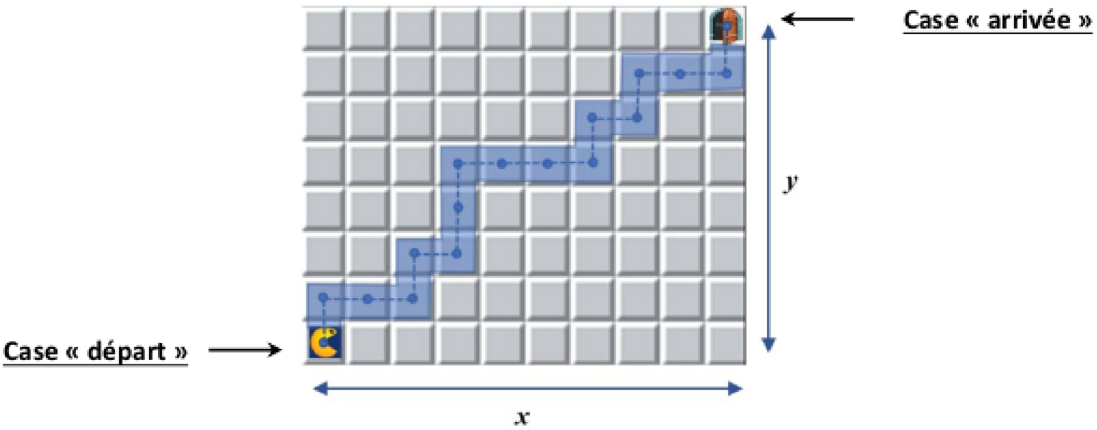
Un jeu vidéo est constitué d'une grille rectangulaire de x cases par y cases, x et y étant des entiers naturels non nuls (dans l'exemple d'illustration ci-dessous on a $x = 10$ et $y = 8$). Un personnage se trouvant en bas à gauche de la grille (case « départ ») doit rejoindre une porte située en haut à droite (case « arrivée ») en se déplaçant suivant deux directions : vers le haut ou vers la droite.

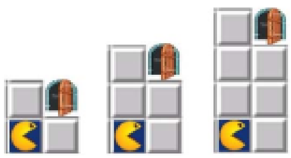


On appelle « chemin » l'ensemble des cases empruntées par le personnage (un exemple de chemin a été tracé ci-dessous). Le but de ce problème est d'étudier le nombre de chemins possibles entre la case de départ et la case d'arrivée, nombre que l'on notera $C(x; y)$.

1 Partie 1 : étude de premiers cas

1. Soit un entier $y \geqslant 1$. Donner la valeur de $C(1; y)$.

2. Donner les valeurs de $C(2;2)$, $C(2;3)$ et $C(2;4)$.



3. Soit un entier $y \geqslant 1$. Déterminer $C(2; y)$ en fonction de y. Justifier la réponse.

4. Justifier que $C(3;4) = C(2;4) + C(2;3) + C(2;2) + C(2;1)$ puis calculer $C(3;4)$.

5. Soit un entier $y \geqslant 1$. Exprimer $C(3; y)$ en fonction de y sous la forme d'une fraction.

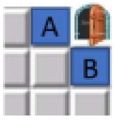
6. Justifier que pour tous les entiers $x \geqslant 1$ et $y \geqslant 1$, on a $C(y; x) = C(x; y)$.

2 Partie 2 : calcul de $C(6;6)$

En considérant les chemins passant par les cases A, B, C, D, E ou F de la grille ci-dessus :

1. Justifier que $C(6;6) = 2 \times \left(C^2(1;6) + C^2(2;5) + C^2(3;4)\right)$. On a noté $C^2(1;6)$ pour $C(1;6) \times C(1;6)$.
2. En déduire la valeur de $C(6;6)$.

3 Partie 3 : calcul de $C(x;y)$ et application à une situation de jeu



1. En considérant les chemins passant par les cases A et B adjacentes à la porte, montrer que pour tous les entiers $x \geqslant 2$ et $y \geqslant 2$, on a :
$$C(x;y) = C(x-1;y) + C(x;y-1)$$
2. Recopier et compléter alors le tableau des $C(x;y)$ ci-dessous (de bas en haut) :

6	1					
5	1					
4	1					
3	1					
2	1					
1	1	1	1	1	1	1
y / x	**1**	**2**	**3**	**4**	**5**	**6**

3. On considère la fonction `Python` ci-dessous.

   `calculer_coef(6)` renvoie (avec `return ligne`) une liste contenant tous les coefficients $C(x;6)$ avec $1 \leqslant x \leqslant 6$.

   Pour tracer le déroulement du programme on a inséré deux instructions `print(ligne)`.

