

# **Solutions des exercices du Cours 3**

Vincent Lepetit

`vincent.lepetit@epfl.ch`

```

cout << "Entrez le premier nombre:" << endl;
cin >> n;
cout << "Entrez le deuxieme nombre:" << endl;
cin >> p;

if ((n < p) && (2 * n >= p))
    cout << "Test 1" << endl;

if ((n < p) || (2 * n > p))
    cout << "Test 2" << endl;

if ( ((n < p) && (2 * n > p)) || (n > p) )
    cout << "Test 3" << endl;

if (!(n > p))
{
    if ((2 * n <= p) && (3 * n > p))
        cout << "A" << endl;

    if (p % 2 == 0)
        cout << "B" << endl;
}
else
    if (3 * p > n)
        cout << "C" << endl;

```

## Exercice

Qu'affiche ce programme quand l'utilisateur entre **10 et 5** ? **5 et 10** ? **5 et 5** ? **1 et 4** ?

- Pour le ET (&&): **les deux** conditions doivent être vraies;
- Pour le OU (||): **au moins l'une** des conditions doit être vraie.

# Solution

```
if ((n < p) && (2 * n >= p))  
    cout << "Test 1" << endl;
```

```
if ((n < p) || (2 * n > p))  
    cout << "Test 2" << endl;
```

```
if ( ((n < p) && (2 * n > p)) || (n > p) )  
    cout << "Test 3" << endl;
```

```
if (!(n > p))  
{  
    if ((2 * n <= p) && (3 * n > p))  
        cout << "A" << endl;  
  
    if (p % 2 == 0)  
        cout << "B" << endl;  
}  
else  
    if (3 * p > n)  
        cout << "C" << endl;
```

$!(n > p) \Leftrightarrow n \leq p$

**n = 10 et p = 5** → Test 2, Test 3, C

**n = 5 et p = 10** → Test 1, Test 2, A, B

**n = 5 et p = 5** → Test 2

**n = 1 et p = 4** → Test 2, B

# Exercice

Ecrivez un programme qui, à partir d'un montant lu en donnée, détermine un montant net obtenu par application d'une remise de:

- 1% si le montant est compris entre 2000F et 5000F (ces valeurs comprises);
- 5% si le montant est supérieur à 5000F.

```
int main(int argc, char ** argv)
{
    float montant, montant_net;

    cout << "Entrez le montant:" << endl;
    cin >> montant;

    // ...

    cout << "Le montant net est " << montant_net << endl;
}
```

# Solution

Ecrivez un programme qui, à partir d'un montant lu en donnée, détermine un montant net obtenu par application d'une remise de:

- 1% si le montant est compris entre 2000F et 5000F (ces valeurs comprises);
- 5% si le montant est supérieur à 5000F.

```
int main(int argc, char ** argv)
{
    float montant, montant_net;

    cout << "Entrez le montant:" << endl;
    cin >> montant;

    if (montant < 2000)
        montant_net = montant; // pas de remise
    else
        if (montant <= 5000) // 2000 <= montant <= 5000
            montant_net = 99 * montant / 100; // remise de 1%
        else // montant > 5000
            montant_net = 95 * montant / 100; // remise de 5%

    cout << "Le montant net est " << montant_net << endl;
}
```

# Attention

Si on fait:

```
if (montant >= 2000 && montant < 5000)
    montant_net = 99 * montant / 100; // remise de 1%
if (montant >= 5000)
    montant_net = 95 * montant / 100; // remise de 5%
```

aucune valeur n'est affectée à la variable `montant_net` si `montant` est inférieur à 2000.

L'instruction

```
cout << "Le montant net est " << montant_net << endl;
```

n'affichera donc pas la valeur attendue.

# Qu'affichent les programmes suivants ?

**A:**

```
for(int i = 0; i < 5; i++)  
    cout << i * i << endl;
```

**B:**

```
for(int j = 0; j < 5; j++)  
    cout << j * j << ", ";  
cout << endl;
```

**C:**

```
int a;  
for(int k = 0; k < 5; k++)  
{  
    a = 2 * k + 1;  
    cout << a << endl;  
}
```

**D:**

```
int a = 1;  
for(int k = 0; k < 5; k++)  
{  
    a = 2 * a + 1;  
    cout << a << endl;  
}
```

**E:**

```
int a = 1;  
for(int k = 0; k < 5; k++)  
{  
    a = 2 * a + 1;  
    cout << a << endl;  
    if (a % 3 == 0)  
        cout << "*" << endl;  
}
```

# A: Solution

**A:**

```
for(int i = 0; i < 5; i++)  
    cout << i * i << endl;
```

affiche:

0

1

4

9

16

# B : Solution

**B:**

```
for(int j = 0; j < 5; j++)  
    cout << j * j << ", ";  
cout << endl;
```

affiche:

0, ◆

# B : Solution

**B:**

```
for(int j = 0; j < 5; j++)  
    cout << j * j << ", ";  
cout << endl;
```

affiche:

0, 1, ◆

# B : Solution

**B:**

```
for(int j = 0; j < 5; j++)  
    cout << j * j << ", ";  
cout << endl;
```

**affiche:**

0, 1, 4, ◆

# B : Solution

**B:**

```
for(int j = 0; j < 5; j++)  
    cout << j * j << ", ";  
cout << endl;
```

**affiche:**

0, 1, 4, 9, ◆

# B : Solution

**B:**

```
for(int j = 0; j < 5; j++)  
    cout << j * j << ", ";  
cout << endl;
```

**affiche:**

0, 1, 4, 9, 16, ◆

# B : Solution

**B:**

```
for(int j = 0; j < 5; j++)  
    cout << j * j << ", ";  
cout << endl;
```

**affiche:**

0, 1, 4, 9, 16,



# B : Solution

**B:**

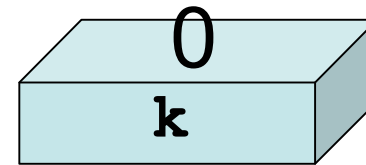
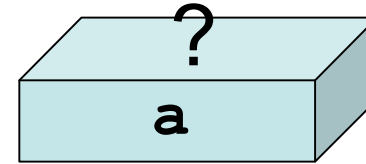
```
for(int j = 0; j < 5; j++)  
    cout << j * j << ", ";  
cout << endl;
```

**affiche:**

0, 1, 4, 9, 16,

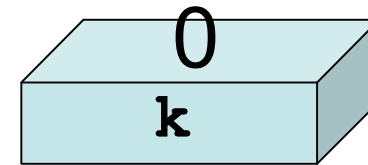
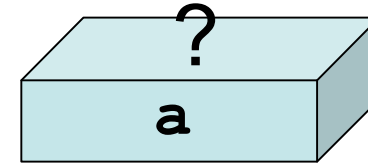
# C: Solution

```
int a;  
→ for(int k = 0; k < 5; k++)  
{  
    a = 2 * k + 1;  
    cout << a << endl;  
}
```



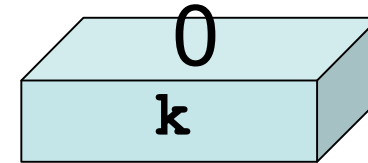
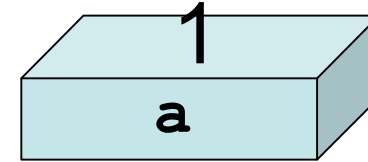
# C: Solution

```
int a;  
→ for(int k = 0; k < 5; k++)  
{  
    a = 2 * k + 1;  
    cout << a << endl;  
}
```



# C: Solution

```
int a;  
for(int k = 0; k < 5; k++)  
{  
    a = 2 * k + 1;  
    cout << a << endl;  
}
```



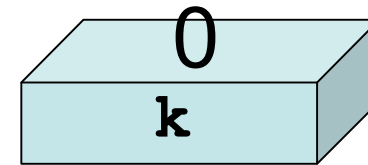
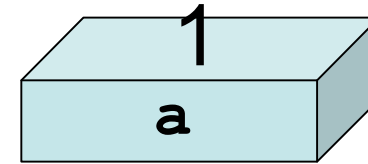
# C: Solution

```
int a;  
for(int k = 0; k < 5; k++)  
{  
    a = 2 * k + 1;  
    cout << a << endl;  
}
```



affiche:

1

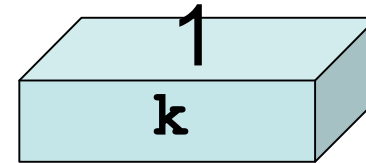
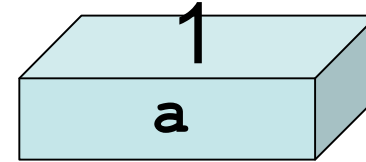


