



Concours Informatique Luxembourgeois 2004

Epreuve de Sélection Préliminaire



Cotation sur 100 points

Problème I - Factorielle

10 points

La factorielle d'un nombre entier naturel n , notée $n!$, est définie comme suit:

$$n! = 1 \times 2 \times 3 \times 4 \times \dots \times n \text{ avec } n \in \mathbb{IN}_0$$

Exemples

$1! = 1$	$8! = 40320$
$2! = 2$	$10! = 3628800$
$3! = 6$	$12! = 479001600$
$4! = 24$	$25! \approx 1,55 \times 10^{25}$
$6! = 720$	$50! \approx 3,04 \times 10^{64}$

Ecrivez un programme qui calcule la factorielle d'un nombre entier naturel.

Entrées et sorties du programme

- ⇒ Le programme lit de l'entrée standard (clavier) le nombre n .
- ⇐ Le programme affiche sur la sortie standard (écran) la factorielle de n .

Restrictions

$n \in \mathbb{IN}_0$, $n \leq 170$ (pour Turbo Pascal, la variable contenant $n!$ sera du type Double)



Remettez le programme sous le nom FACTOR.xxx, avec xxx=PAS ou C(PP). Remettez également le fichier binaire exécutable FACTOR.EXE correspondant au programme.

Problème II - Nombres sympathiques

20 points

Par définition, un nombre sympathique s est un nombre entier naturel

- ♦ qui est une puissance de 2 (2^n avec $n \in \mathbb{IN}_0$)
- ou
- ♦ qui ne se compose que de chiffres représentant une puissance de 2 (2^n avec $n \in \mathbb{IN}_0$)

Exemple: nombres sympathiques inférieurs à 100

2; 4; 8; 16; 22; 24; 28; 32; 42; 44; 48; 64; 82; 84; 88

Ecrivez un programme qui détermine si un nombre est sympathique.

Entrées et sorties du programme

- ⇒ Le programme lit de l'entrée standard (clavier) le nombre s .
- ⇐ Le programme affiche sur la sortie standard (écran) le texte VRAI si s est un nombre sympathique, le texte FAUX sinon.



Restrictions

$s \in \mathbb{N}_0, 2 \leq s \leq 2147483647$ (pour Turbo Pascal, s sera une variable du type LongInt)



Remettez le programme sous le nom SYMPA.xxx, avec xxx=PAS ou C(PP). Remettez également le fichier binaire exécutable SYMPA.EXE correspondant au programme.

Problème III - Décodage de messages

30 points

Le système de codage CIL permet de transmettre des messages confidentiels. Un message codé d'après la méthode CIL consiste en une longue chaîne de caractères qui contient le message confidentiel.

Exemple

Le message codé

`qn9è%dr.eç3RS4eoan@Ed06e6xzD9-Ohvd?om-u Y8s&R/99mTGad@rjtd52i9k/*"2360J(/ü8hrDe0oujrr/rewts10!ta[Gç2`

contient le message confidentiel [Rendez-vous/mardi/20/heures!](#)

Le premier caractère du message confidentiel est situé à la position initiale 12. Les caractères suivants sont toujours décalés de 3 positions vers la droite par rapport au caractère précédant. Le couple (12; 3) est appelé la clé de codage.

Ecrivez un programme qui effectue un décodage d'un message codé suivant la méthode CIL.

Entrées et sorties du programme

- ⇒ Le programme lit du fichier texte INPUT.TXT le message codé d'après le format suivant:
 - La première ligne contient la longueur x du message codé.
 - La deuxième ligne contient la longueur n du message confidentiel.
 - La troisième ligne contient la position du premier caractère du message confidentiel dans le message codé.
 - La quatrième ligne contient le décalage d .
 - La cinquième ligne contient le message codé.
- ⇐ Le programme écrit dans un fichier texte OUTPUT.TXT le message confidentiel.

Exemple

Le fichier d'entrée correspondant à l'exemple ci-dessus a le format suivant.

```
99
28
12
3
qn9è%dr.eç3RS4eoan@Ed06e6xzD9-Ohvd?om-u Y8s&R/99mTGad@rjtd52i9k/*"2360J(/ü8hrDe0oujrr/rewts10!ta[Gç2
```

Restrictions

$x \in \mathbb{N}_0, x \leq 255$

$n \in \mathbb{N}_0, n \leq 255$

$d \in \mathbb{N}_0, d \leq 127$

Dans le message codé et le message confidentiel, l'espace est remplacé par le trait oblique.



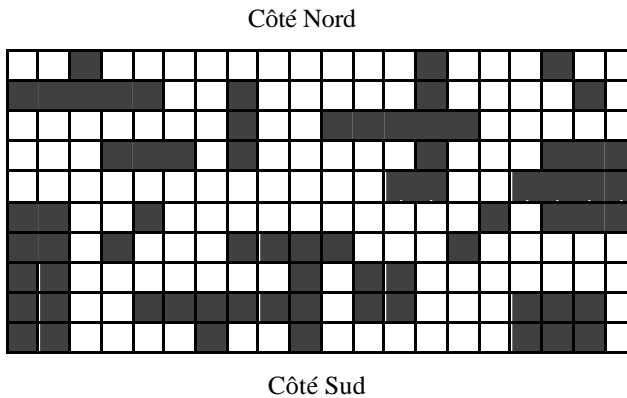
Remettez le programme sous le nom DECODAGE.xxx, avec xxx=PAS ou C(PP). Remettez également le fichier binaire exécutable DECODAGE.EXE correspondant au programme. Vous n'avez pas besoin de remettre des fichiers textes.

Problème IV - Chemin le plus court**40 points**

Sur un terrain rectangulaire $m \times n$ sont positionnés des obstacles. Il s'agit de trouver la longueur du plus court chemin du côté Nord vers le côté Sud du terrain en ne se déplaçant que vers le bas respectivement vers la gauche ou la droite.

Exemple

Voici un terrain 20×12 .



La longueur du chemin le plus court est 12.

Ecrivez un programme qui calcule le chemin le plus court en direction Nord-Sud d'un terrain rectangulaire quelconque.

Entrées et sorties du programme

- ⇒ Le programme lit du fichier texte INPUT.TXT les dimensions du terrain ainsi que la position des obstacles dans le format suivant:
La première ligne contient les dimensions du terrain: longueur et largeur (séparées par un espace).
Les lignes suivantes contiennent les positions des obstacles: 0 ≡ pas d'obstacle, 1 ≡ obstacle.
- ⇐ Le programme affiche sur la sortie standard (écran) le chemin le plus court. S'il n'y a pas de chemin, le programme affiche le nombre -1.

Exemple

Le fichier d'entrée correspondant à l'exemple ci-dessus a le format suivant.

```
20 10
0010000000001000100
11111001000001000010
00000001001111100000
00011101000001000111
00000000000011001111
11001000000000010111
11010001111000100000
11000000010110000000
1100111110110001110
11000010010000001110
```

**Restrictions**

$$m \in \mathbb{N}_0, 2 \leq n \leq 127$$

$$n \in \mathbb{N}_0, 2 \leq m \leq 127$$



Remettez le programme sous le nom CHEMIN.xxx, avec xxx=PAS ou C(PP). Remettez également le fichier binaire exécutable CHEMIN.EXE correspondant au programme. Vous n'avez pas besoin de remettre des fichiers textes.