

The background image shows a waterfall in a dense forest. The water flows over several layers of dark, reddish-brown rocks, creating a series of small pools. The surrounding trees are lush and green, with some ferns visible. A blue rectangular frame covers the lower half of the image, containing the text.

Flux de Contrôle

Avant de commencer:

Des questions sur Karel?

Elle connaît quatre commandes:

move(); bougez

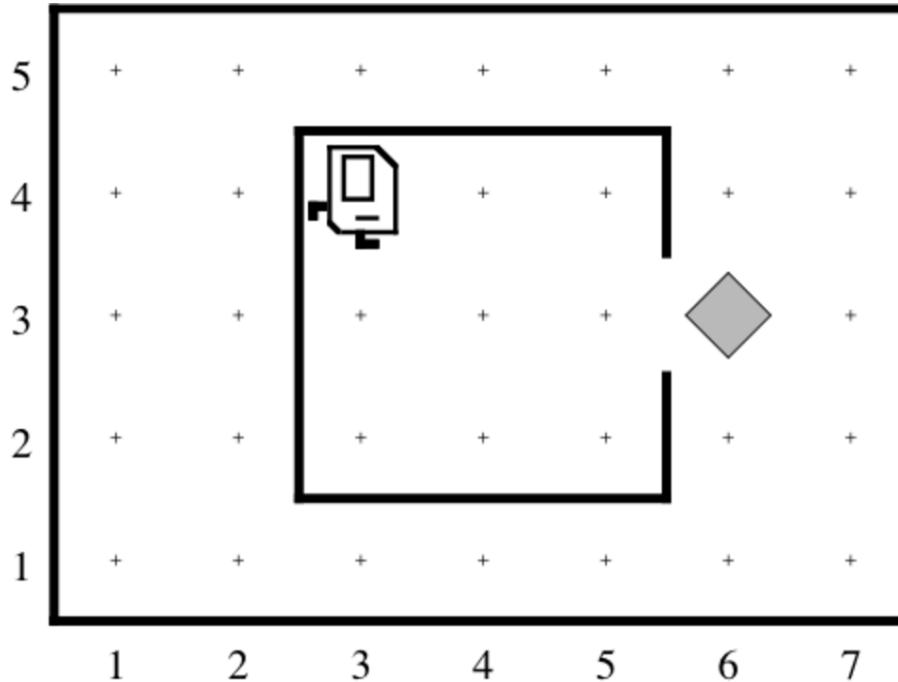
turnLeft(); tournez à gauche

putBeeper(); mettre le bip

pickBeeper(); prendre le bip



Puiser de l'eau Karel



Avez-vous eu du mal à accomplir cette tâche?

On va réviser l'ordre des tâches.

Comment l'ordinateur a-t-il exécuté ce que vous avez écrit?

```
public void run() {  
bougezALaPorte();  
puisezlEau();  
returnnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezlEau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}}
```

```
public void run() {  
    bougezALaPorte();  
    puisezEau();  
    returnezDansLaMaison();  
}
```

Cette partie du *code source* s'appelle une *méthode*.

```
private void tournezADroite() {  
    turnLeft();  
    turnLeft();  
    turnLeft();  
}
```

Cette partie du *code source* s'appelle une *méthode*.

```
private void bougezALaPorte() {  
    move();  
    move();  
    tournezADroite();  
    move();  
    turnLeft();  
    move();  
}
```

Cette partie du *code source* s'appelle une *méthode*.

```
private void puisezEau() {  
    pickBeeper(); //prendre le bip  
}
```

Cette partie du *code source* s'appelle une *méthode*.

```
private void returnezDansLaMaison() {  
    turnLeft();  
    turnLeft();  
    move();  
    tournezADroite();  
    move();  
    turnLeft();  
    move();  
    move();  
    turnLeft();  
    turnLeft();  
}
```

Cette partie du *code source* s'appelle une *méthode*.

Les noms des méthodes

private void turnez A Droite() { 

private void turnezA Droite() { 

private void turnezADroite() {

Pas d'espaces!

Les noms des méthodes

private void puisez l'Eau() {



private void puisez lEau() {



private void puisez lEau() {

Pas d'espace!

Pas d'apostrophe!

```
public void run() {  
    bougezALaPorte();  
    puisezLEau();  
    returnnezDansLaMaison();  
}  
  
private void tournezADroite() {  
    turnLeft();  
    turnLeft();  
    turnLeft();  
}  
  
private void bougezALaPorte() {  
    move();  
    move();  
    tournezADroite();  
    move();  
    turnLeft();  
    move();  
}  
  
private void puisezLEau() {  
    pickBeeper(); //prendre le bip  
}  
  
private void returnnezDansLaMaison() {  
    turnLeft();  
    turnLeft();  
    move();  
    tournezADroite();  
    move();  
    turnLeft();  
    move();  
    move();  
    turnLeft();  
    turnLeft();  
}}}
```

```
public void run() {  
    bougezALaPorte();  
    puisezLEau();  
    returnnezDansLaMaison();  
}  
  
private void tournezADroite() {  
    turnLeft();  
    turnLeft();  
    turnLeft();  
}  
  
private void bougezALaPorte() {  
    move();  
    move();  
    tournezADroite();  
    move();  
    turnLeft();  
    move();  
}  
  
private void puisezLEau() {  
    pickBeeper(); //prendre le bip  
}  
  
private void returnnezDansLaMaison() {  
    turnLeft();  
    turnLeft();  
    move();  
    tournezADroite();  
    move();  
    turnLeft();  
    move();  
    move();  
    turnLeft();  
    turnLeft();  
}}
```

```
public void run() {  
bougezALaPorte();  
puisezL'Eau();  
returnnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezL'Eau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}}}
```

```
public void run() {  
bougezALaPorte();  
puisezEau();  
returnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezEau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}}
```

```
public void run() {  
bougezALaPorte();  
puisezEau();  
returnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezEau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}}
```

```
public void run() {  
bougezALaPorte();  
puisezL'Eau();  
returnnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezL'Eau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}}
```

```
public void run() {  
bougezALaPorte();  
puisezlEau();  
returnnezDansLaMaison();  
}  
}
```

```
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
}
```

```
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
}
```

```
private void puisezlEau() {  
pickBeeper(); //prendre le bip  
}  
}
```

```
private void returnnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}  
}
```

```
public void run() {  
bougezALaPorte();  
puisezlEau();  
returnnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezlEau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}}
```

```
public void run() {  
bougezALaPorte();  
puisezL'Eau();  
returnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezL'Eau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}}
```

```
public void run() {  
    bougezALaPorte();  
    puisezL'Eau();  
    returnnezDansLaMaison();  
}  
  
private void tournezADroite() {  
    turnLeft();  
    turnLeft();  
    turnLeft();  
}  
  
private void bougezALaPorte() {  
    move();  
    move();  
    tournezADroite();  
    move();  
    turnLeft();  
    move();  
}  
  
private void puisezL'Eau() {  
    pickBeeper(); //prendre le bip  
}  
  
private void returnnezDansLaMaison() {  
    turnLeft();  
    turnLeft();  
    move();  
    tournezADroite();  
    move();  
    turnLeft();  
    move();  
    move();  
    turnLeft();  
    turnLeft();  
}}}
```

```
public void run() {  
bougezALaPorte();  
puisezL'Eau();  
returnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezL'Eau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}}
```

```
public void run() {  
bougezALaPorte();  
puisezL'Eau();  
returnnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezL'Eau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}}
```

```
public void run() {  
bougezALaPorte();  
puisezL'Eau();  
returnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezL'Eau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}}
```

```
public void run() {  
bougezALaPorte();  
puisezL'Eau();  
returnnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezL'Eau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}}
```

```
public void run() {  
    bougezALaPorte();  
    puisezLEau();  
    returnnezDansLaMaison();  
}  
  
private void tournezADroite() {  
    turnLeft();  
    turnLeft();  
    turnLeft();  
}  
  
private void bougezALaPorte() {  
    move();  
    move();  
    tournezADroite();  
    move();  
    turnLeft();  
    move();  
}  
  
private void puisezLEau() {  
    pickBeeper(); //prendre le bip  
}  
  
private void returnnezDansLaMaison() {  
    turnLeft();  
    turnLeft();  
    move();  
    tournezADroite();  
    move();  
    turnLeft();  
    move();  
    move();  
    turnLeft();  
    turnLeft();  
}}}
```

```
public void run() {  
bougezALaPorte();  
puisezEau();  
returnezDansLaMaison();  
}
```

```
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}
```

```
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}
```

```
private void puisezEau() {  
pickBeeper(); //prendre le bip  
}
```

```
private void returnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}}
```

```
public void run() {  
bougezALaPorte();  
puisezEau();  
returnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezEau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}}
```

```
public void run() {  
bougezALaPorte();  
puisezLEau();  
returnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezLEau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}}
```

```
public void run() {  
    bougezALaPorte();  
    puisezEau();  
    returnezDansLaMaison();  
}
```

```
private void tournezADroite() {  
    turnLeft();  
    turnLeft();  
    turnLeft();  
}
```

```
private void bougezALaPorte() {  
    move();  
    move();  
    tournezADroite();  
    move();  
    turnLeft();  
    move();  
}
```

```
private void puisezEau() {  
    pickBeeper(); //prendre le bip  
}
```

```
private void returnezDansLaMaison() {  
    turnLeft();  
    turnLeft();  
    move();  
    tournezADroite();  
    move();  
    turnLeft();  
    move();  
    move();  
    turnLeft();  
    turnLeft();  
}}
```

```
public void run() {  
bougezALaPorte();  
puisezL'Eau();  
returnnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezL'Eau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}}}
```

```
public void run() {  
bougezALaPorte();  
puisezL'Eau();  
returnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezL'Eau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}}
```

```
public void run() {  
bougezALaPorte();  
puisezL'Eau();  
returnnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezL'Eau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}}
```

```
public void run() {  
bougezALaPorte();  
puisezLEau();  
returnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezLEau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
turnLeft();  
}}}
```

```
public void run() {  
bougezALaPorte();  
puisezL'Eau();  
returnnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezL'Eau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move():  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}}}
```

```
public void run() {  
bougezALaPorte();  
puisezlEau();  
returnnezDansLaMaison();  
}  
}
```

```
private void tournezADroite() {
```

```
turnLeft();  
turnLeft();  
turnLeft();  
}
```

```
private void bougezALaPorte() {
```

```
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}
```

```
private void puisezlEau() {
```

```
pickBeeper(); //prendre le bip  
}
```

```
private void returnnezDansLaMaison() {
```

```
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}}
```

```
public void run() {  
bougezALaPorte();  
puisezL'Eau();  
returnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezL'Eau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}}
```

```
public void run() {  
    bougezALaPorte();  
    puisezEau();  
    returnnezDansLaMaison();  
}  
  
private void tournezADroite() {  
    turnLeft();  
    turnLeft();  
    turnLeft();  
}  
  
private void bougezALaPorte() {  
    move();  
    move();  
    tournezADroite();  
    move();  
    turnLeft();  
    move();  
}  
  
private void puisezEau() {  
    pickBeeper(); //prendre le bip  
}  
  
private void returnnezDansLaMaison() {  
    turnLeft();  
    turnLeft();  
    move();  
    tournezADroite();  
    move();  
    turnLeft();  
    move();  
    move();  
    turnLeft();  
    turnLeft();  
}}}
```

```
public void run() {  
    bougezALaPorte();  
    puisezLEau();  
    returnnezDansLaMaison();  
}  
  
private void tournezADroite() {  
    turnLeft();  
    turnLeft();  
    turnLeft();  
}  
  
private void bougezALaPorte() {  
    move();  
    move();  
    tournezADroite();  
    move();  
    turnLeft();  
    move();  
}  
  
private void puisezLEau() {  
    pickBeeper(); //prendre le bip  
}  
  
private void returnnezDansLaMaison() {  
    turnLeft();  
    turnLeft();  
    move();  
    tournezADroite();  
    move();  
    turnLeft();  
    move();  
    move();  
    turnLeft();  
    turnLeft();  
}}}
```

```
public void run() {  
bougezALaPorte();  
puisezLEau();  
returnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezLEau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}}
```

```
public void run() {  
bougezALaPorte();  
puisezL'Eau();  
returnnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezL'Eau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}}
```

```
public void run() {  
bougezALaPorte();  
puisezL'Eau();  
returnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezL'Eau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}}
```

```
public void run() {  
bougezALaPorte();  
puisezL'Eau();  
returnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezL'Eau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}}
```

```
public void run() {  
bougezALaPorte();  
puisezL'Eau();  
returnnezDansLaMaison();  
}
```

```
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}
```

```
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}
```

```
private void puisezL'Eau() {  
pickBeeper(); //prendre le bip  
}
```

```
private void returnnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}}
```

```
public void run() {  
bougezALaPorte();  
puisezlEau();  
returnnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezlEau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move():  
    turnLeft();  
    turnLeft();  
}}
```

```
public void run() {  
bougezALaPorte();  
puisezEau();  
returnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezEau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}  
}}
```

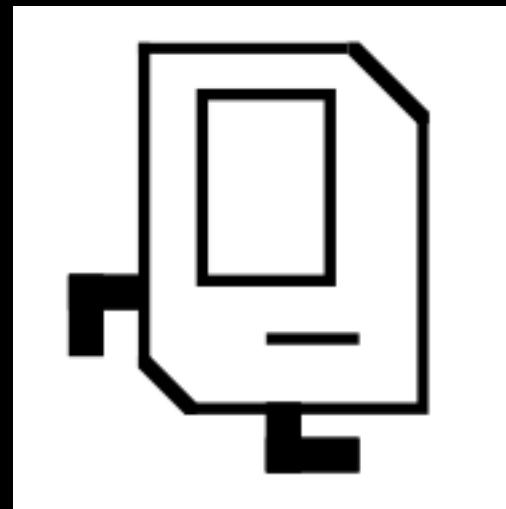
```
public void run() {  
bougezALaPorte();  
puisezL'Eau();  
returnezDansLaMaison();  
}  
  
private void tournezADroite() {  
turnLeft();  
turnLeft();  
turnLeft();  
}  
  
private void bougezALaPorte() {  
move();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
}  
  
private void puisezL'Eau() {  
pickBeeper(); //prendre le bip  
}  
  
private void returnezDansLaMaison() {  
turnLeft();  
turnLeft();  
move();  
tournezADroite();  
move();  
turnLeft();  
move();  
move();  
turnLeft();  
turnLeft();  
}  
}}
```

Maintenant on va commander Karel à faire les autres tâches pour nous

Moi, j'ai beaucoup des habilles a
lavez...