# C: Solution

```
int a;  
→ for(int k = 0; k < 5; k++)  
{  
    a = 2 * k + 1;  
    cout << a << endl;  
}
```

affiche:

1

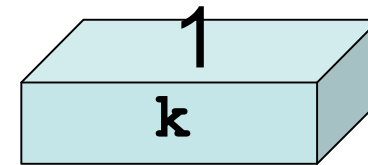
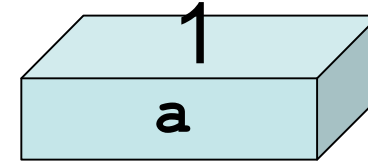


# C: Solution

```
int a;  
→ for(int k = 0; k < 5; k++)  
{  
    a = 2 * k + 1;  
    cout << a << endl;  
}
```

affiche:

1



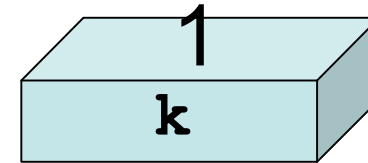
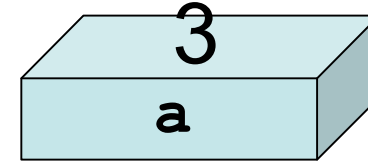
# C: Solution

```
int a;  
for(int k = 0; k < 5; k++)  
{  
    a = 2 * k + 1;  
    cout << a << endl;  
}
```



affiche:

1



# C: Solution

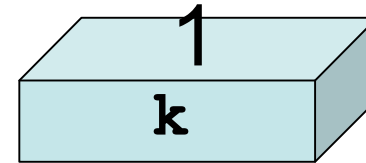
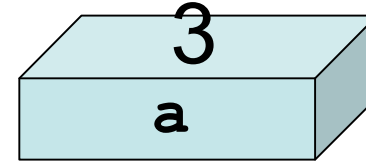
```
int a;  
for(int k = 0; k < 5; k++)  
{  
    a = 2 * k + 1;  
    cout << a << endl;  
}
```



affiche:

1

3



# C: Solution

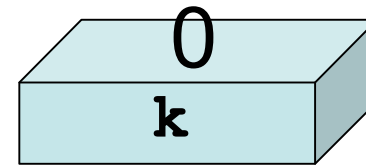
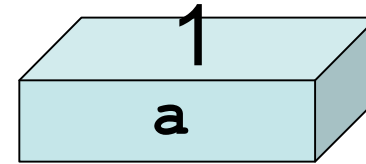
```
int a;  
for(int k = 0; k < 5; k++)  
{  
    a = 2 * k + 1;  
    cout << a << endl;  
}
```

**affiche:**

1  
3  
5  
7  
9

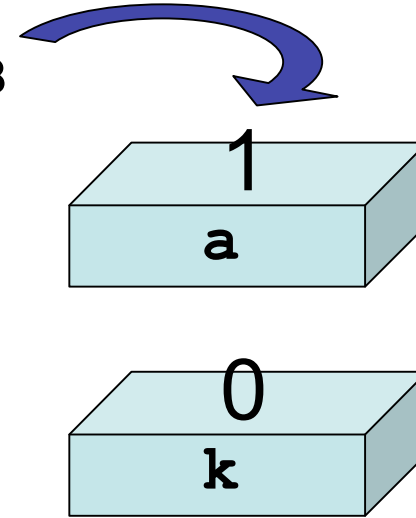
# D: Solution

```
int a = 1;  
for(int k = 0; k < 5; k++)  
{  
    a = 2 * a + 1;  
    cout << a << endl;  
}
```



# D: Solution

$$2 * a + 1 = 2 * 1 + 1 = 3$$

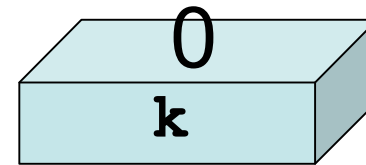
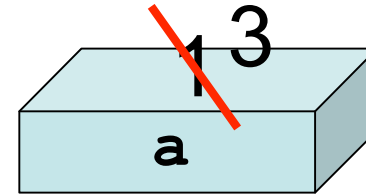


```
int a = 1;
for(int k = 0; k < 5; k++)
{
    a = 2 * a + 1;
    cout << a << endl;
}
```



# D: Solution

```
int a = 1;  
for(int k = 0; k < 5; k++)  
{  
    a = 2 * a + 1;  
    cout << a << endl;  
}
```



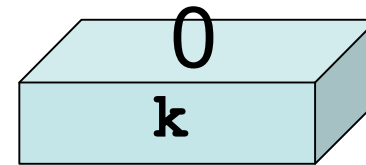
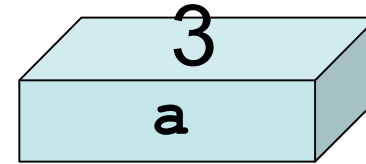
# D: Solution

```
int a = 1;  
for(int k = 0; k < 5; k++)  
{  
    a = 2 * a + 1;  
    cout << a << endl;  
}
```



affiche:

3



## D: Solution

```
int a = 1;
for(int k = 0; k < 5; k++)
{
    a = 2 * a + 1;
    cout << a << endl;
}
```

**affiche:**

3

7

15

31

63

## E : Solution

```
int a = 1;
for(int k = 0; k < 5; k++)
{
    a = 2 * a + 1;
    cout << a << endl;
    if (a % 3 == 0)
        cout << "*" << endl;
}
```

## E : Solution

```
int a = 1;
for(int k = 0; k < 5; k++)
{
    a = 2 * a + 1;
    cout << a << endl;
    if (a % 3 == 0)
        cout << "*" << endl;
}
```

**affiche:**

```
3
*
7
15
*
31
63
*
```

# Un exercice plus appliqué

**Vous allez écrire un programme qui affiche l'évolution d'une population animale, selon un modèle simple pour cette évolution:**

$$X_{n+1} = rX_n \left( 1 - \frac{X_n}{X_{\max}} \right)$$

où:

- $X_n$  est le nombre d'individus à l'année  $n$ ;
- $X_{\max}$  est la la taille maximale de la population;
- $r$  est appelé l'indice de fécondité.

Justification intuitive du modèle: la relation

$$X_{n+1} = RX_n$$

est trop simpliste, puisqu'elle résulte en une croissance exponentielle.

→ le taux de croissance  $R$  doit diminuer quand la population croît, en autre parce qu'il y a alors moins de nourriture disponible pour chaque individu. Si on prend  $R$  évoluant selon:

$$R = r \left( 1 - \frac{X_n}{X_{\max}} \right)$$

on retrouve la formule précédente:  $R$  est maximal quand  $X_n$  vaut 0, et nul quand  $X_n$  vaut  $X_{\max}$ .

# Évolution de la population

Ecrivez le programme qui affiche l'évolution d'une population animale, selon un modèle simple pour cette évolution:

$$X_{n+1} = rX_n \left( 1 - \frac{X_n}{X_{\max}} \right)$$

Le programme commencera par demander à l'utilisateur la taille initiale de la population  $X_0$  et l'indice de fécondité  $r$ . On supposera que la taille maximale  $X_{\max}$  de la population vaut 1000.

Le programme devra afficher la taille prévue de la population pour les 10 premières années.

Modifiez votre programme pour qu'il calcule la population pour 100 ans, mais n'affiche le résultat que pour les années multiples de 10.

```
Entrez la taille initiale:
```

```
20
```

```
Entrez l'indice de fecondite:
```

```
1.4
```

```
Population annee 1: 27 [= 1.4 * 20 * (1 - 20/1000)]
```

```
Population annee 2: 36 [= 1.4 * 27 * (1 - 27/1000)]
```

```
Population annee 3: 48 [= 1.4 * 36 * (1 - 36/1000)]
```

```
....
```

# Solution

```
int taille;
float r;

cout << "Entrez la taille initiale:" << endl;
cin >> taille;
cout << "Entrez l'indice de fecondite:" << endl;
cin >> r;

for(int annee = 1; annee <= 10; annee++)
{
    taille = int(r * taille * (1 - float(taille) / 1000));
    cout << " Population annee " << annee << ": " << taille << endl;
}
```

$$X_{n+1} = rX_n \left( 1 - \frac{X_n}{X_{\max}} \right)$$

# Solution

```
int taille;
float r;

cout << "Entrez la taille initiale:" << endl;
cin >> taille;
cout << "Entrez l'indice de fecondite:" << endl;
cin >> r;

for(int annee = 1; annee <= 100; annee++)
{
    taille = int(r * taille * (1 - float(taille) / 1000));
    if (annee % 10 == 0)
        cout << " Population annee " << annee << ": " << taille << endl;
}
```

# Qu'affichent les programmes suivants ?

**F:**

```
for(int i = 0; i < 5; i++)
{
    for(int j = 0; j < 5; j++)
        cout << "*";
    cout << endl;
}
```

**G:**

```
for(int i = 0; i < 5; i++)
{
    for(int j = 0; j < i; j++)
        cout << "*";
    cout << endl;
}
```

**H:**

```
for(int i = 0; i < 5; i++)
{
    for(int j = 0; j < 5; j++)
        if (i == j)
            cout << "*";
        else
            cout << " ";
    cout << endl;
}
```

# F: Solution

**F:**


```
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        cout << "*";  
    cout << endl;  
}
```

Affichage:

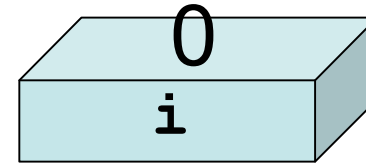


# F: Solution

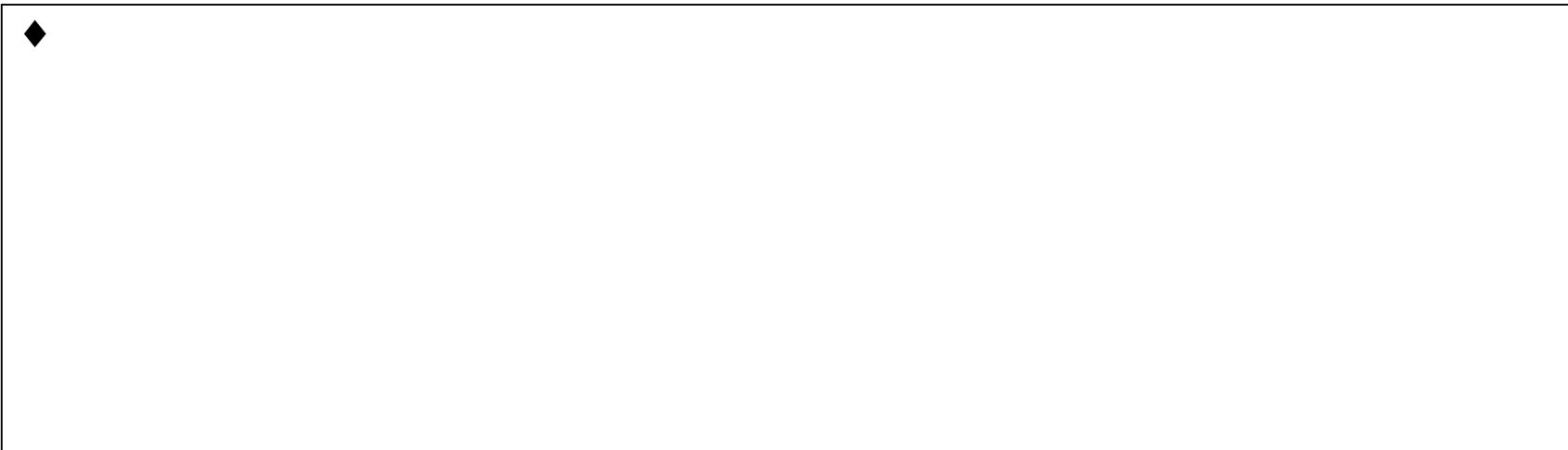
**F:**



```
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        cout << "*";  
    cout << endl;  
}
```




Affichage:

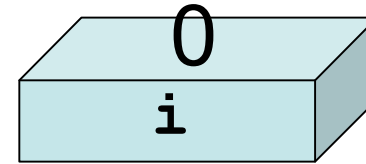


# F: Solution

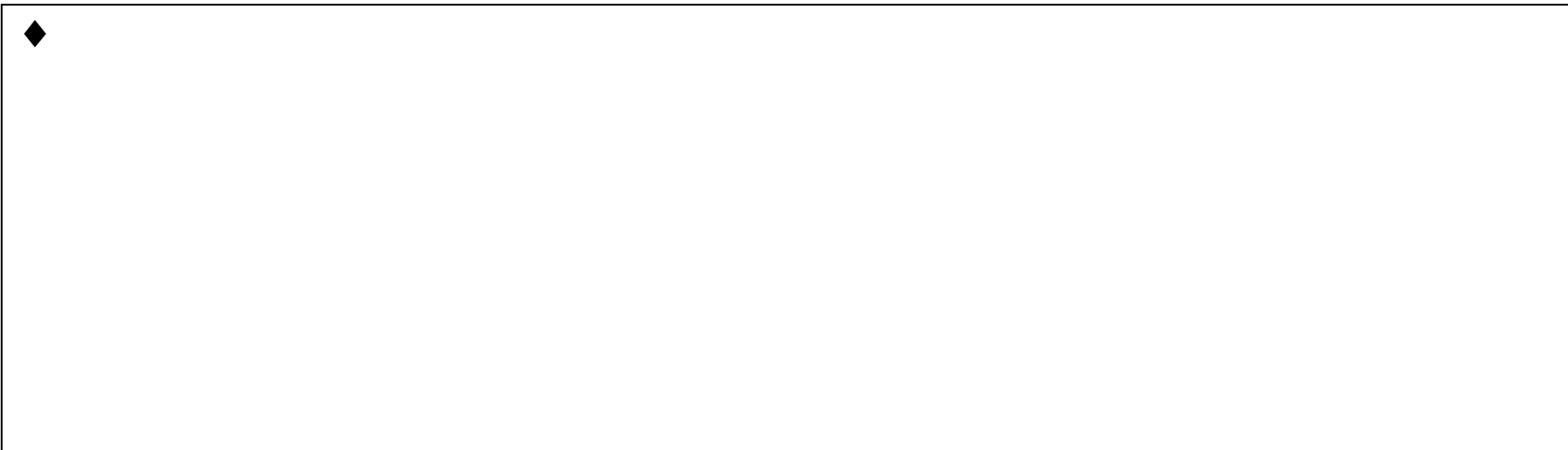
**F:**



```
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        cout << "*";  
    cout << endl;  
}
```

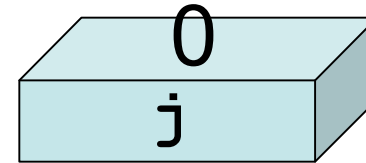
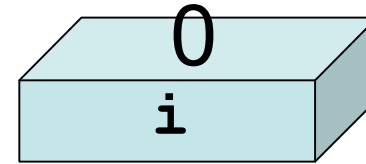


Affichage:

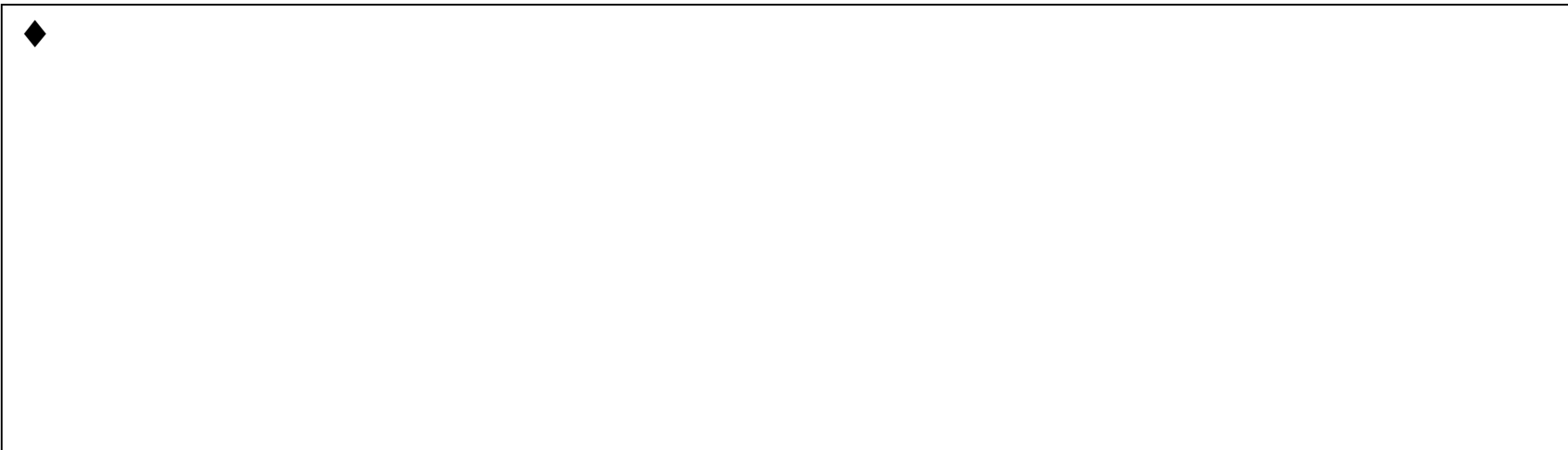


# F: Solution

```
F:  
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        cout << "*";  
    cout << endl;  
}
```

