

Attention: il faut que vos fichiers binaires soient exécutables indépendamment de votre environnement de programmation et soient compatibles WIN32.

Problème I - TRIANGLE DE PASCAL

20 points



Le mathématicien Blaise Pascal a trouvé le triangle suivant qui est un arrangement géométrique de coefficients binomiaux. Ce triangle est composé de plusieurs lignes. Chaque ligne contient des nombres entiers positifs différents de 0. Le sommet du triangle et les extrémités des lignes valent 1. Les autres nombres sont égaux à la somme de deux nombres de la ligne précédente, celui au dessus et celui au dessus à gauche.

Considérons la dernière ligne dans l'exemple donné: le nombre 36 est obtenu en additionnant 28 et 8 (tout comme le deuxième nombre 36 de cette ligne).

Les nombres de ce triangle correspondent aux coefficients des polynômes $(x + y)^n$. Par exemple pour $n = 3$:

$$(x + y)^3 = 1x^3 + 3x^2y + 3xy^2 + 1y^3$$

Entrée

Le programme lit le nombre entier positif M, correspondant au nombre de lignes qu'on veut afficher (10 dans l'exemple), au clavier.

Sortie

Le programme affiche les M premières lignes du triangle de Pascal à l'écran.

Exemple

```

10
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
    
```

Restriction

$$M \leq 20$$



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

Problème II - ACKERMANN

20 points

Une partie importante de l'informatique théorique traite des fonctions dites primitives-récurrentes. Au début, on pensait que toutes les fonctions étaient de ce type, jusqu'à ce que le mathématicien Wilhelm Ackermann présentât une fonction qui croît trop vite pour être primitive-récurrente.

La fonction est définie comme suit:

$$\begin{aligned} A(0, m) &= m + 1 \\ A(n + 1, 0) &= A(n, 1) \\ A(n + 1, m + 1) &= A(n, A(n + 1, m)) \end{aligned}$$

On demande d'écrire un programme qui implémente cette fonction.



Wilhelm Ackermann

Entrée

Le programme lit deux nombres entiers positifs X et Y, séparés par un espace, au clavier.

Sortie

Le programme affiche le résultat de A(X, Y) à l'écran.


Exemples

Lecture au clavier: 2 2

Affichage à l'écran: 7

Lecture au clavier: 3 7

Affichage à l'écran: 1021

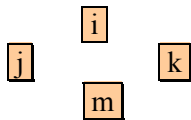
 Remettez le programme sous le nom ACKER.xxx, avec xxx=PAS ou C(PP). Remettez également le fichier binaire exécutable ACKER.EXE correspondant au programme.

Problème III - LE TAQUIN

60 points

Le taquin est un jeu d'enfants, contenant les nombres de 1 à 15 dans une matrice quadratique 4x4, avec un emplacement vide, représenté par la valeur 0 (zéro). Au début, les quinze nombres sont répartis de façon aléatoire et le jeu consiste à les ordonner. Pour cela, le jeu permet de glisser des nombres adjacents à l'emplacement vide dans celui-ci.

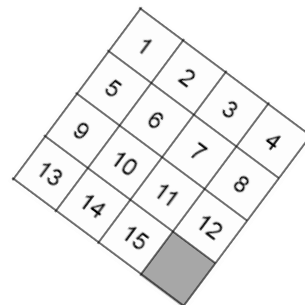
Ecrivez un programme qui permet de jouer ce jeu. Les glissements se font à l'aide des touches suivantes:



Le programme génère un jeu initial et l'affiche à l'écran. Ensuite, il lit la direction de glissement au clavier (touches 'i', 'j', 'm' et 'k'). La nouvelle configuration est ensuite affichée jusqu'à ce que le taquin est résolu. Affichez un message adéquat en ce moment (p.ex.: 'Bravo!').

Exemple d'exécution

12	6	4	1
13	7	0	5
2	3	8	14
9	10	11	15



Dans cette configuration, la direction:

'i' fait glisser le nombre 8 vers le haut,
'm' fait glisser le nombre 4 vers le bas,
'j' fait glisser le nombre 5 vers la gauche,
'k' fait glisser le nombre 7 vers la droite.


D'autres glissements ne sont pas permis. Après l'entrée de la direction donnée par la touche 'i', par exemple, on obtient:

12	6	4	1
13	7	8	5
2	3	0	14
9	10	11	15

Les deux configurations suivantes montrent un taquin terminé:

1	2	3	4	1	2	3	4
5	6	7	8	5	6	7	8
9	10	11	12	9	10	11	0
13	14	15	0	12	13	14	15

Selon la configuration initiale, on peut obtenir soit la première soit la deuxième de ces configurations finales.

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