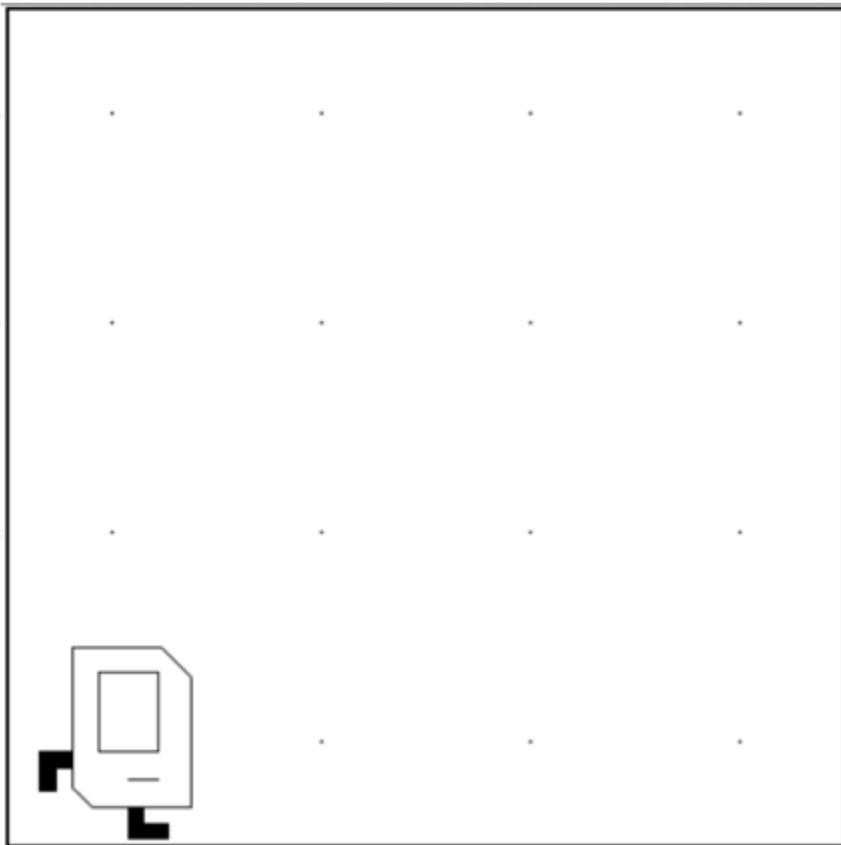
Moi, j'ai beaucoup des habilles a
lavez...
100 pièces des habilles!

Karel peut-elle m'aider?

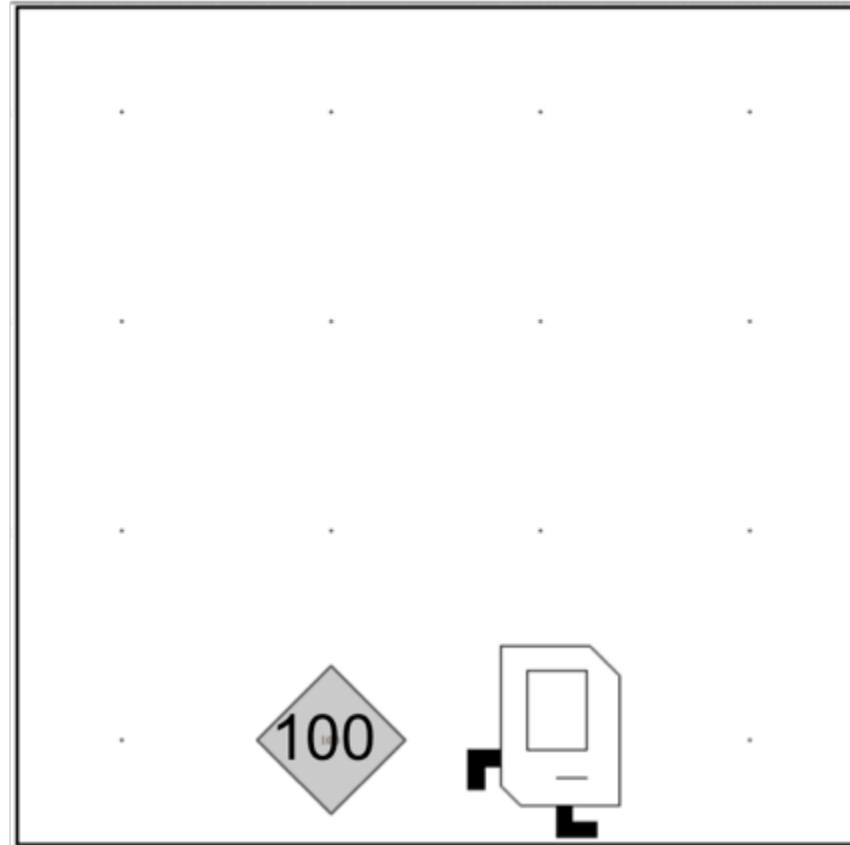


Karel, lavez 100 pièce des habilles.

Avant



Après



J'ai besoin d'un volontaire



For Loop

```
for(int i = 0; i < N; i++) {  
    // répéter N fois  
}
```

Anatomie d'un For Loop

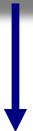
for c'est **pour** en anglais. On veux répéter une tâche *pour* un certain nombre de fois



```
for(int i = 0; i < N; i++) {  
    // répéter N fois  
}
```

For Loop

int signifier **nombre entier**
(integer en anglais).



```
for(int i = 0; i < N; i++) {  
    // répéter N fois  
}
```

nombre entier: 1, 4, 350, 1000

decimal: 1,5

Karel ne peux pas faire quelque chose un moitié.

Karel, lavez 100 pièce des habilles

```
import stanford.karel.*;

public class Habilles extends SuperKarel {
    public void run() {
        move();
        // Cette "for loop" va repeater le
        code de dans 100 fois.
        for (int i = 0; i < 100; i++) {
            putBeeper();
        }
        move();
    }
}
```

Karel, lavez 15 pièce des habilles

```
import stanford.karel.*;

public class Habilles extends SuperKarel {
    public void run() {
        move();
        // Cette "for loop" va repeater le
        code de dans 15 fois.
        for (int i = 0; i < 15; i++) {
            putBeeper();
        }
        move();
    }
}
```

Karel, lavez 200 pièce des habilles

```
import stanford.karel.*;

public class Habilles extends SuperKarel {
    public void run() {
        move();
        // Cette "for loop" va repeater le
        code de dans 200 fois.
        for (int i = 0; i < 200; i++) {
            putBeeper();
        }
        move();
    }
}
```

Karel, lavez 75 pièce des habilles

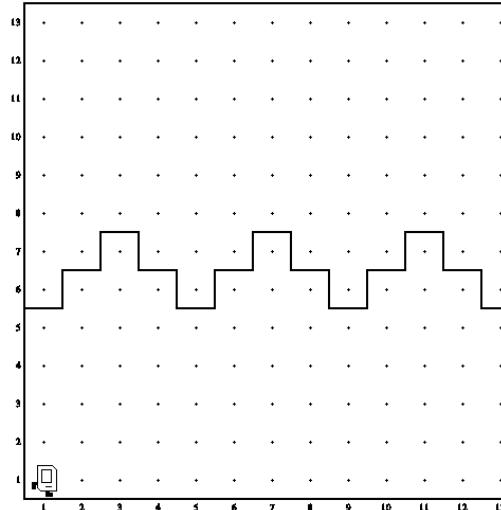
```
import stanford.karel.*;

public class Habilles extends SuperKarel {
    public void run() {
        move();
        // Cette "for loop" va repeater le
        code de dans 75 fois.
        for (int i = 0; i < 75; i++) {
            putBeeper();
        }
        move();
    }
}
```

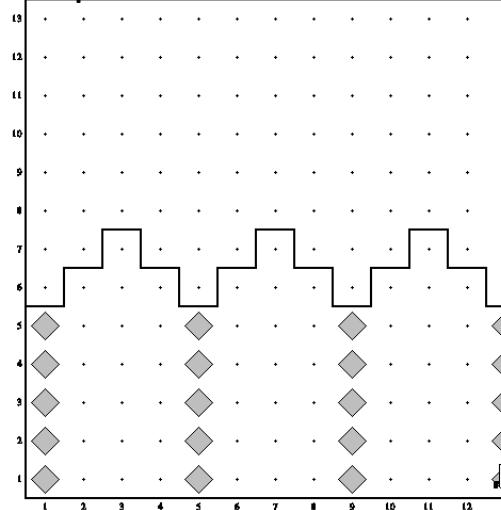
Savez-vous comment faire construire à Karel la Grande Mosquée de Conakry?



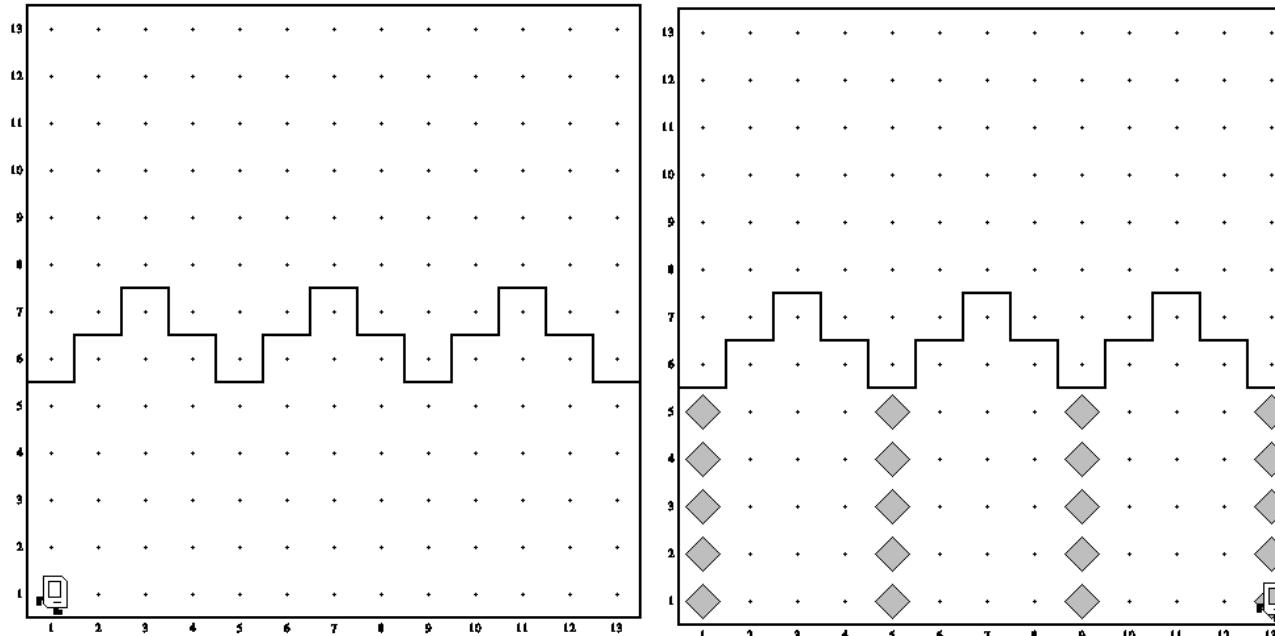
Avant



Après

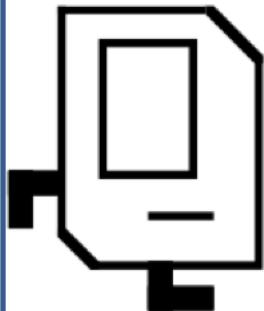


Savez-vous comment faire construire à Karel la Grande Mosquée de Conakry?



UTILISEZ TOUTES QUE KAREL CONNAIT

```
for(int i = 0; i < N; i++) {  
    // répéter N fois  
}
```



move(); bougez
turnLeft(); tournez à gauche
putBeeper(); mettre le bip
pickBeeper(); prendre le bip

```
private void nom() {  
    declarations dans le corps de  
    la méthode  
}
```