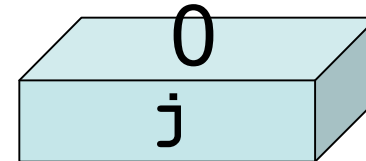
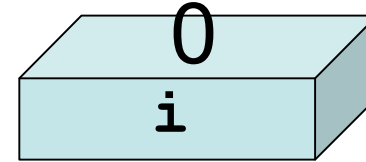


Affichage:

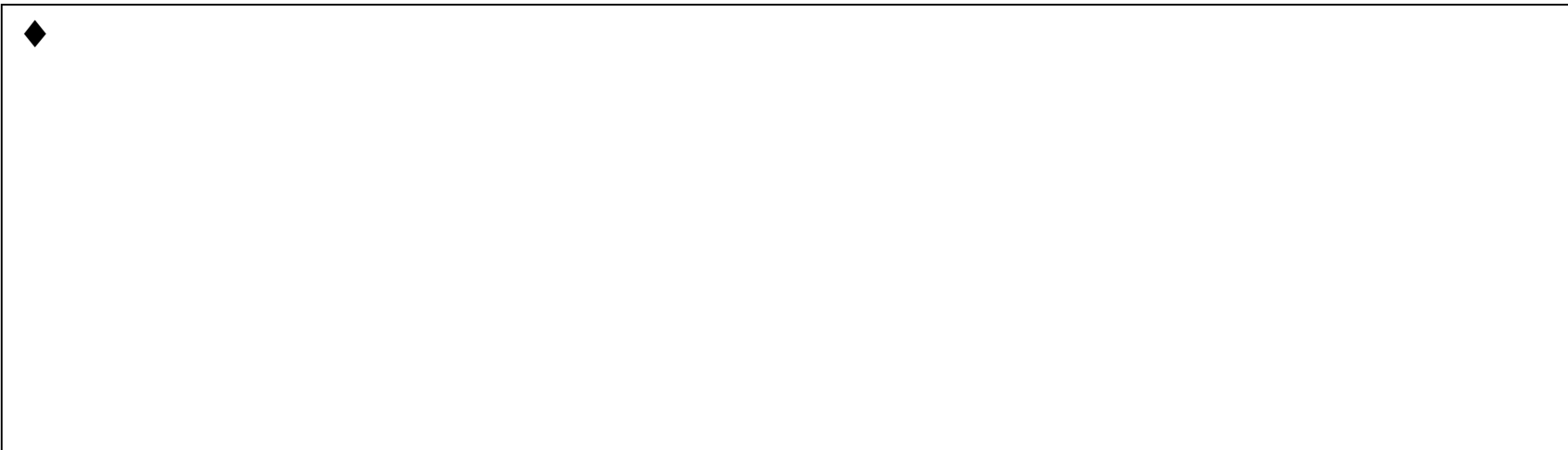


# F: Solution

```
F:  
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        cout << "*";  
    cout << endl;  
}
```

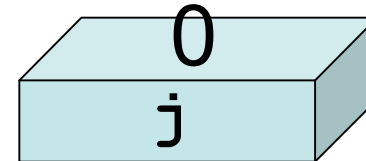
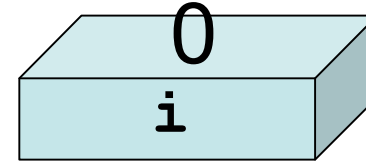


Affichage:



# F: Solution

```
F:  
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        cout << "*";  
    cout << endl;  
}
```

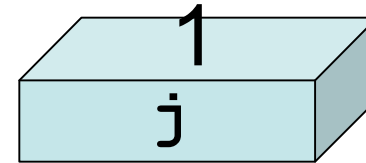
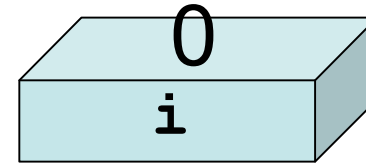


Affichage:

\*◆

# F: Solution

```
F:  
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        cout << "*";  
    cout << endl;  
}
```

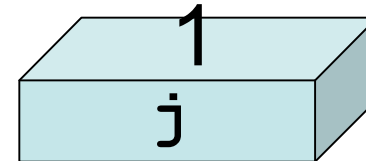
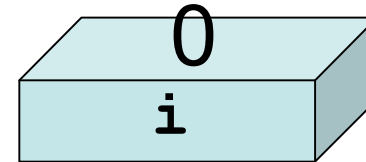


Affichage:

\* ◆

# F: Solution

```
F:  
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        cout << "*";  
    cout << endl;  
}
```

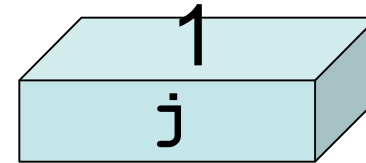
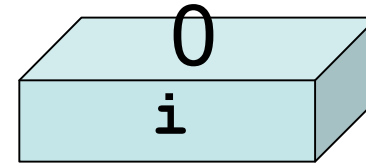


Affichage:

\*◆

# F: Solution

```
F:  
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        cout << "*";  
    cout << endl;  
}
```



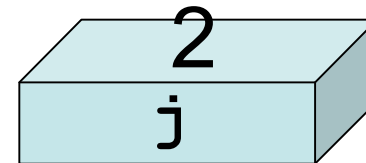
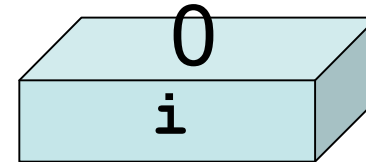
Affichage:

```
**◆
```

# F: Solution

**F:**

```
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        cout << "*";  
    cout << endl;  
}
```



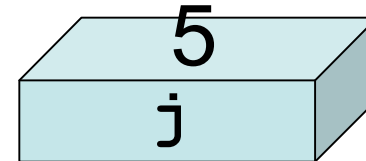
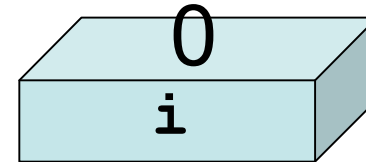
Affichage:

\*\*◆

# F: Solution

**F:**

```
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        cout << "*";  
    cout << endl;  
}
```

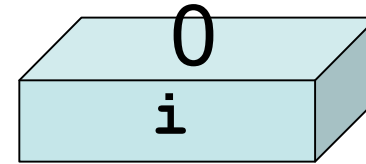


Affichage:

```
*****◆
```

# F: Solution

```
F:  
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        cout << "*";  
    cout << endl;  
}
```




Affichage:

```
*****
```

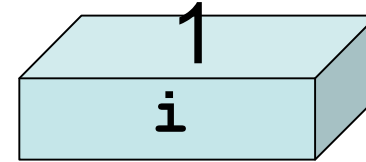


# F: Solution

**F:**



```
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        cout << "*";  
    cout << endl;  
}
```




**Affichage:**

```
*****
```

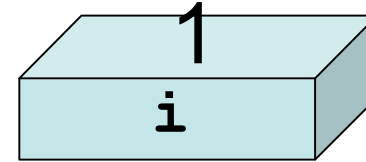


# F: Solution

**F:**



```
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        cout << "*";  
    cout << endl;  
}
```



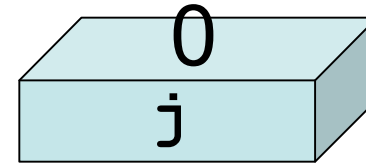
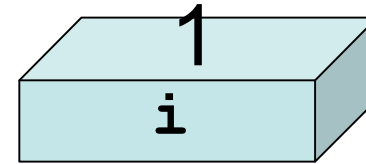
**Affichage:**

```
*****
```



# F: Solution

```
F:  
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        cout << "*";  
    cout << endl;  
}
```



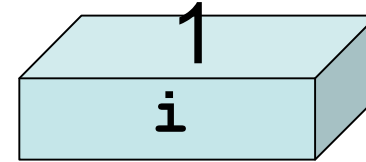
Affichage:

```
*****
```



# F: Solution

```
F:  
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        cout << "*";  
    cout << endl;  
}
```