   Déterminer l'affichage obtenu lorsqu'on exécute `calculer_coef(6)` dans une console `Python`.

```python
def calculer_coef(n):
    """
    Paramètre : un entier n >= 1
    Valeur renvoyée : une liste contenant tous les coefficients C(x;n)
        avec 1 <= x <= n
    """
    #initialisation de la première ligne
    ligne = [1 for k in range(n)]
    print(ligne) #trace d'exécution
    # n-1 >= y > 0 et y = y - 1 à chaque itération
    for y in range(2, n + 1):
        for x in range(2, n + 1):
            ligne[x - 1] = ligne[x - 2] + ligne[x - 1]
        print(ligne) #trace d'exécution
    return ligne
```

4. Compléter la fonction `Python` ci-dessous pour que `calculer_tableau(n)` renvoie une liste de liste contenant tous les coefficients $C(x;y)$ pour tous les entiers x et y tels que $1 \leqslant x \leqslant n$ et $1 \leqslant y \leqslant n$.

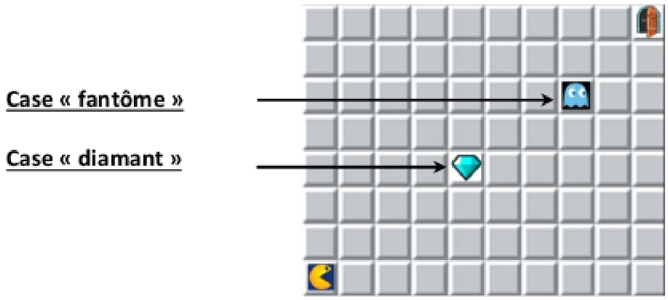
   Tester en ligne en suivant le lien https://frama.link/dm-denombrement.

   On donne un exemple d'exécution. La première ligne (les éléments `coef[0][x]`) et la première colonne (les éléments `coef[y][0]`), ne représentent pas les coefficients $C(x;y)$: $C(x;y)$ est représenté par `coef[y][x]` avec $1 \leqslant \mathtt{x} \leqslant n$ et $1 \leqslant \mathtt{y} \leqslant n$.

```
>>> coef = calculer_tableau(4)
>>> coef
[[0, 0, 0, 0, 0],
 [0, 1, 1, 1, 1],
 [0, 1, 2, 3, 4],
 [0, 1, 3, 6, 10],
 [0, 1, 4, 10, 20]]
```

```python
def calculer_tableau(n):
    """
    Paramètre : un entier n >= 1
    Valeur renvoyée : une liste de listes contenant tous les C(x;y)
    avec 1 <= x <= n et 1 <= y <= n
    """
    #initialisation de la première ligne
    coef = [ [0 for x in range(n+1)] for y in range(n+1)]
    for y in range(1, n + 1):
        for x in range(1, n + 1):
            if ..................:
                coef[y][x] = 1
            else:
                coef[y][x] = .....................
    return coef
```

5. Application à une configuration de jeu :



Le personnage doit rejoindre la case « arrivée » en passant par la case « diamant » et en évitant la case « fantôme ».

Calculer le nombre de chemins gagnants, c'est-à-dire respectant ces deux contraintes.

4 Partie 4 : calcul explicite de $C(x; y)$

Pour représenter un chemin, chaque déplacement d'une case vers le haut est noté **H** et chaque déplacement d'une case vers la droite est noté **D**.
Ainsi le chemin donné en exemple au début de l'énoncé est représenté par la liste :

H D D H D H H D D D H D H D D H

1. Justifier que pour tous les entiers $x \geqslant 1$ et $y \geqslant 1$, $C(x; y)$ est égal au nombre de façons de choisir la place de $x-1$ lettres D parmi la liste de $x+y-2$ lettres représentant un chemin allant de la case de départ vers la case de coordonnées $(x; y)$.

2. On considère $x-1$ objets qu'on ordonne dans une liste. Expliquer pourquoi il existe $1 \times 2 \times \ldots \times (x-1)$ listes ainsi ordonnées.

3. En déduire que pour tous les entiers $x \geqslant 1$ et $y \geqslant 1$: $C(x; y) = \dfrac{\overbrace{(x+y-2) \times (x+y-3) \times \ldots \times y}^{x-1 \text{ facteurs}}}{1 \times 2 \times \ldots \times (x-1)}$.

   Dans le cours sur le dénombrement, on verra que $C(x; y)$ est égal au coefficient binomial $\dbinom{x+y-2}{x-1}$.

4. Utiliser le résultat précédent pour calculer la valeur de $C(10; 8)$ puis déterminer la proportion du nombre de chemins gagnants de la question **5.** de la partie 3, par rapport à la valeur $C(10; 8)$ (arrondir le résultat à 10^{-4}).

5. Compléter la fonction `Python` ci-dessous pour que `coef(x, y)` renvoie la valeur de $C(x; y)$. Tester en ligne en suivant le lien https://frama.link/dm-denombrement.

```python
def coef(x, y):
    """Paramètres : deux entiers x >= 1 et y >= 1
    Valeur renvoyée : C(x ; y)
    """
    num = 1
    denom = 1
    for k in range(1, x):
        num = num * ......
        denom = .........
    return num // denom
```