PRATIQUE! Construire la Grande Mosquée

CS Bridge

Référence ▾

Projets ▾

Des exemples ▾

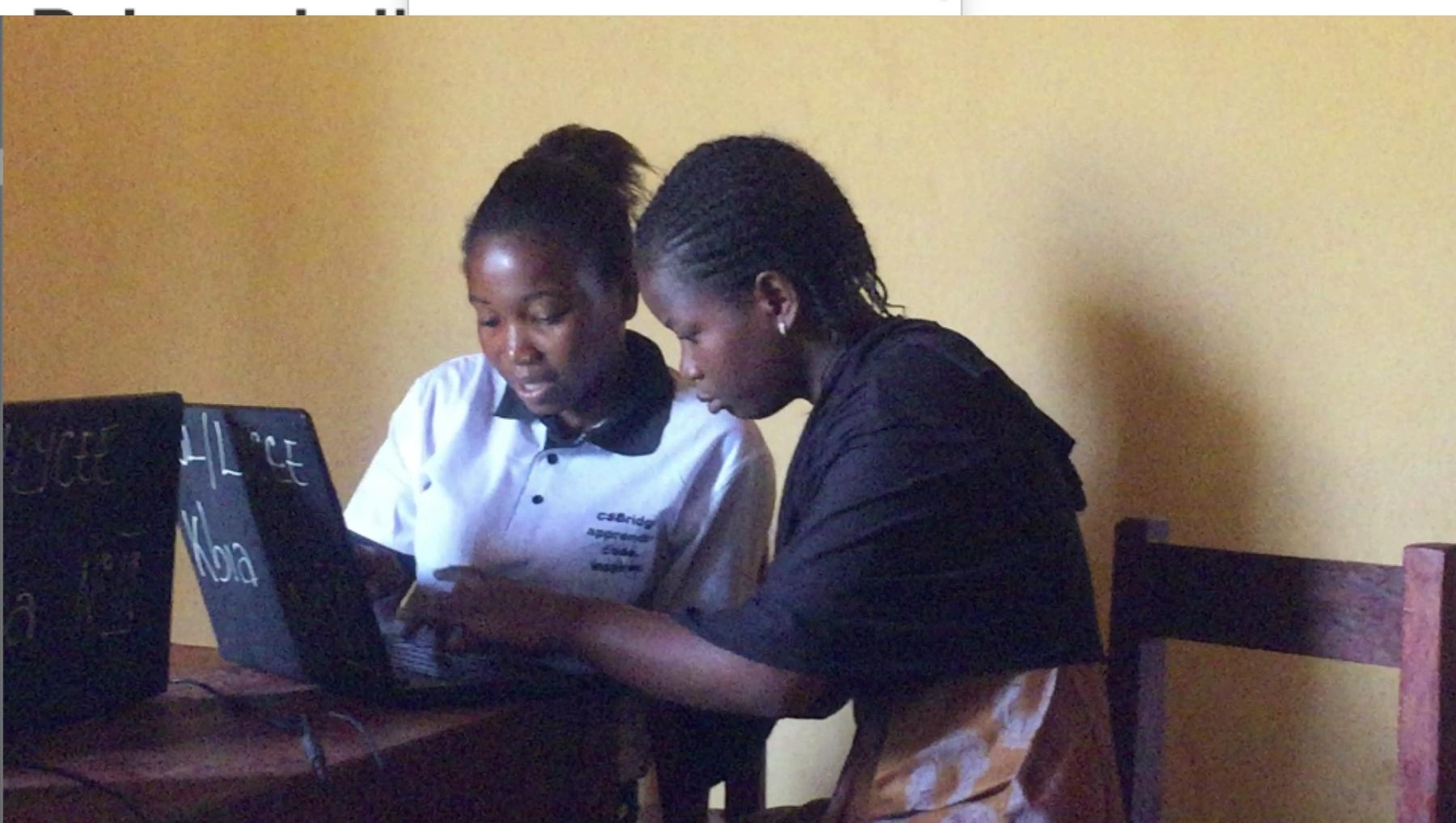
Les diapositives ▾

Formes ▾

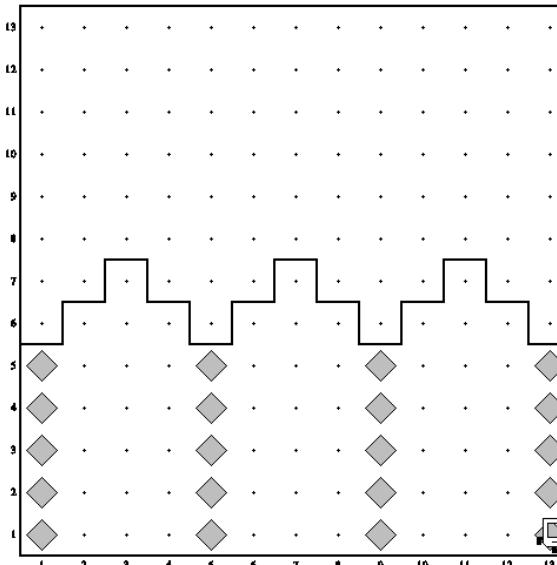
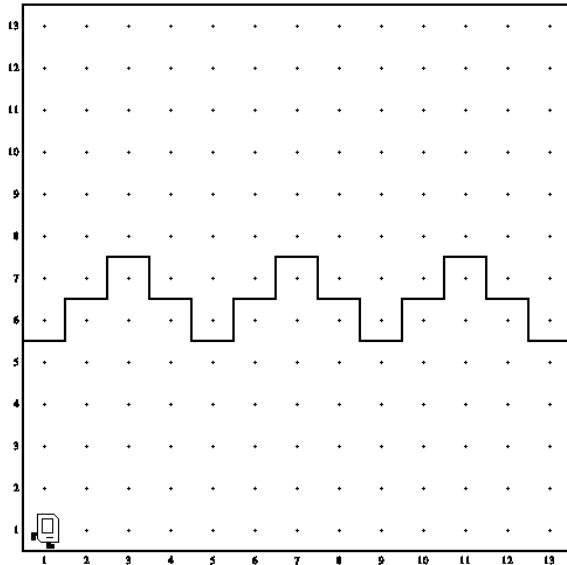


Puiser de l'Eau

Construire La Grande Mosquée de Conakry

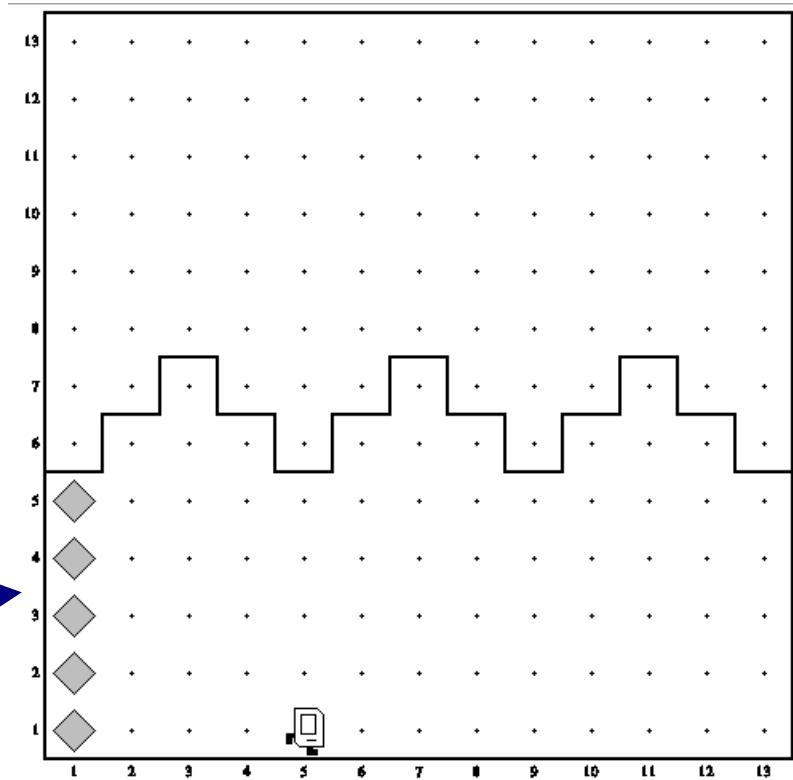


Comment avez-vous le decomposer?



Il y a beaucoup de façons à faire. Je vais vous montrer 1:

construireUnPilier(); bougezAuProchainePilier();

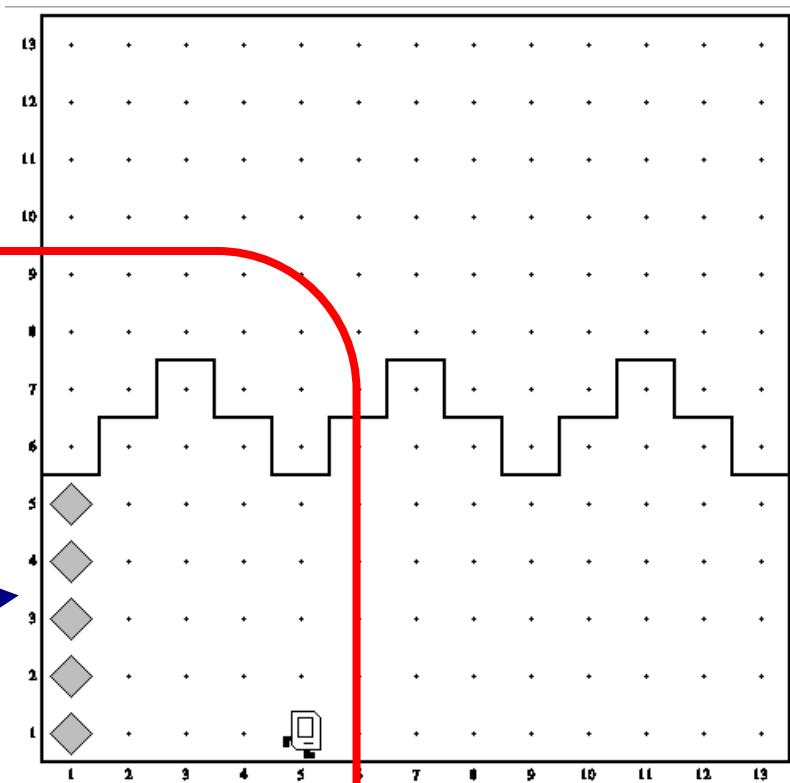


construireUnPilier();

bougezAuProchainePilier();

On veux repéter c'est deux méthodes
combien de fois???

construireUnPilier(); bougezAuProchainePilier();

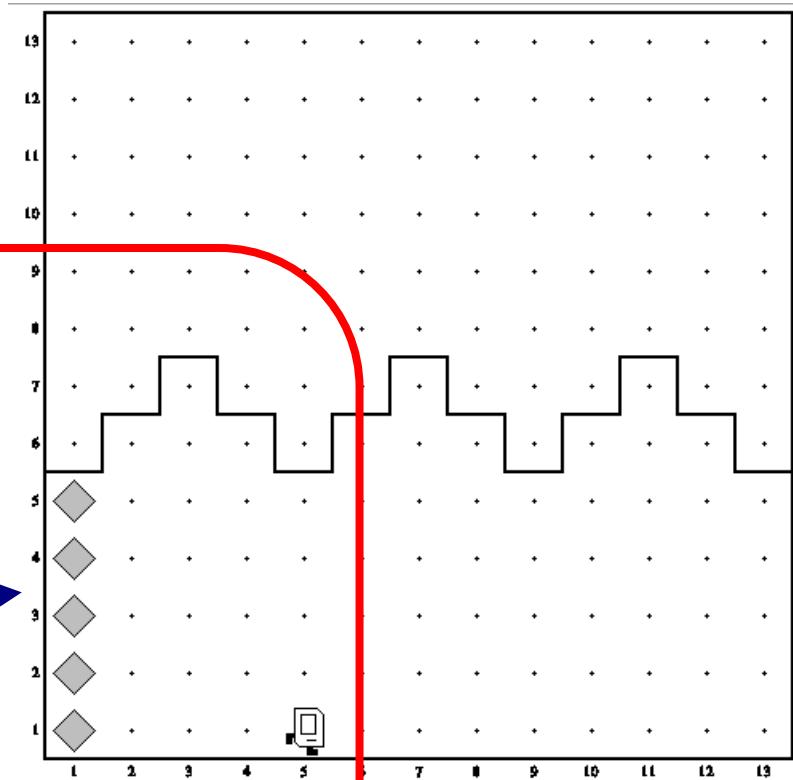


construireUnPilier();

bougezAuProchainePilier();

Utilisez un **for loop** pour
repetez 3 fois

Les première 3 piliers....



construireUnPilier();

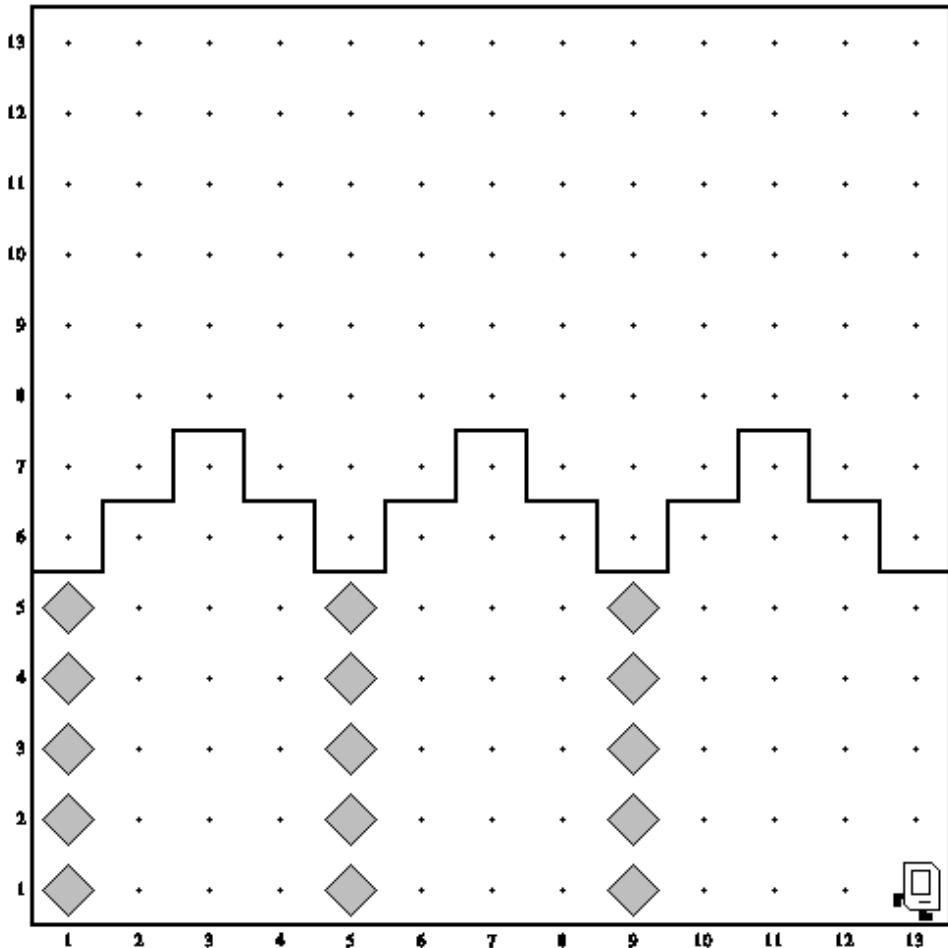
bougezAuProchainePilier();

Utilisez un **for loop** pour
repetez 3 fois

Pourquoi pas repetez 4 fois???????

Les premiere 3 piliers....

