Instruction `break` : exemple

Exemple d'utilisation de `break` :
une **mauvaise** (!) façon de simuler une boucle avec condition d'arrêt

```
while (true) {
  Instruction 1;
  ...
  if (condition d'arrêt)
      break;
}
autres instructions;
```

Question : quelle est la bonne façon d'écrire le code ci-dessus ?

```
do {
  Instruction 1;
  ...
} while not (condition d'arrêt);
autres instructions;
```


Jean-Cédric Chappelier
& Jamila Sam
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Instruction `continue` : exemple

Exemple d'utilisation de `continue` :

```
{
  int i(0);
  while (i < 100) {
    ++i;
    if ((i % 2) == 0) continue;
    // la suite n'est exécutée que pour les
    // entiers impairs !
    Instructions;
    ...
  }
}
```

Question : quelle est une meilleure façon d'écrire le code ci-dessus ?
(on suppose que *`Instructions; ...`* ne modifie pas la valeur de `i`)

```
for (int i(1); i < 100; i += 2) {
  Instructions;
  ...
}
```

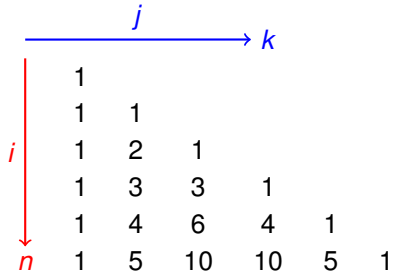

Jean-Cédric Chappelier
& Jamila Sam
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Etude de cas : les $\binom{n}{k}$

Comment commencer le calcul des $\binom{n}{k}$ tel que vu en cours ICC ?
(version « programmation dynamique » ; on ne fait ici que le début, nous continuerons lorsque nous aurons vu les tableaux en C++)

Rappel du problème :

- on nous donne n et k
- on veut calculer $\binom{n}{k}$ par deux boucles sur le « triangle de Pascal » :



Jean-Cédric Chappelier
& Jamila Sam
ECOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Etude de cas : les $\binom{n}{k}$

- On va donc commencer par une boucle en i (de 1 en 1) :
  `for (int i(...); i ...; ++i)`
- Dans laquelle on aura une boucle en j (de 1 en 1) :
  `for (int j(...); j ...; ++i)`

Quelles bornes pour i ? ☞ entre 0 et n

Quelles bornes pour j ? ☞ entre 0 et i

On a donc à ce stade :

```
for (int i(0); i <= n; ++i) {
  for (int j(0); j <= i; ++j) {

  }
 }
```


Jean-Cédric Chappelier
& Jamila Sam
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Etude de cas : les $\binom{n}{k}$

Peut-on faire mieux ?

☞ Oui : il **n'**est **pas** nécessaire d'aller au delà de k pour la boucle en j

Donc veut donc que j soit inférieur à i mais sans non plus dépasser k.

Comment cela s'écrit-il ?

```
(j <= i) or (j <= k)
```

ou bien

```
(j <= i) and (j <= k)
```

?

Jean-Cédric Chappelier
& Jamila Sam
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

`or` ou `and` ?

Il est souvent difficile de correctement choisir entre la conjonction `and` et la disjonction `or`. Quelques pistes :

- tout d'abord que veut-on : que la condition soit vraie ou qu'elle soit fausse ?
  - ☞ Ici on veut *continuer* tant qu'elle est *vraie*
- parfois il est plus facile de passer par la contraposée (sa négation) :
  - ☞ Ici on veut *s'arrêter* dès que la condition est *fausse* ;
    dès que j est plus grand que i ou k (...donc sa négation : et)
- tester pour un cas ambigu.
  - ☞ Ici : tester dans un cas où justement j est entre k et i pour i plus grand que k (puisque c'est bien ces cas là que nous voulons optimiser)
    dans ce cas ($k < j < i$) :
    - `(j <= i) or (j <= k)` est vraie (`true or false`), la boucle continuera donc ; ce qui n'est pas ce que nous voulons ;
    - `(j <= i) and (j <= k)` est fausse (`true and false`), la boucle aura donc été arrêtée ; ce que nous voulons.

☞ Ici c'est donc bien « `(j <= i) and (j <= k)` »

Etude de cas : les $\binom{n}{k}$

On obtient donc :

```
for (int i(0); i <= n; ++i) {
  for (int j(0); (j <= i) and (j <= k); ++j) {
    // ...
  }
}
```

On pourrait aussi écrire (avec `#include <algorithm>`) :

```
for (int i(0); i <= n; ++i) {
  for (int j(0); j <= min(i, k); ++j) {
    // ...
  }
}
```

Reste à voir quoi mettre dans la boucle...
...ce que nous aborderons plus tard une fois vus les tableaux.

Jean-Cédric Chappelier
& Jamila Sam
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Devinette

```
#include <iostream>
using namespace std;

int main()
{
  int min(1);
  int max(100);

  cout << "Pensez à un nombre entre "
       << min << " et " << max
       << endl;

  // boucle conditionnelle a posteriori
  do {
    int pivot( (min+max) / 2 );

    cout << "Votre nb est il < > = à "
         << pivot << " ? ";
    char rep;
    cin >> rep;
    if (rep == '>') {
      min = pivot;
    } else if (rep == '<') {
      max = pivot;
    } else {
      cout << "J'ai trouvé : "
           << pivot << endl;
      min = max = pivot;
    }

  } while (min < max);

  return 0;
}
```


Jean-Cédric Chappelier
& Jamila Sam
ECOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE