisateur au début du programme.