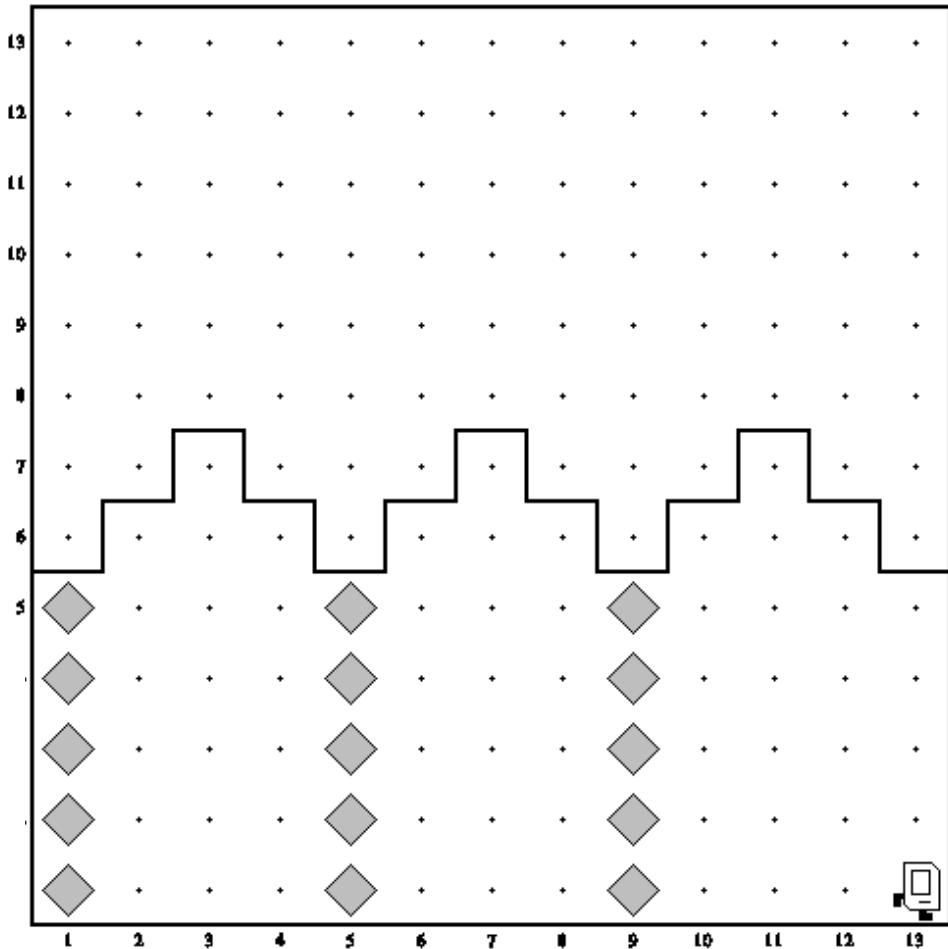
```
public void run() {  
    for (int i = 0; i < 3; i++) {  
        construireUnPilier();  
        bougezAuProchainePilier();  
    }  
}
```



Les premiere 3 piliers....

```
public void run() {  
    for (int i = 0; i < 3; i++) {  
        construireUnPilier();  
        bougezAuProchainePilier();  
    }  
}
```

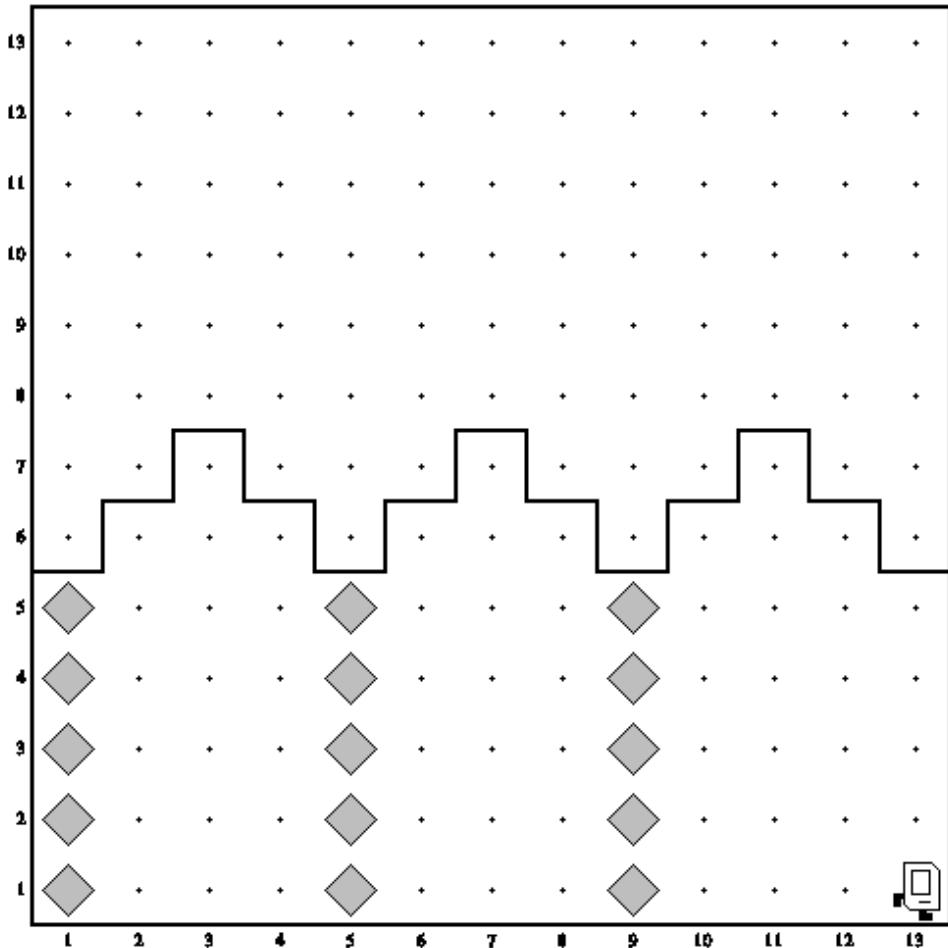
Comment pouvons-nous construire le dernier?



Les premiere 3 piliers....

```
public void run() {  
    for (int i = 0; i < 3; i++) {  
        construireUnPilier();  
        bougezAuProchainePilier();  
    }  
    construireUnPilier();  
}
```

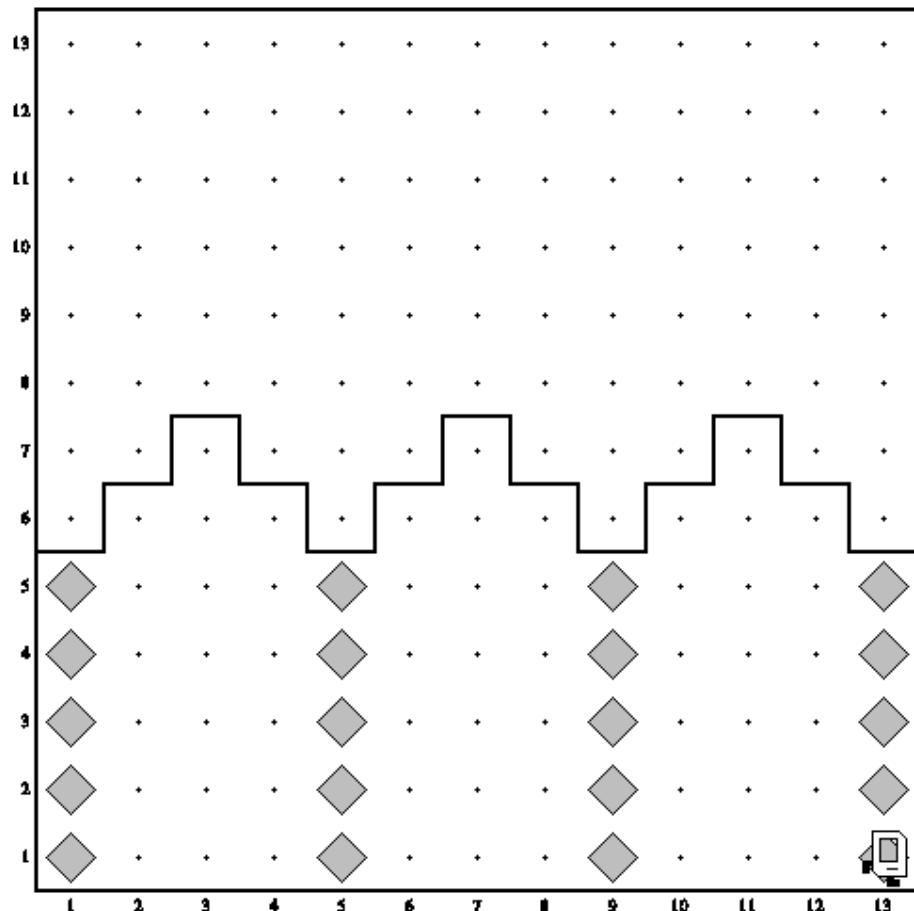
En dehors de la for loop!!!



Les premiere 3 piliers....

```
public void run() {  
    for (int i = 0; i < 3; i++) {  
        construireUnPilier();  
        bougezAuProchainePilier();  
    }  
    construireUnPilier();  
}
```

En dehors de la for loop!!!



Comment avez-vous construire la Mosquée?

```
public void run() {  
    for (int i = 0; i < 3; i++) {  
        construireUnPilier();  
        bougezAuProchainePilier();  
    }  
    construireUnPilier();  
}  
  
private void construireUnPilier() {  
    // Karel mettre les bips pour construire le pilier  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        putBeeper();  
        move();  
    }  
    putBeeper();  
  
    // Karel descend le pilier  
    turnLeft();  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
    turnLeft();  
}  
  
private void bougezAuProchainePilier() {  
    for (int i = 0; i < 4; i++) {  
        move();  
    }}}
```

Comment avez-vous construire la Mosquée?

```
public void run() {  
    for (int i = 0; i < 3; i++) {  
        construireUnPilier();  
        bougezAuProchainePilier();  
    }  
    construireUnPilier();  
}  
  
private void construireUnPilier() {  
    // Karel mettre les bips pour construire le pilier  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        putBeeper();  
        move();  
    }  
    putBeeper();  
  
    // Karel descend le pilier  
    turnLeft();  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
    turnLeft();  
}  
  
private void bougezAuProchainePilier() {  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
}
```

Les commentaires peuvent vous aider à vous rappeler ce que fait votre code et aider les autres utilisateurs qui l'utilisent à le comprendre.

Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

    // Karel descend le pilier
    turnLeft();
    turnLeft();
    for (int i = 0; i < 4; i++) {
        move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }
}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

    // Karel descend le pilier
    turnLeft();
    turnLeft();
    for (int i = 0; i < 4; i++) {
        move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }
}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

    // Karel descend le pilier
    turnLeft();
    turnLeft();
    for (int i = 0; i < 4; i++) {
        move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }
}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

    // Karel descend le pilier
    turnLeft();
    turnLeft();
    for (int i = 0; i < 4; i++) {
        move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }
}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

    // Karel descend le pilier
    turnLeft();
    turnLeft();
    for (int i = 0; i < 4; i++) {
        move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }
}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

    // Karel descend le pilier
    turnLeft();
    turnLeft();
    for (int i = 0; i < 4; i++) {
        move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }
}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {  
    for (int i = 0; i < 3; i++) {  
        construireUnPilier();  
        bougezAuProchainePilier();  
    }  
    construireUnPilier();  
}  
  
private void construireUnPilier() {  
    // Karel mettre les bips pour construire le pilier  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        putBeeper();  
        move();  
    }  
    putBeeper();  
  
    // Karel descend le pilier  
    turnLeft();  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
    turnLeft();  
}  
  
private void bougezAuProchainePilier() {  
    for (int i = 0; i < 4; i++) {  
        move();  
    }}}
```

1

Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

    // Karel descend le pilier
    turnLeft();
    turnLeft();
    for (int i = 0; i < 4; i++) {
        move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }
}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

    // Karel descend le pilier
    turnLeft();
    turnLeft();
    for (int i = 0; i < 4; i++) {
        move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }
}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {  
    for (int i = 0; i < 3; i++) {  
        construireUnPilier();  
        bougezAuProchainePilier();  
    }  
    construireUnPilier();  
}  
  
private void construireUnPilier() {  
    // Karel mettre les bips pour construire le pilier  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        putBeeper();  
        move();  
    }  
    putBeeper();  
  
    // Karel descend le pilier  
    turnLeft();  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
    turnLeft();  
}  
  
private void bougezAuProchainePilier() {  
    for (int i = 0; i < 4; i++) {  
        move();  
    }}}
```

2

Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

    // Karel descend le pilier
    turnLeft();
    turnLeft();
    for (int i = 0; i < 4; i++) {
        move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }
}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

    // Karel descend le pilier
    turnLeft();
    turnLeft();
    for (int i = 0; i < 4; i++) {
        move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }
}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {  
    for (int i = 0; i < 3; i++) {  
        construireUnPilier();  
        bougezAuProchainePilier();  
    }  
    construireUnPilier();  
}  
  
private void construireUnPilier() {  
    // Karel mettre les bips pour construire le pilier  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        putBeeper();  
        move();  
    }  
    putBeeper();  
  
    // Karel descend le pilier  
    turnLeft();  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
    turnLeft();  
}  
  
private void bougezAuProchainePilier() {  
    for (int i = 0; i < 4; i++) {  
        move();  
    }}}
```

3

Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

    // Karel descend le pilier
    turnLeft();
    turnLeft();
    for (int i = 0; i < 4; i++) {
        move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }
}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

    // Karel descend le pilier
    turnLeft();
    turnLeft();
    for (int i = 0; i < 4; i++) {
        move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }
}
```

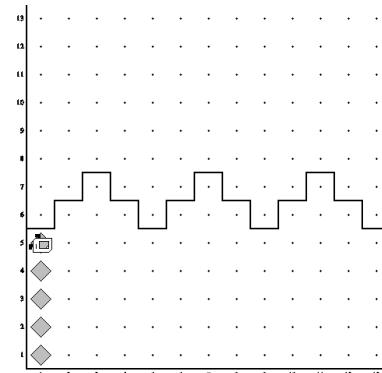
Flux de Contrôle: Construire le Mosquée

```
public void run() {  
    for (int i = 0; i < 3; i++) {  
        construireUnPilier();  
        bougezAuProchainePilier();  
    }  
    construireUnPilier();  
}  
  
private void construireUnPilier() {  
    // Karel mettre les bips pour construire le pilier  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        putBeeper();  
        move();  
    }  
    putBeeper();  
  
    // Karel descend le pilier  
    turnLeft();  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
    turnLeft();  
}  
  
private void bougezAuProchainePilier() {  
    for (int i = 0; i < 4; i++) {  
        move();  
    }}}
```

4

Flux de Contrôle: Construire le Mosquée

```
public void run() {  
    for (int i = 0; i < 3; i++) {  
        construireUnPilier();  
        bougezAuProchainePilier();  
    }  
    construireUnPilier();  
}  
  
private void construireUnPilier() {  
    // Karel mettre les bips pour construire le pilier  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        putBeeper();  
        move();  
    }  
    putBeeper();  
  
    // Karel descend le pilier  
    turnLeft();  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
    turnLeft();  
}  
  
private void bougezAuProchainePilier() {  
    for (int i = 0; i < 4; i++) {  
        move();  
    }}}
```



Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

    // Karel descend le pilier
    turnLeft();
    turnLeft();
    for (int i = 0; i < 4; i++) {
        move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }
}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