Affichage:

\*\*\*\*\*

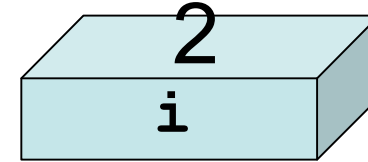
\*\*\*\*\*



# F: Solution



```
F:  
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        cout << "*";  
    cout << endl;  
}
```



Affichage:

```
*****  
*****  
◆
```

# F: Solution

```
F:  
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        cout << "*";  
    cout << endl;  
}
```



Affichage:

```
*****  
*****  
*****  
*****  
*****  
◆
```

# G: Solution

**G:**

```
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < i; j++)  
        cout << "*";  
    cout << endl;  
}
```

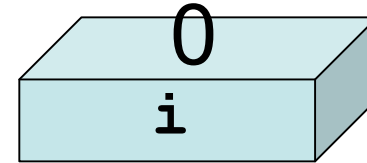
Affichage:



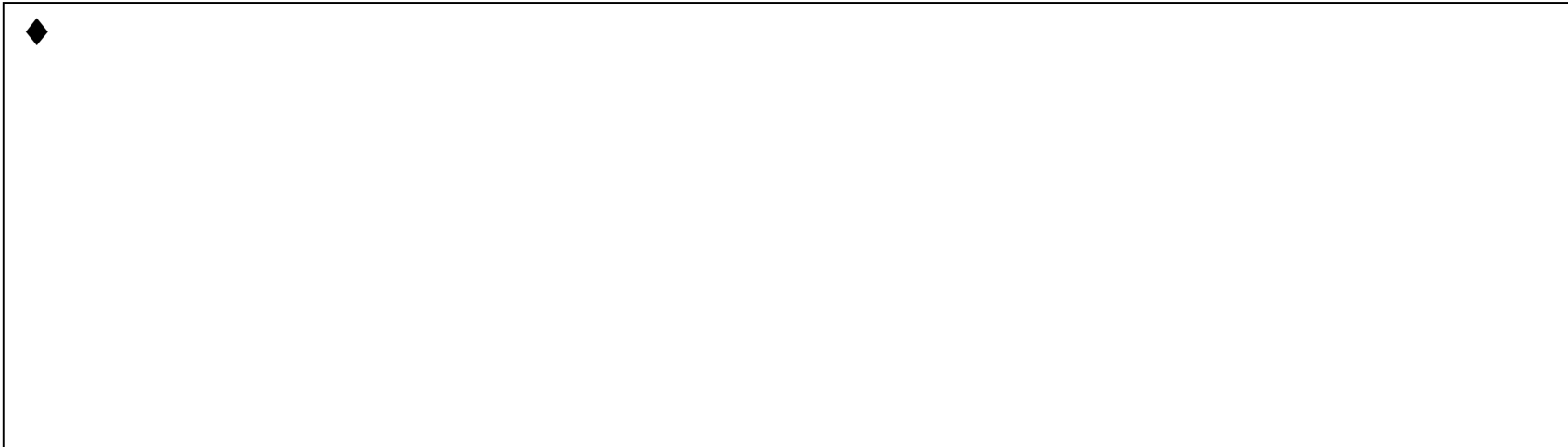
# G: Solution



```
G:  
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < i; j++)  
        cout << "*";  
    cout << endl;  
}
```

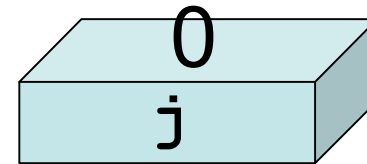
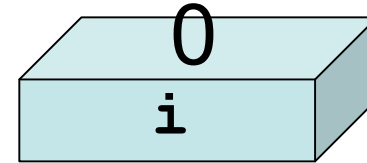


Affichage:

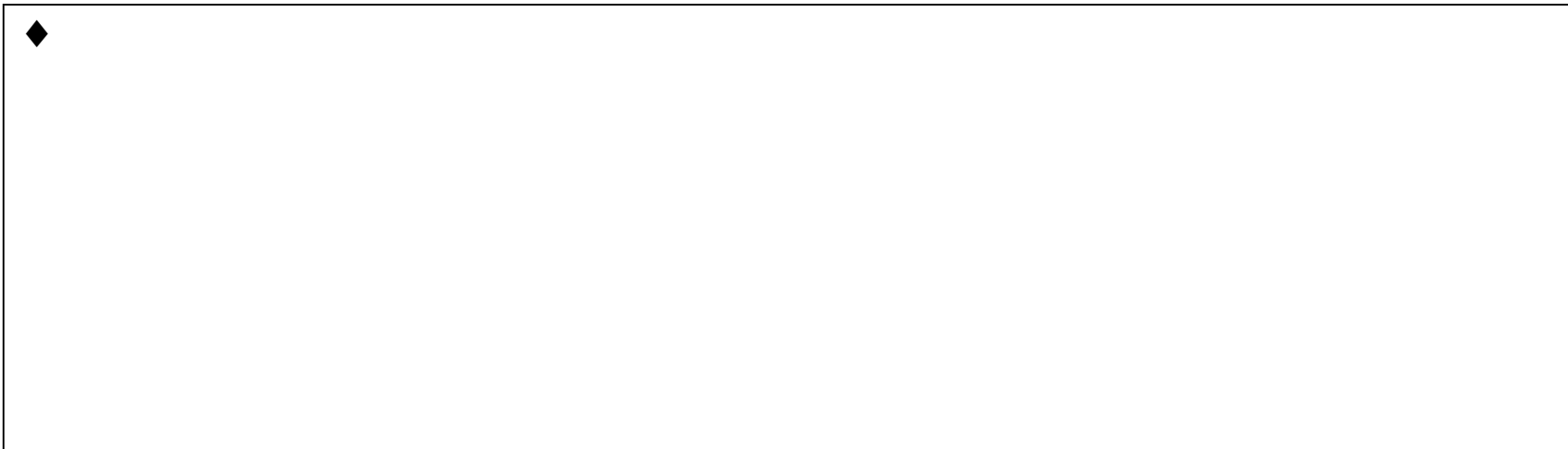


# G: Solution

```
G:  
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < i; j++)  
        cout << "*";  
    cout << endl;  
}
```

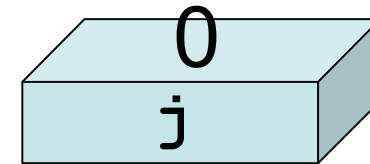
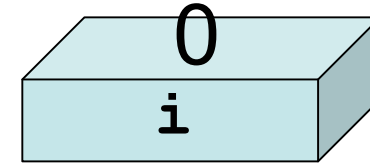


Affichage:

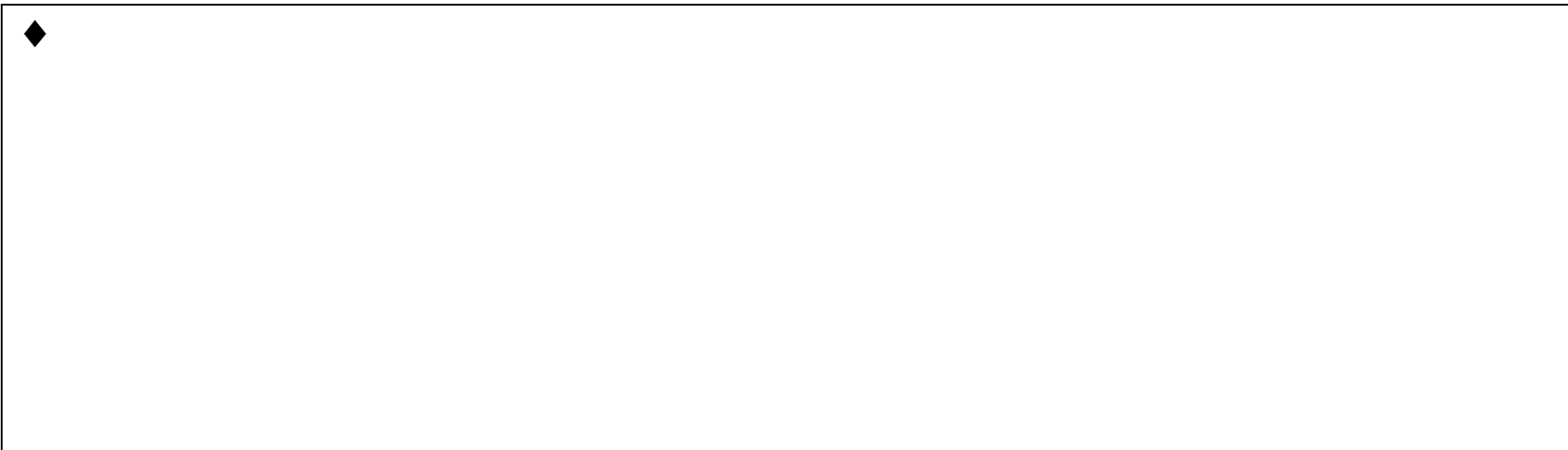


# G: Solution

```
G:  
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < i; j++)  
        cout << "*";  
    cout << endl;  
}
```

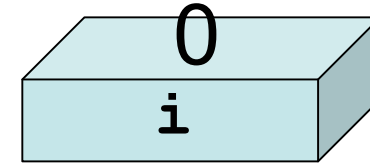


Affichage:

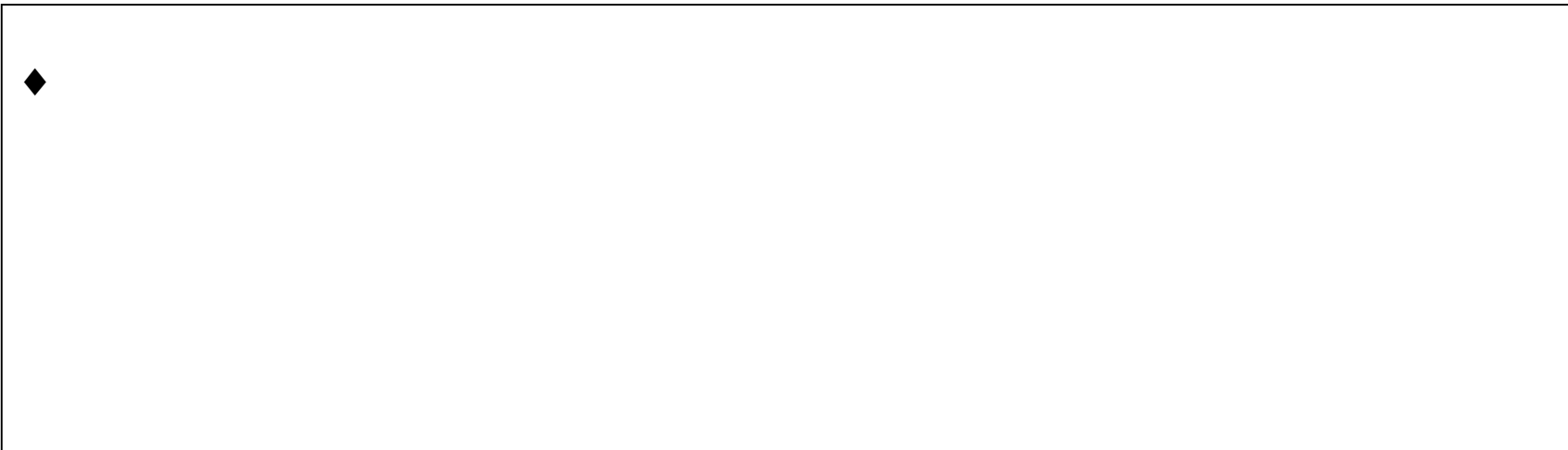


# G: Solution

```
G:  
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < i; j++)  
        cout << "*";  
    cout << endl;  
}
```

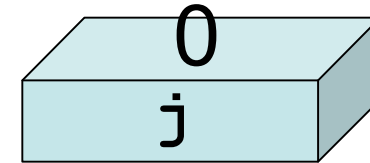
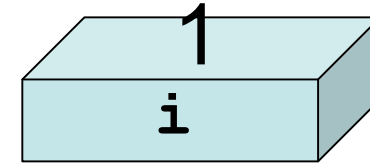


Affichage:

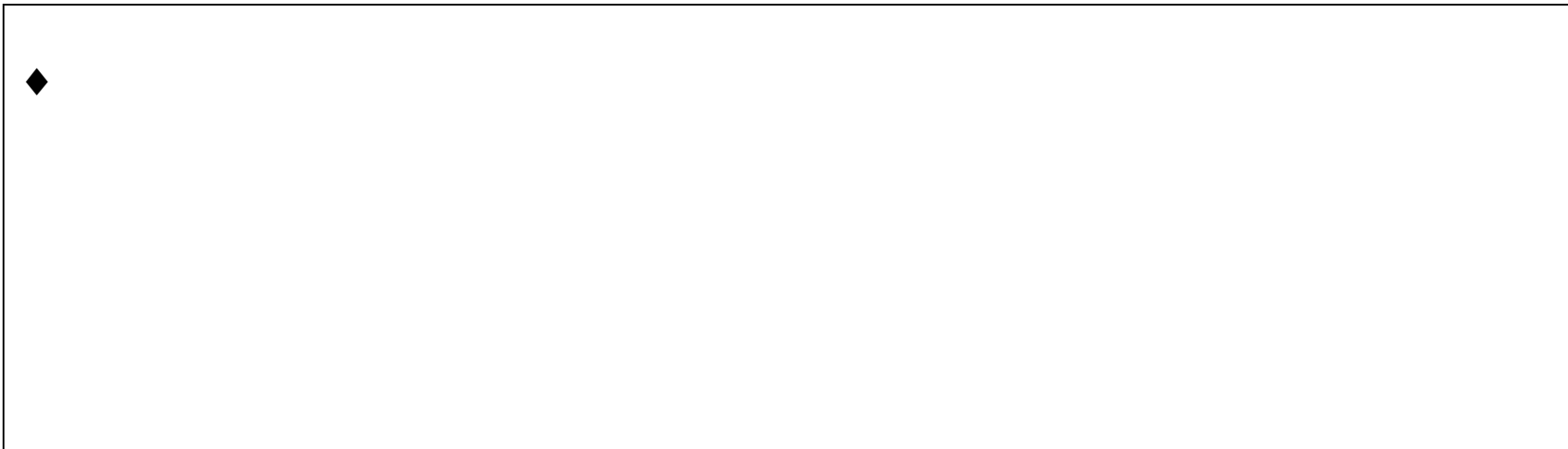


# G: Solution

```
G:  
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < i; j++)  
        cout << "*";  
    cout << endl;  
}
```



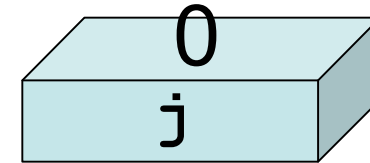
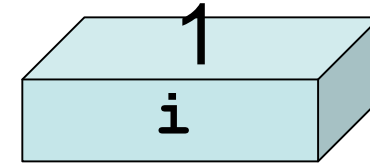
Affichage:



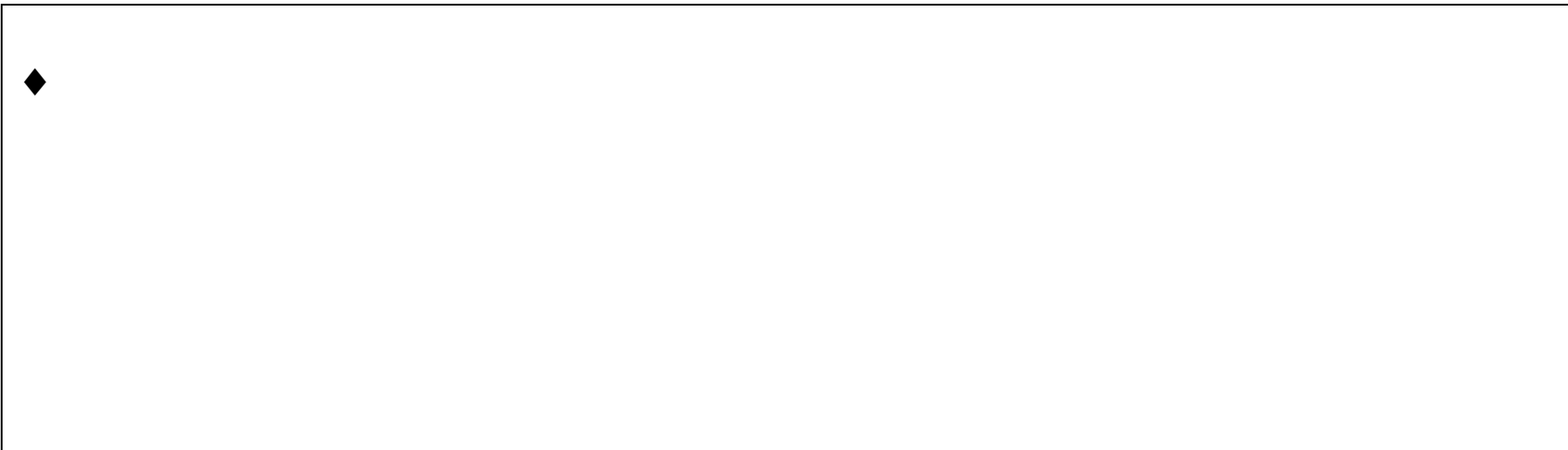
# G: Solution

**G:**

```
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < i; j++)  
        cout << "*";  
    cout << endl;  
}
```

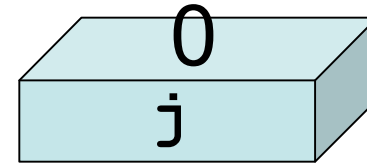
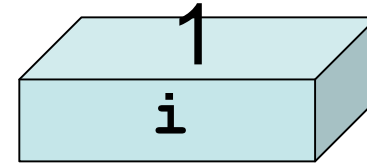


Affichage:



# G: Solution

```
G:  
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < i; j++)  
        cout << "*";  
    cout << endl;  
}
```



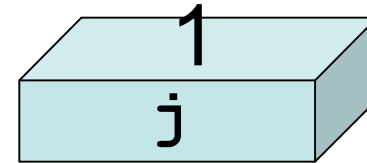
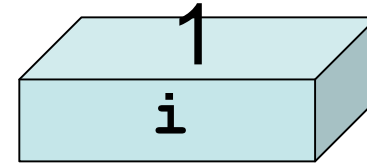
Affichage:

\* ◆

# G: Solution

**G:**

```
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < i; j++)  
        cout << "*";  
    cout << endl;  
}
```



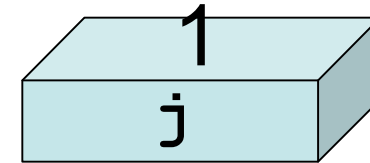
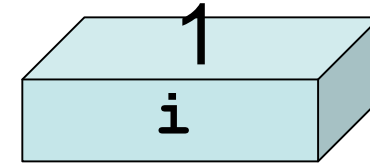
Affichage:

\* ◆

# G: Solution

**G:**

```
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < i; j++)  
        cout << "*";  
    cout << endl;  
}
```

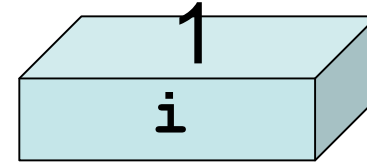


Affichage:

\*◆

# G: Solution

```
G:  
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < i; j++)  
        cout << "*";  
    cout << endl;  
}
```



Affichage:

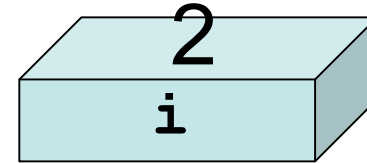
\*



# G: Solution



```
G:  
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < i; j++)  
        cout << "*";  
    cout << endl;  
}
```



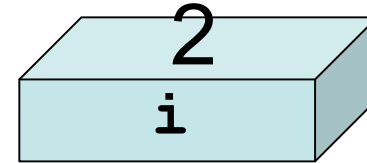
Affichage:

\*



# G: Solution

```
G:  
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < i; j++)  
        cout << "*";  
    cout << endl;  
}
```



Affichage:

```
*  
**  
◆
```

# G: Solution

**G:**

```
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < i; j++)  
        cout << "*";  
    cout << endl;  
}
```



**Affichage:**

```
*  
**  
***  
****  
◆
```

# H : Solution

**H:**

```
for(int i = 0; i < 5; i++)
{
    for(int j = 0; j < 5; j++)
        if (i == j)
            cout << "*";
        else
            cout << " ";
    cout << endl;
}
```

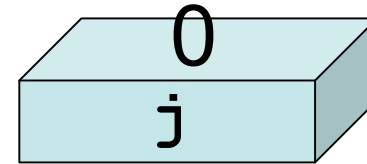
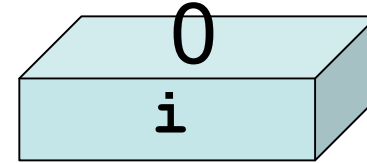
Affichage:



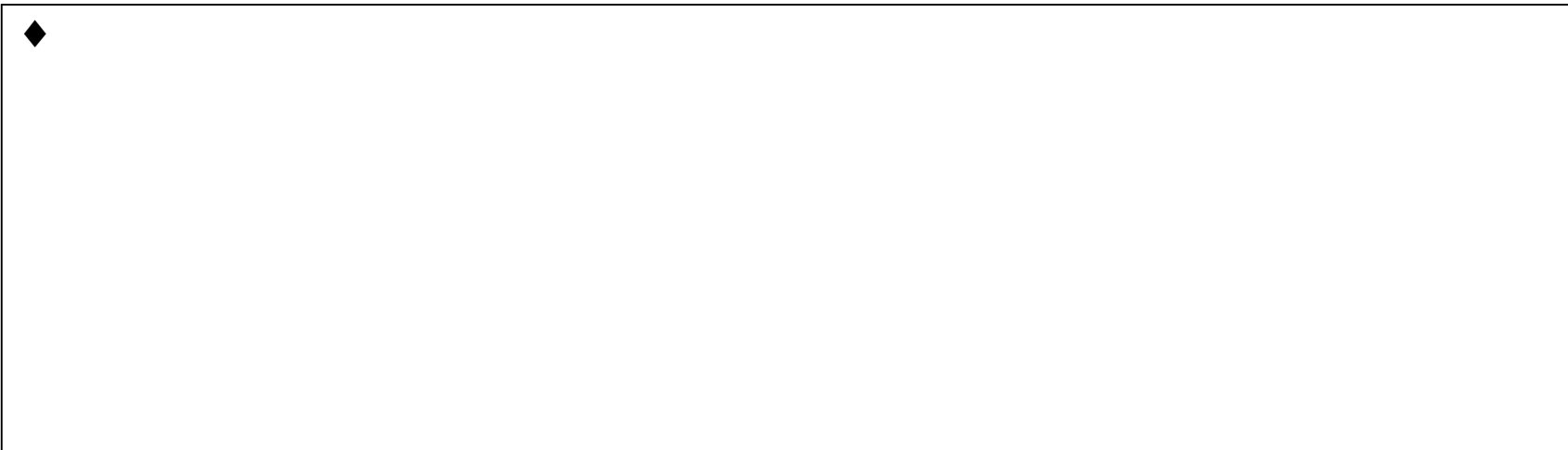
# H : Solution

**H:**

```
for(int i = 0; i < 5; i++)
{
    for(int j = 0; j < 5; j++)
        if (i == j)
            cout << "*";
        else
            cout << " ";
    cout << endl;
}
```



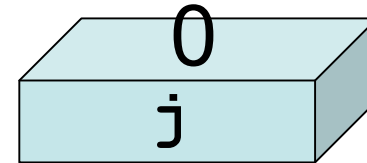
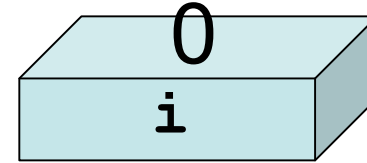
Affichage:



# H : Solution

**H:**

```
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        if (i == j)  
            cout << "*";  
        else  
            cout << " ";  
    cout << endl;  
}
```



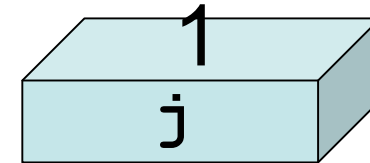
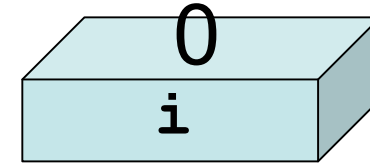
Affichage:

\*◆

# H : Solution

**H:**

```
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        if (i == j)  
            cout << "*";  
        else  
            cout << " ";  
    cout << endl;  
}
```



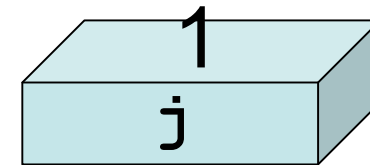
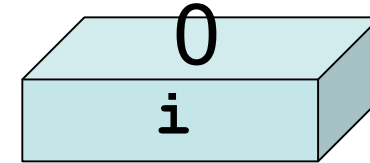
Affichage:

\*◆

# H : Solution

**H:**

```
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        if (i == j)  
            cout << "*";  
        else  
            cout << " ";  
    cout << endl;  
}
```