    // Karel descend le pilier
    turnLeft();
    turnLeft(); turnLeft();
    for (int i = 0; i < 4; i++) {
        move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }
}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

    // Karel descend le pilier
    turnLeft();
    turnLeft();
    for (int i = 0; i < 4; i++) {
        move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }
}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

    // Karel descend le pilier
    turnLeft();
    turnLeft();
    for (int i = 0; i < 4; i++) {
        move(); move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }
}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {  
    for (int i = 0; i < 3; i++) {  
        construireUnPilier();  
        bougezAuProchainePilier();  
    }  
    construireUnPilier();  
}  
  
private void construireUnPilier() {  
    // Karel mettre les bips pour construire le pilier  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        putBeeper();  
        move();  
    }  
    putBeeper();  
  
    // Karel descend le pilier  
    turnLeft();  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
    turnLeft();  
}  
  
private void bougezAuProchainePilier() {  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
}
```

1

Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

    // Karel descend le pilier
    turnLeft();
    turnLeft();
    for (int i = 0; i < 4; i++) {
        move(); move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }
}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {  
    for (int i = 0; i < 3; i++) {  
        construireUnPilier();  
        bougezAuProchainePilier();  
    }  
    construireUnPilier();  
}  
  
private void construireUnPilier() {  
    // Karel mettre les bips pour construire le pilier  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        putBeeper();  
        move();  
    }  
    putBeeper();  
  
    // Karel descend le pilier  
    turnLeft();  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
    turnLeft();  
}  
  
private void bougezAuProchainePilier() {  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
}
```

2

Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

    // Karel descend le pilier
    turnLeft();
    turnLeft();
    for (int i = 0; i < 4; i++) {
        move(); move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }
}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {  
    for (int i = 0; i < 3; i++) {  
        construireUnPilier();  
        bougezAuProchainePilier();  
    }  
    construireUnPilier();  
}  
  
private void construireUnPilier() {  
    // Karel mettre les bips pour construire le pilier  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        putBeeper();  
        move();  
    }  
    putBeeper();  
  
    // Karel descend le pilier  
    turnLeft();  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
    turnLeft();  
}  
  
private void bougezAuProchainePilier() {  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
}
```

3

Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

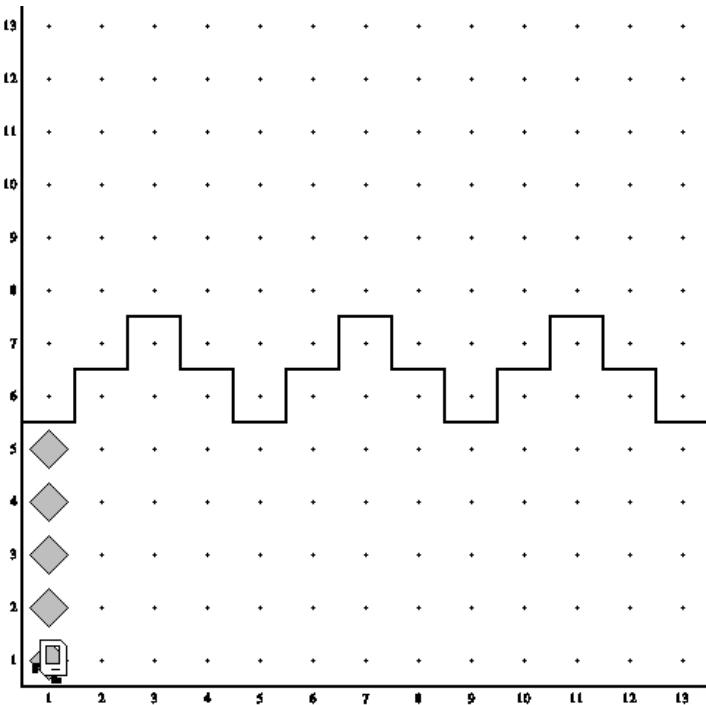
    // Karel descend le pilier
    turnLeft();
    turnLeft();
    for (int i = 0; i < 4; i++) {
        move(); move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }
}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {  
    for (int i = 0; i < 3; i++) {  
        construireUnPilier();  
        bougezAuProchainePilier();  
    }  
    construireUnPilier();  
}  
  
private void construireUnPilier() {  
    // Karel mettre les bips pour construire le pilier  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        putBeeper();  
        move();  
    }  
    putBeeper();  
  
    // Karel descend le pilier  
    turnLeft();  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
    turnLeft();  
}  
  
private void bougezAuProchainePilier() {  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
}
```

4



Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

    // Karel descend le pilier
    turnLeft();
    turnLeft();
    for (int i = 0; i < 4; i++) {
        move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }
}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

    // Karel descend le pilier
    turnLeft();
    turnLeft();
    for (int i = 0; i < 4; i++) {
        move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }
}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {  
    for (int i = 0; i < 3; i++) {  
        construireUnPilier();  
        bougezAuProchainePilier();  
    }  
    construireUnPilier();  
}  
  
private void construireUnPilier() {  
    // Karel mettre les bips pour construire le pilier  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        putBeeper();  
        move();  
    }  
    putBeeper();  
  
    // Karel descend le pilier  
    turnLeft();  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
    turnLeft();  
}  
  
private void bougezAuProchainePilier() {  
    for (int i = 0; i < 4; i++) {  
        move();  
    }}}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

    // Karel descend le pilier
    turnLeft();
    turnLeft();
    for (int i = 0; i < 4; i++) {
        move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }}}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

    // Karel descend le pilier
    turnLeft();
    turnLeft();
    for (int i = 0; i < 4; i++) {
        move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }}}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {  
    for (int i = 0; i < 3; i++) {  
        construireUnPilier();  
        bougezAuProchainePilier();  
    }  
    construireUnPilier();  
}  
  
private void construireUnPilier() {  
    // Karel mettre les bips pour construire le pilier  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        putBeeper();  
        move();  
    }  
    putBeeper();  
  
    // Karel descend le pilier  
    turnLeft();  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
    turnLeft();  
}  
  
private void bougezAuProchainePilier() {  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
}}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

    // Karel descend le pilier
    turnLeft();
    turnLeft();
    for (int i = 0; i < 4; i++) {
        move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }}}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {  
    for (int i = 0; i < 3; i++) {  
        construireUnPilier();  
        bougezAuProchainePilier();  
    }  
    construireUnPilier();  
}  
  
private void construireUnPilier() {  
    // Karel mettre les bips pour construire le pilier  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        putBeeper();  
        move();  
    }  
    putBeeper();  
  
    // Karel descend le pilier  
    turnLeft();  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
    turnLeft();  
}  
  
private void bougezAuProchainePilier() {  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

    // Karel descend le pilier
    turnLeft();
    turnLeft();
    for (int i = 0; i < 4; i++) {
        move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }}}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {  
    for (int i = 0; i < 3; i++) {  
        construireUnPilier();  
        bougezAuProchainePilier();  
    }  
    construireUnPilier();  
}  
  
private void construireUnPilier() {  
    // Karel mettre les bips pour construire le pilier  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        putBeeper();  
        move();  
    }  
    putBeeper();  
  
    // Karel descend le pilier  
    turnLeft();  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
    turnLeft();  
}  
  
private void bougezAuProchainePilier() {  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {
    for (int i = 0; i < 3; i++) {
        construireUnPilier();
        bougezAuProchainePilier();
    }
    construireUnPilier();
}

private void construireUnPilier() {
    // Karel mettre les bips pour construire le pilier
    turnLeft();
    for (int i = 0; i < 4; i++) {
        putBeeper();
        move();
    }
    putBeeper();

    // Karel descend le pilier
    turnLeft();
    turnLeft();
    for (int i = 0; i < 4; i++) {
        move();
    }
    turnLeft();
}

private void bougezAuProchainePilier() {
    for (int i = 0; i < 4; i++) {
        move();
    }}}
```

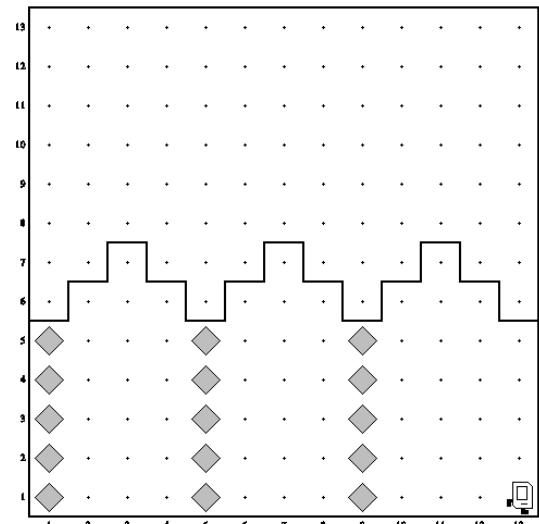
Flux de Contrôle: Construire le Mosquée

```
public void run() {  
    for (int i = 0; i < 3; i++) {  
        construireUnPilier();  
        bougezAuProchainePilier();  
    }  
    construireUnPilier();  
}  
  
private void construireUnPilier() {  
    // Karel mettre les bips pour construire le pilier  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        putBeeper();  
        move();  
    }  
    putBeeper();  
  
    // Karel descend le pilier  
    turnLeft();  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
    turnLeft();  
}  
  
private void bougezAuProchainePilier() {  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
}
```

Flux de Contrôle: Construire le Mosquée

```
public void run() {  
    for (int i = 0; i < 3; i++) {  
        construireUnPilier();  
        bougezAuProchainePilier();  
    }  
    construireUnPilier();  
}  
  
private void construireUnPilier() {  
    // Karel mettre les bips pour construire le pilier  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        putBeeper();  
        move();  
    }  
    putBeeper();  
  
    // Karel descend le pilier  
    turnLeft();  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
    turnLeft();  
}  
  
private void bougezAuProchainePilier() {  
    for (int i = 0; i < 4; i++) {  
        move();  
    }}}
```

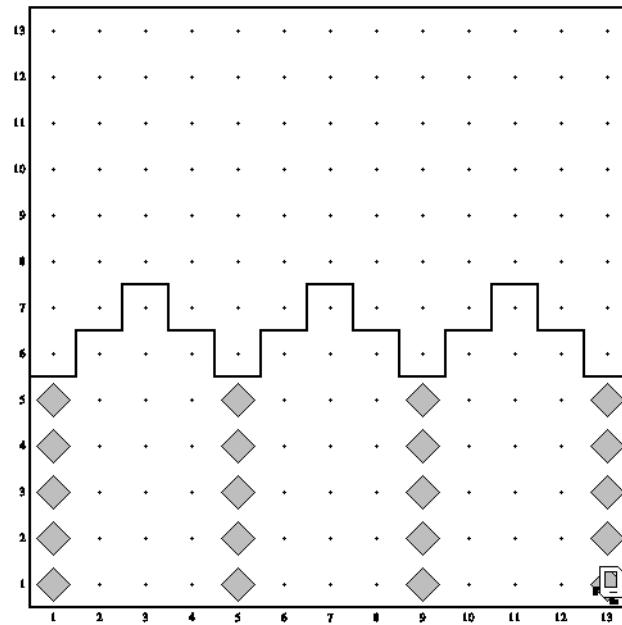
Elle va répéter TOUT cela 3 fois



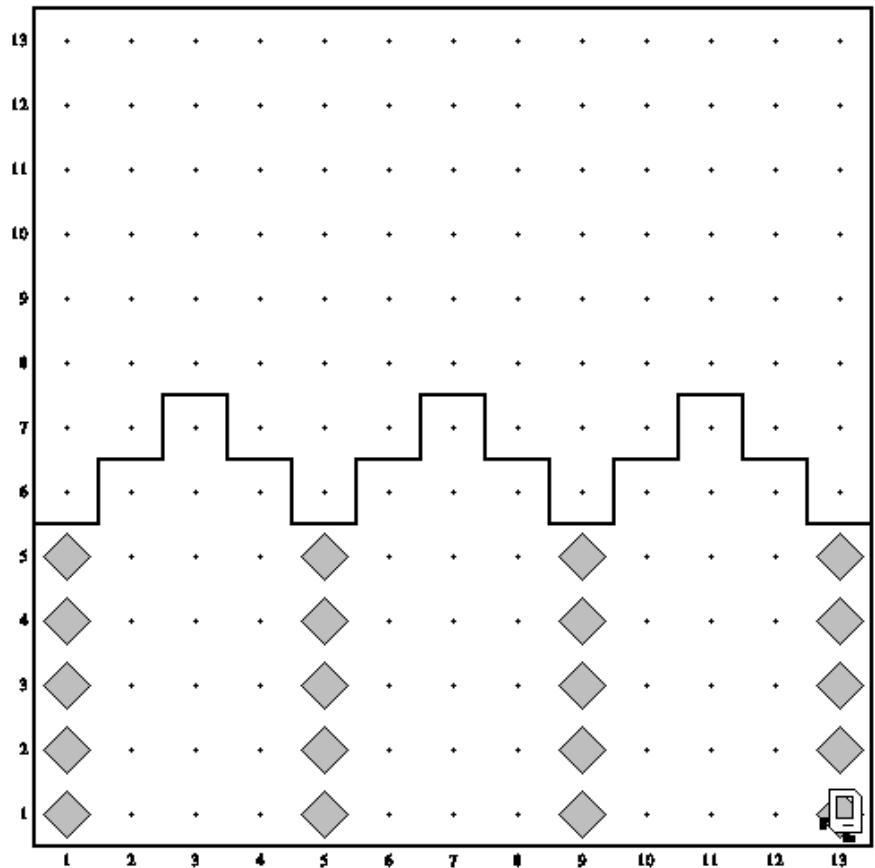
Flux de Contrôle: Construire le Mosquée

```
public void run() {  
    for (int i = 0; i < 3; i++) {  
        construireUnPilier();  
        bougezAuProchainePilier();  
    }  
    construireUnPilier();  
}  
  
private void construireUnPilier() {  
    // Karel mettre les bips pour construire le pilier  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        putBeeper();  
        move();  
    }  
    putBeeper();  
  