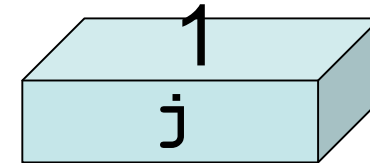
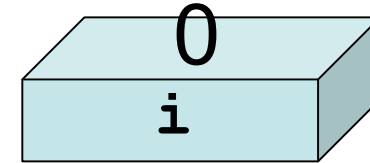
Affichage:

\*◆

# H : Solution

**H:**

```
for(int i = 0; i < 5; i++)
{
    for(int j = 0; j < 5; j++)
        if (i == j)
            cout << "*";
        else
            cout << " ";
    cout << endl;
}
```



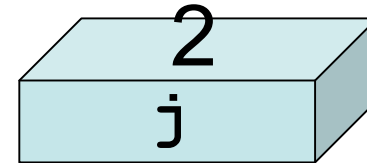
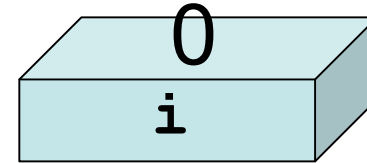
Affichage:

```
* _ ◆
```

# H : Solution

**H:**

```
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        if (i == j)  
            cout << "*";  
        else  
            cout << " ";  
    cout << endl;  
}
```



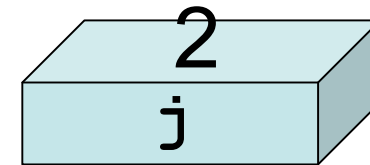
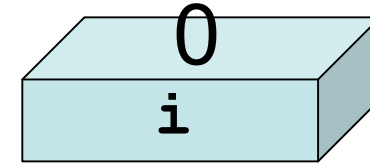
Affichage:

```
* _ ◆
```

# H : Solution

**H:**

```
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        if (i == j)  
            cout << "*";  
        else  
            cout << " ";  
    cout << endl;  
}
```



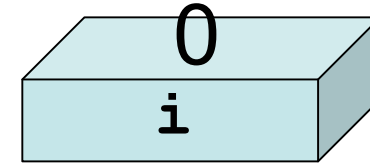
Affichage:

```
* _ ◆
```

# H : Solution

**H:**

```
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        if (i == j)  
            cout << "*";  
        else  
            cout << " ";  
    cout << endl;  
}
```



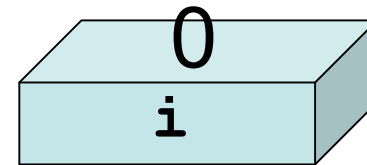
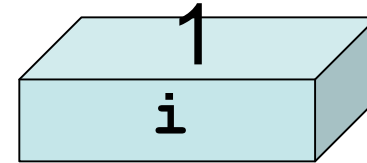
Affichage:

```
* _ _ _ _  
◆
```

# H : Solution

**H:**

```
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        if (i == j)  
            cout << "*";  
        else  
            cout << " ";  
    cout << endl;  
}
```



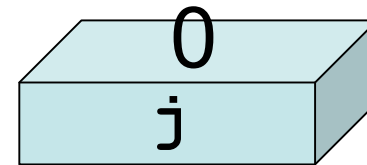
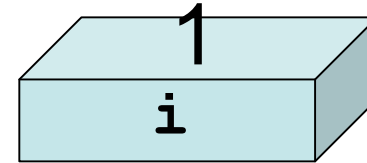
Affichage:

```
* - - - -  
◆
```

# H : Solution

H:

```
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        if (i == j)  
            cout << "*";  
        else  
            cout << " ";  
    cout << endl;  
}
```



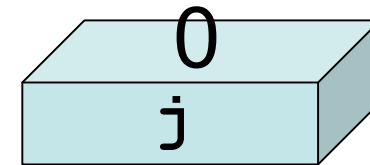
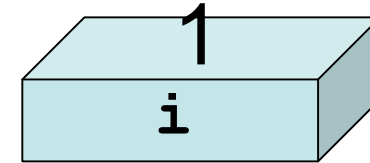
Affichage:

```
* - - - -  
◆
```

# H : Solution

**H:**

```
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        if (i == j)  
            cout << "*";  
        else  
            cout << " ";  
    cout << endl;  
}
```



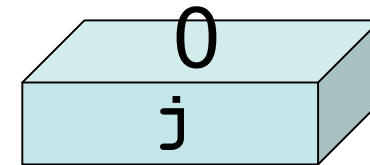
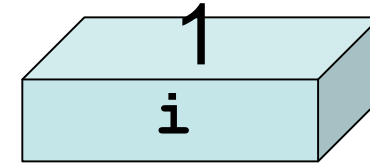
Affichage:

```
* - - - -  
◆
```

# H : Solution

**H:**

```
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        if (i == j)  
            cout << "*";  
        else  
            cout << " ";  
    cout << endl;  
}
```



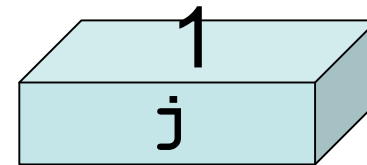
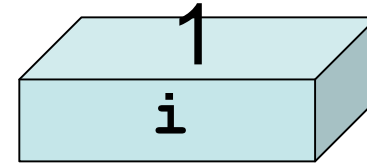
Affichage:

```
* _ _ _ _  
- ◆
```

# H : Solution

**H:**

```
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        if (i == j)  
            cout << "*";  
        else  
            cout << " ";  
    cout << endl;  
}
```



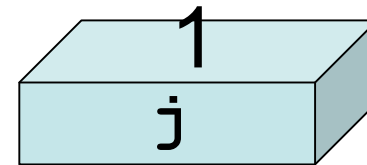
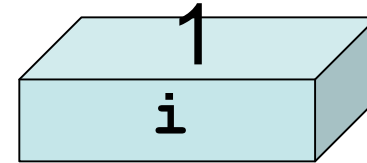
Affichage:

```
* _ _ _ _  
- ◆
```

# H : Solution

**H:**

```
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        if (i == j)  
            cout << "*";  
        else  
            cout << " ";  
    cout << endl;  
}
```



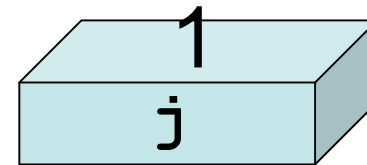
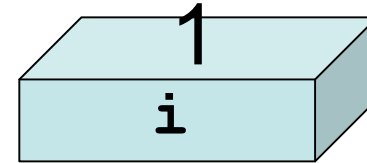
Affichage:

```
* _ _ _ _  
- ◆
```

# H : Solution

**H:**

```
for(int i = 0; i < 5; i++)
{
    for(int j = 0; j < 5; j++)
        if (i == j)
            cout << "*";
        else
            cout << " ";
    cout << endl;
}
```



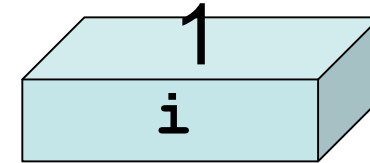
Affichage:

```
* _ _ _ _
_ * ◆
```

# H : Solution

**H:**

```
for(int i = 0; i < 5; i++)
{
    for(int j = 0; j < 5; j++)
        if (i == j)
            cout << "*";
        else
            cout << " ";
    cout << endl;
}
```

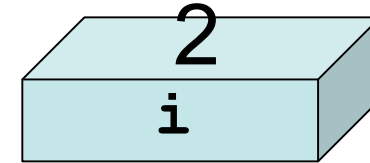


Affichage:

```
* - - - -
- * - - - -
◆
```

# H : Solution

```
H:  
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        if (i == j)  
            cout << "*";  
        else  
            cout << " ";  
    cout << endl;  
}
```



Affichage:

```
* _ _ _ _  
_ * _ _ _  
_ _ * _ _ ◆
```

# H : Solution

**H:**

```
for(int i = 0; i < 5; i++)
{
    for(int j = 0; j < 5; j++)
        if (i == j)
            cout << "*";
        else
            cout << " ";
    cout << endl;
}
```



**Affichage:**

```
* _ _ _ _
_ * _ _ _
_ _ * _ _
_ _ _ * _
_ _ _ _ *
◆
```

# H : Solution

```
H:  
for(int i = 0; i < 5; i++)  
{  
    for(int j = 0; j < 5; j++)  
        if (i == j)  
            cout << "*";  
        else  
            cout << " ";  
    cout << endl;  
}
```



Affichage:

```
*  
 *  
  *  
   *  
    *  
◆
```