    // Karel descend le pilier  
    turnLeft();  
    turnLeft();  
    for (int i = 0; i < 4; i++) {  
        move();  
    }  
    turnLeft();  
}  
  
private void bougezAuProchainePilier() {  
    for (int i = 0; i < 4; i++) {  
        move();  
    }}}
```

Finally, elle doit terminer la dernière pilier!



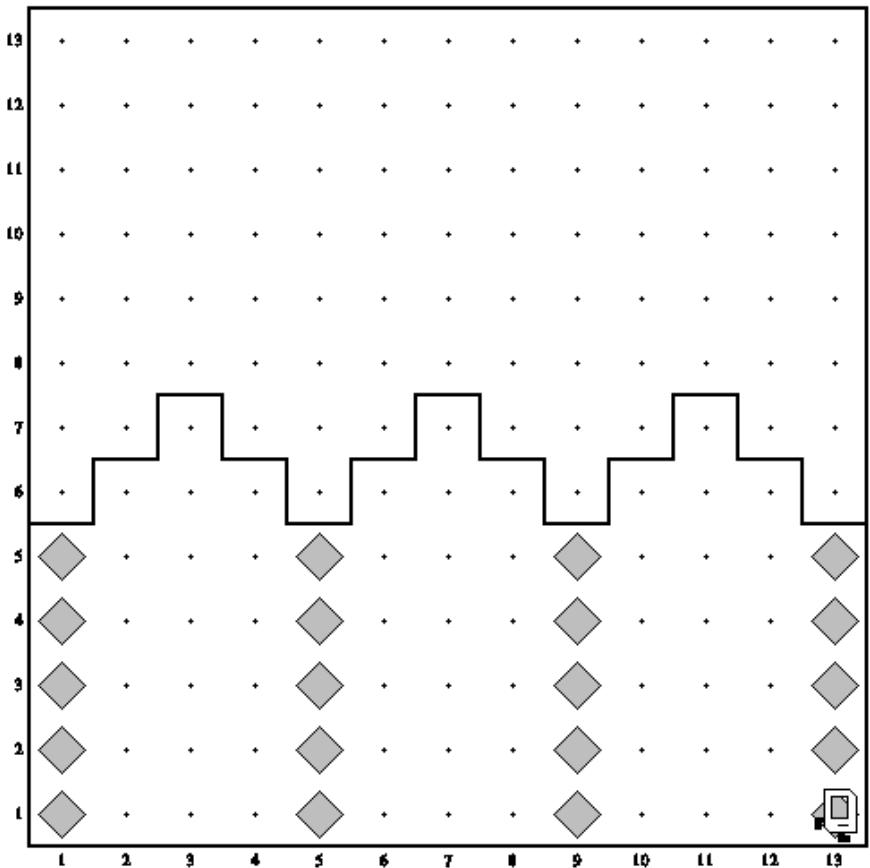
QUESTIONS?



Qui l'a fait d'une manière différente? Povez-vous l'expliquer?



FELICITATIONS!!!

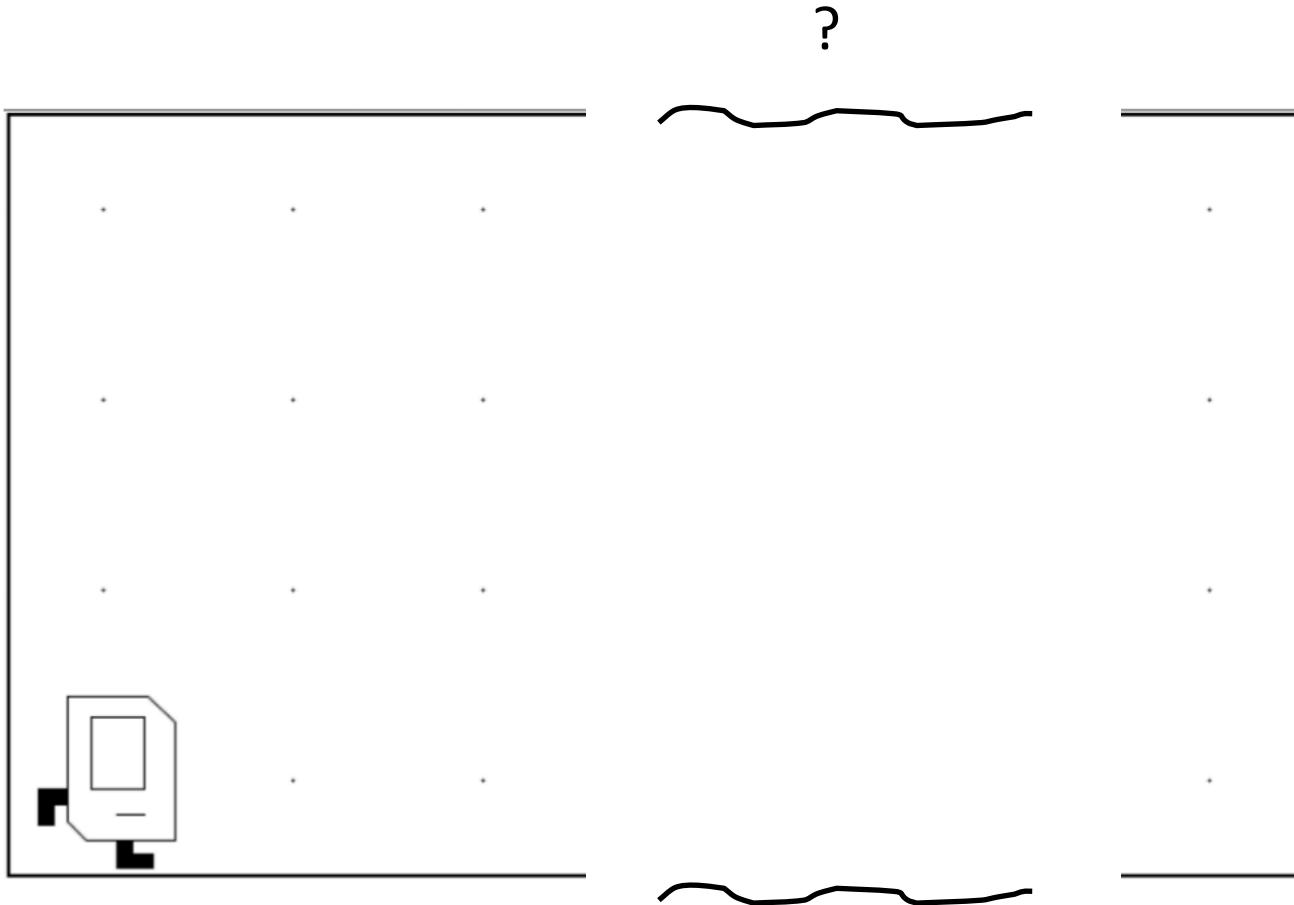


Flux de Contrôle

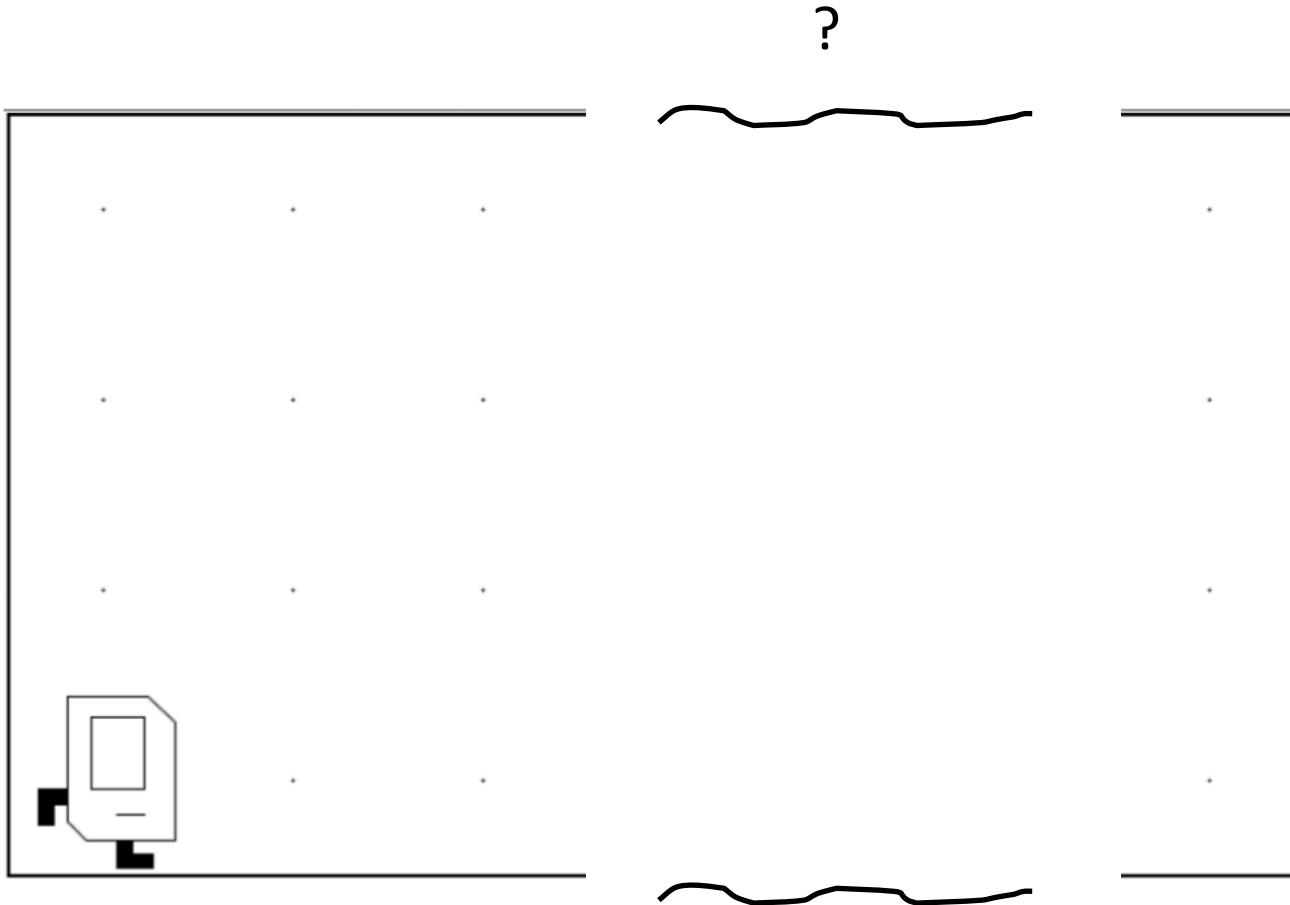
Jusqu'à présent, nous avons des programmes écrits qui se comportent de la même manière à chaque fois.

- Les programmes plus complexes se comportent différemment selon les mondes ou les cas.
- Nous allons apprendre à écrire des programmes Karel qui peuvent faire des choses basées sur ce que le monde est comme!

Programmes Généralisés

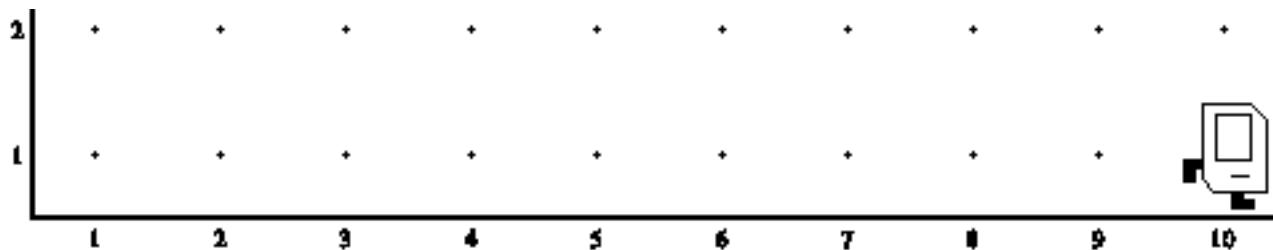
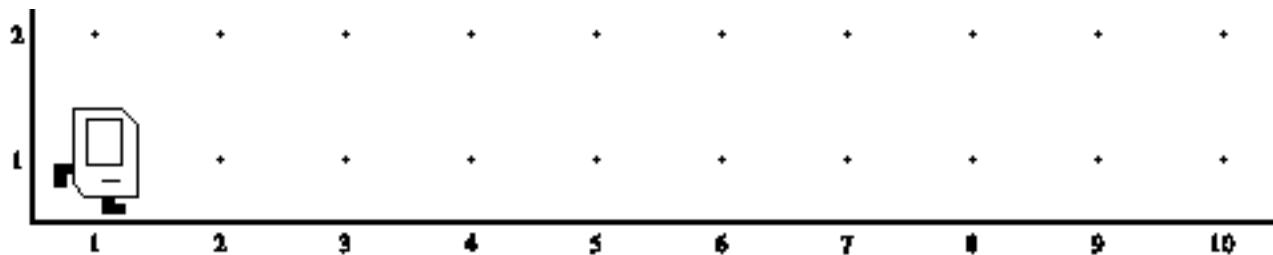


Flux de Contrôle: While Loops



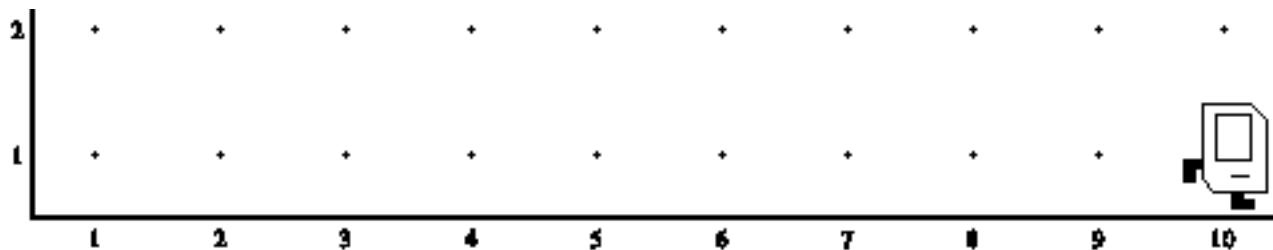
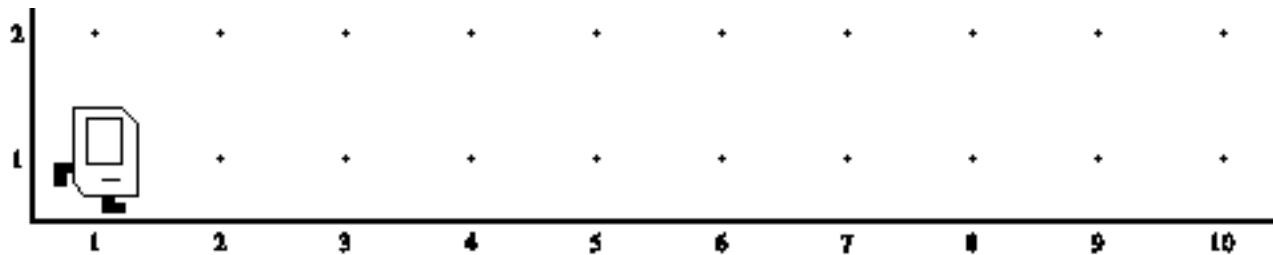
Flux de Contrôle: While Loops

Je veux que Karel bouge jusqu'à atteindre un mur. Comment je fais ça?



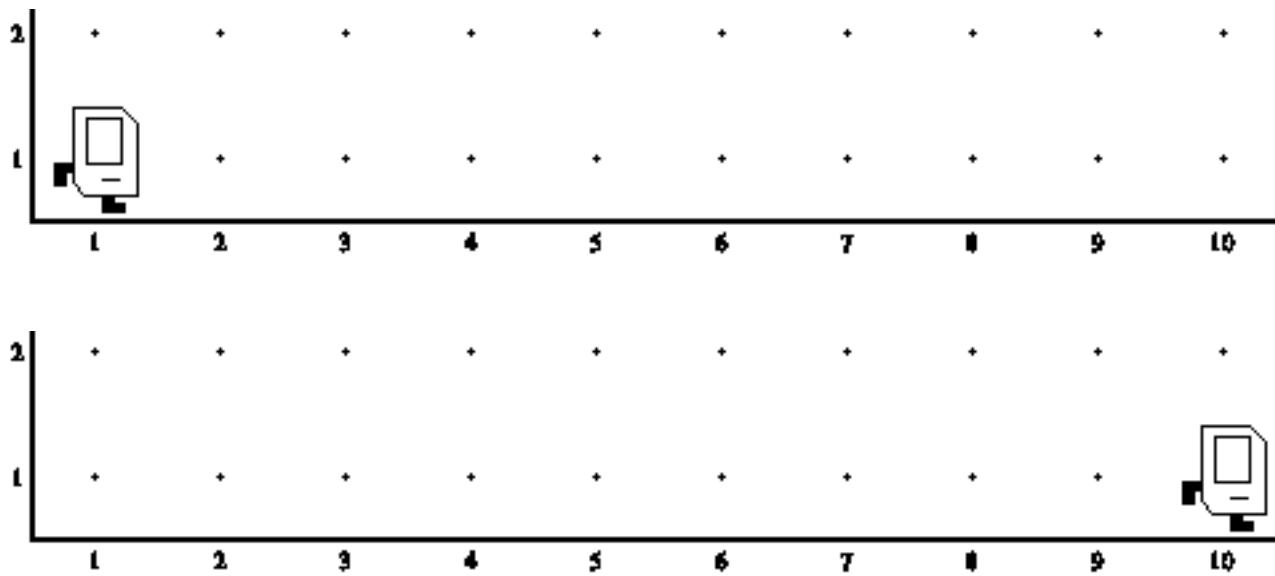
Flux de Contrôle: While Loops

Je ne veux pas écrire `move()` plus que 1 fois!



Flux de Contrôle: While Loops

Aussi, on ne connaît pas la taille du monde!
De plus, nous voulons que notre programme
soit aussi généralisé que possible et fonctionne
dans beaucoup de mondes différents.



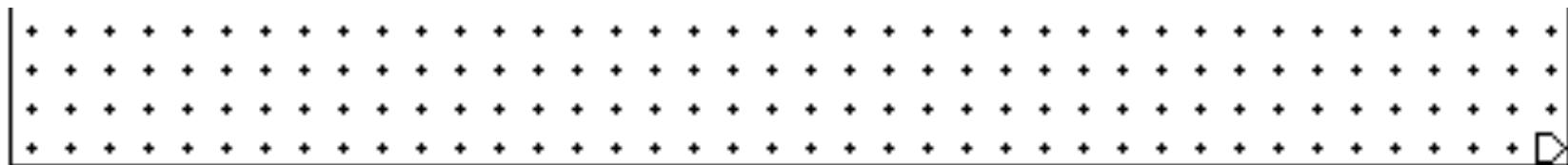
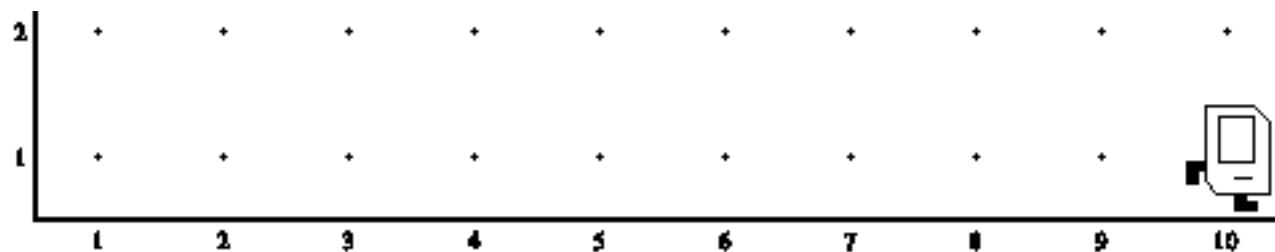
Flux de Contrôle: While Loops

Par exemple...

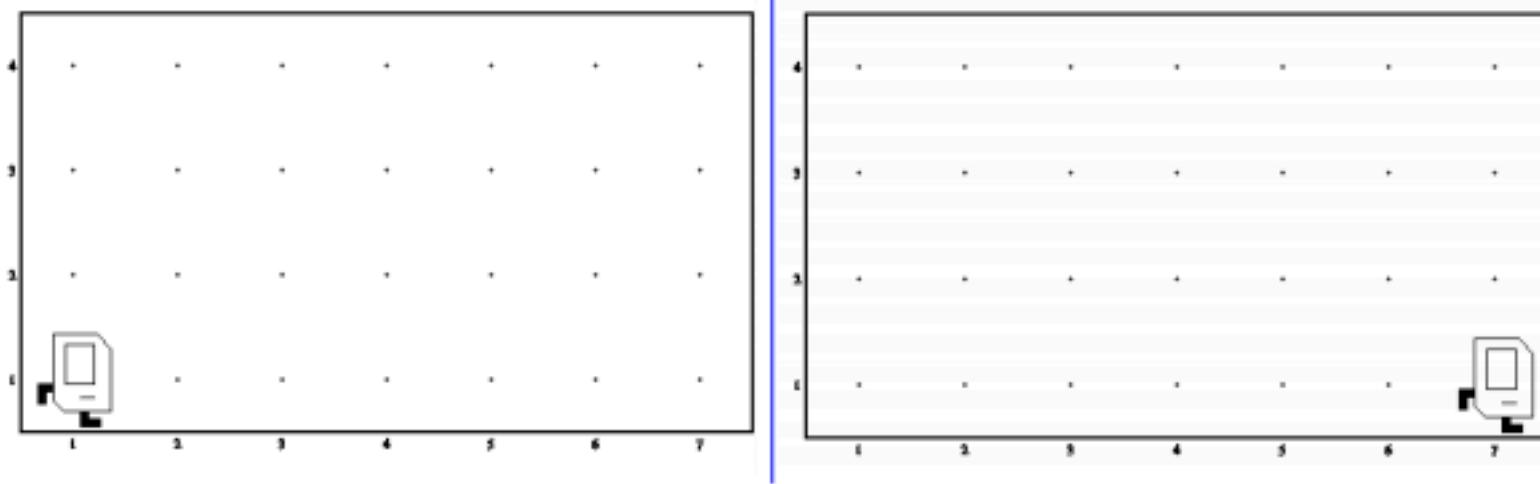
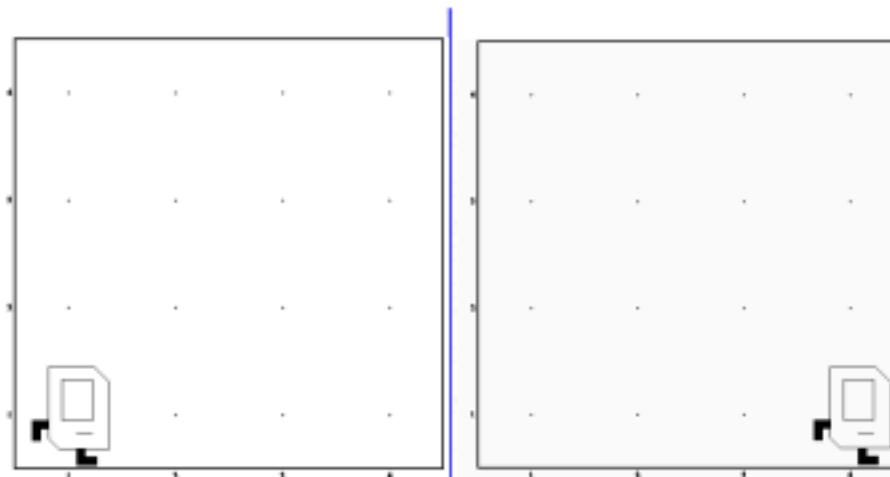


Flux de Contrôle: While Loops

Avec ce nouvel outil, nous pouvons utiliser le même code pour les deux mondes!



Travailler dans n'importe quel monde...



While Loop

(Tandis que)

Flux de Contrôle: While Loops

On utilise un **while loop**:

```
while (condition) {  
    declaration;  
    declaration;  
  
    ...  
}
```

Répète les instructions dans le corps jusqu'à ce que la condition ne soit plus vraie. Karel vérifie chaque fois la condition, puis exécute toutes les instructions. avant de vérifier à nouveau l'état.

Conditions Possible

| Tester | Contraire | Ce qu'il vérifie |
|------------------|--------------------|---------------------------------------|
| frontIsClear() | frontIsBlocked() | Y a-t-il un mur devant Karel? |
| leftIsClear() | leftIsBlocked() | Y a-t-il un mur à gauche de Karel? |
| rightIsClear() | rightIsBlocked() | Y a-t-il un mur à la droite de Karel? |
| beepersPresent() | noBeepersPresent() | Y a-t-il des beepers dans ce coin? |
| beepersInBag() | noBeepersInBag() | Karel fait-il face au nord? |
| facingNorth() | notFacingNorth() | Karel fait-il face au est? |
| facingEast() | notFacingEast() | Karel fait-il face au nord? |
| facingSouth() | notFacingSouth() | Karel fait-il face au sud? |
| facingWest() | notFacingWest() | Karel fait-il face au l'ouest? |



Karel Référence
Console Référence
Générateur aléatoire Référence
Graphique Référence
Événements (Events) Référence
Array Lists

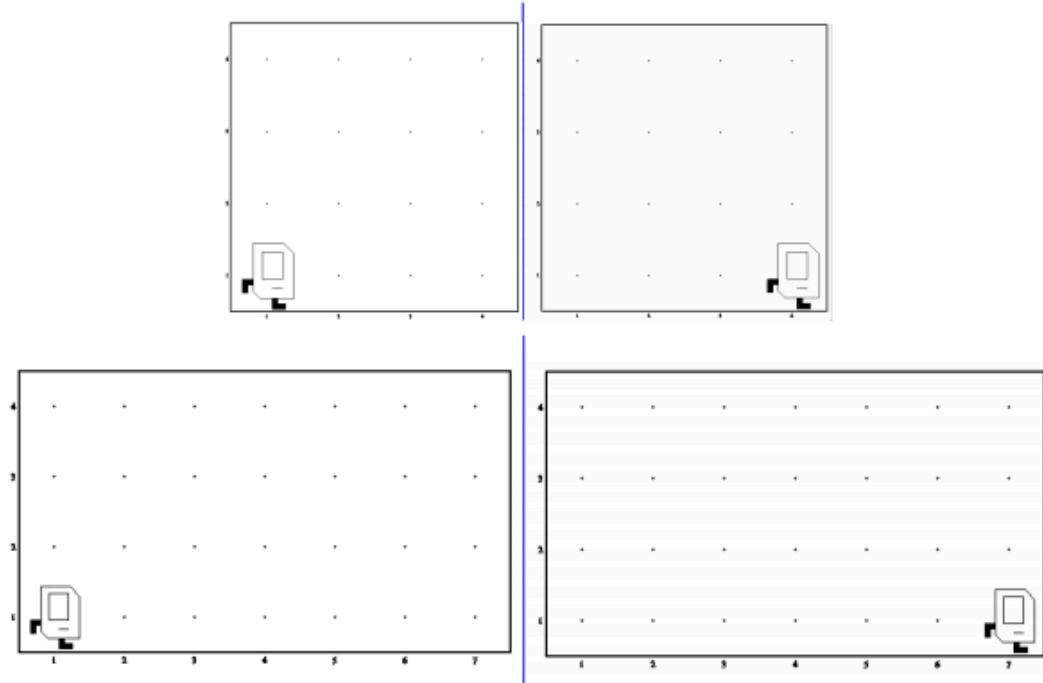
Section à l'informatique

C'est sur le site Web du cours, sous «Référence».

Flux de Contrôle: While Loops

Maintenant on peux dire:

```
while (frontIsClear()) {  
    move();  
}
```



C'est moins répétitif, et cela fonctionne dans n'importe quel taille de monde!

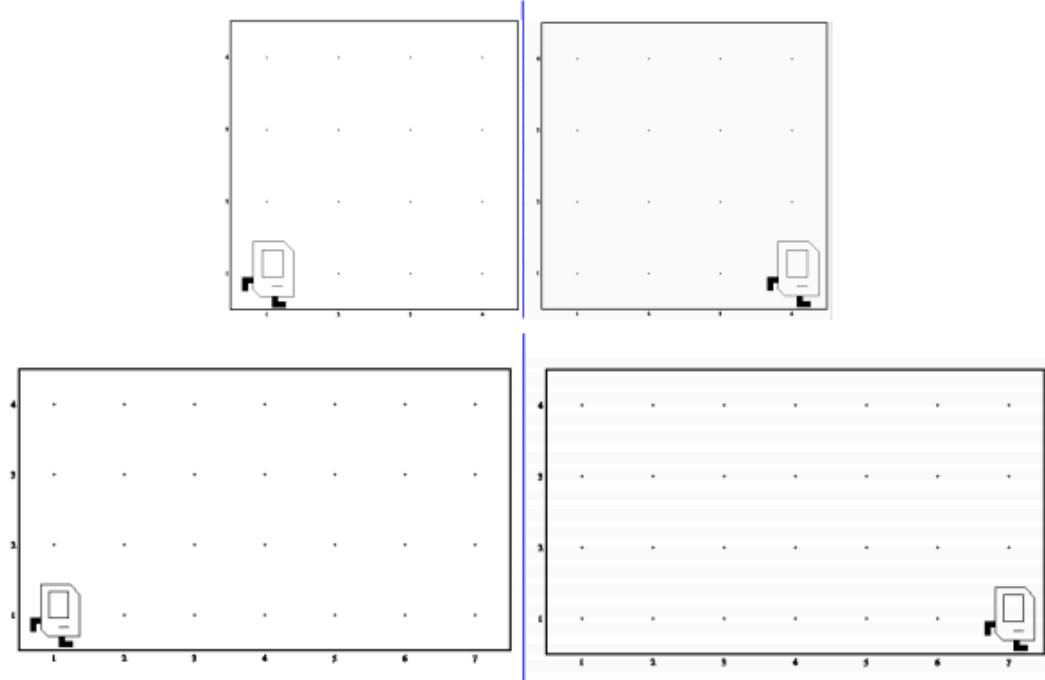
J'ai besoin deux volontaires: Karel et le commandant



Demonstration: While Loops

Maintenant on peut dire:

```
while (frontIsClear()) {  
    move();  
}
```



C'est moins répétitif, et cela fonctionne dans n'importe quel taille de monde!

Revision: Loops

Répéter le processus

Savoir combiner de fois

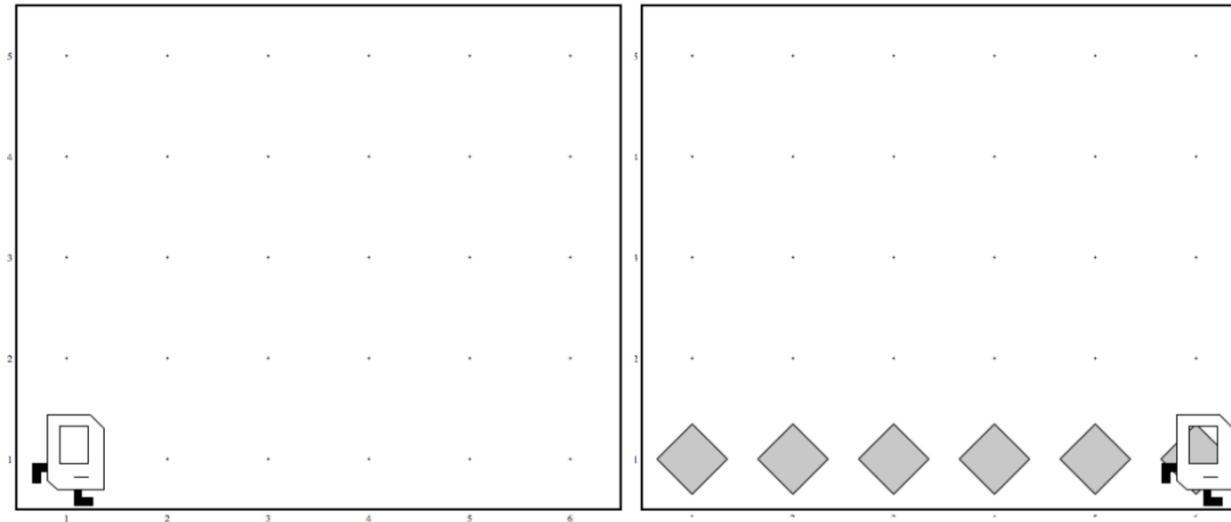
Vous ne savez pas combien de fois

For Loop

While Loop

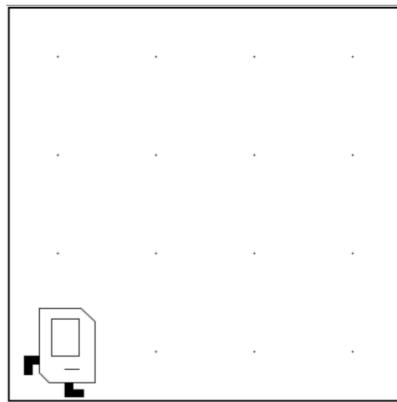
Exemple: While Loops

Je veux que Karel mette une ligne de bips jusqu'à ce le mur. Comment puis-je faire cela?

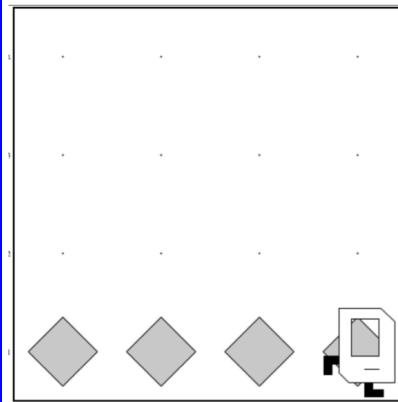


Travailler dans n'importe quelle monde!

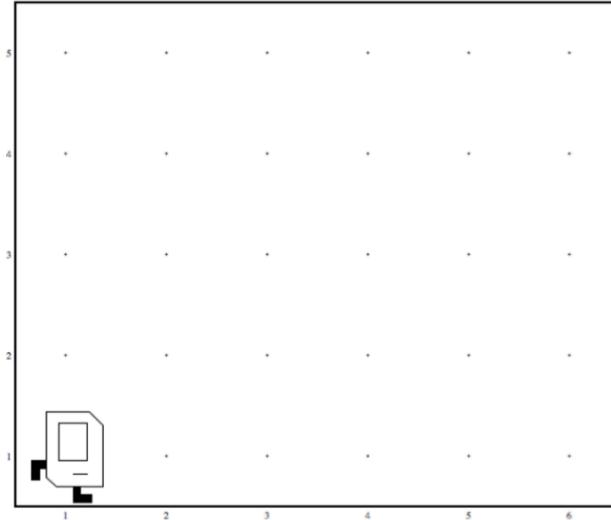
Avant



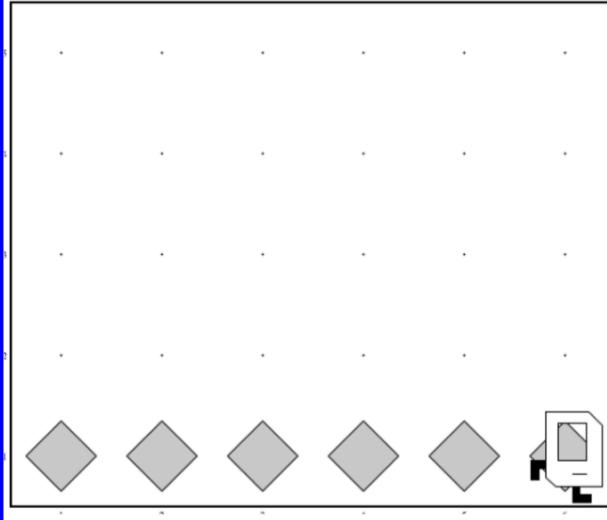
Après



Avant



Après



While Loop

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {
        // exemple while loop
        while(condition) {
            code à répéter
        }
    }
}
```

Verifier la condition

Vrai -> fais le

Verifier la condition

Vrai -> fais le

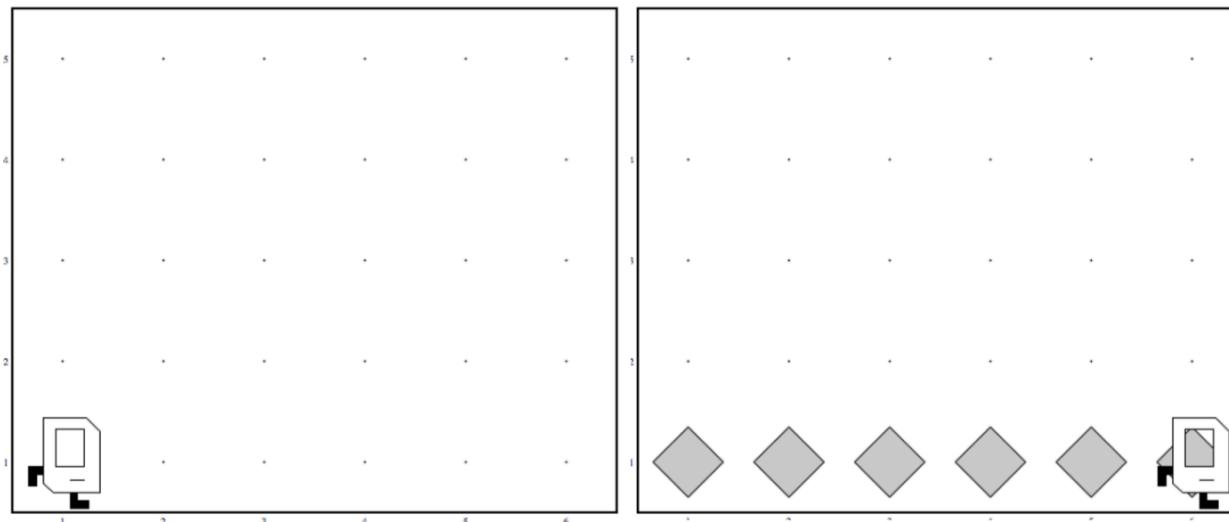
....

Verifier la condition

Faux-> ne le fais pas

Continuer le reste du programme

While Loop



On va utiliser la condition `frontIsClear()`
Ca va vérifier si il y a un mur devant Karel...

Placez les Bips sur la Ligne

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // exemple while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra mettre la bip
        putBeeper();
    }
}
```



Flux de Contrôle

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop... avant est clair
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra mettre le bip
        putBeeper();
    }
}
```



Place Beeper Line

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra mettre le bip
        putBeeper();
    }
}
```



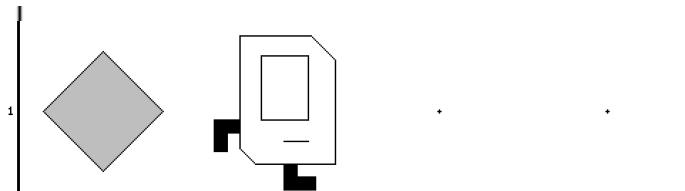
Place Beeper Line

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra mettre le bip
        putBeeper();
    }
}
```



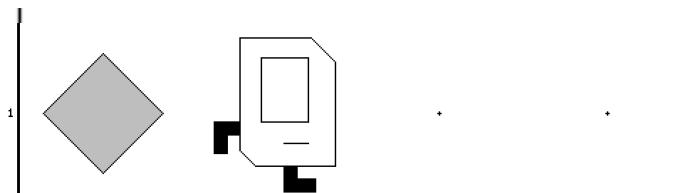
Place Beeper Line

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra mettre le bip
        putBeeper();
    }
}
```



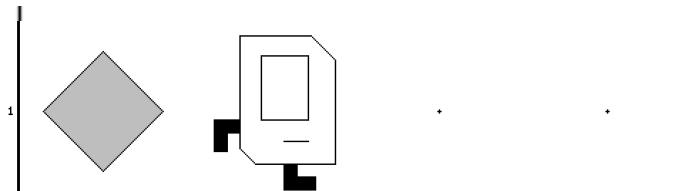
Place Beeper Line

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra mettre le bip
        putBeeper();
    }
}
```



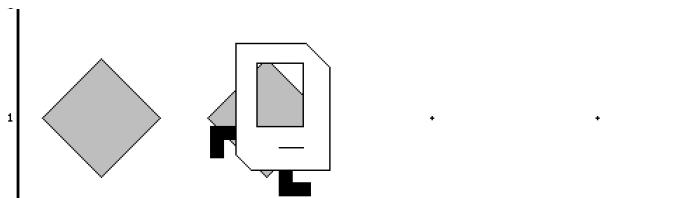
Place Beeper Line

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra mettre le bip
        putBeeper();
    }
}
```



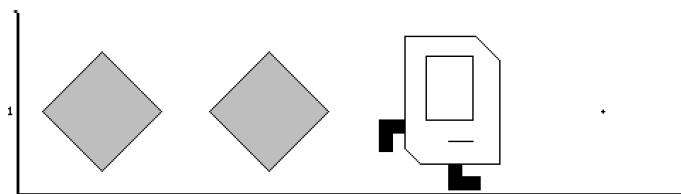
Place Beeper Line

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra mettre le bip
        putBeeper();
    }
}
```



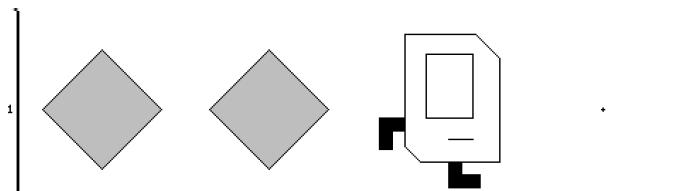
Place Beeper Line

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra mettre le bip
        putBeeper();
    }
}
```



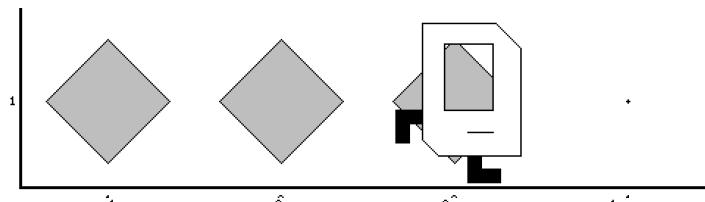
Place Beeper Line

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra put beeper
        putBeeper();
    }
}
```



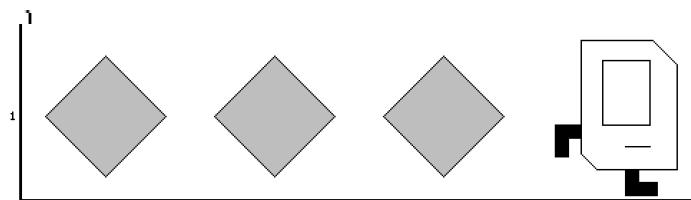
Place Beeper Line

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra put beeper
        putBeeper();
    }
}
```



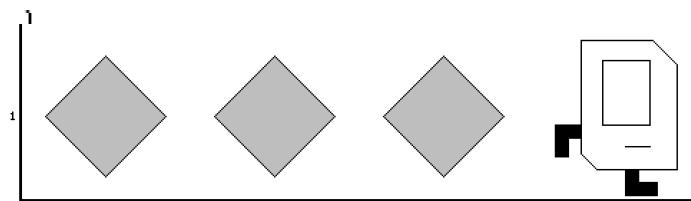
Place Beeper Line

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra put beeper
        putBeeper();
    }
}
```



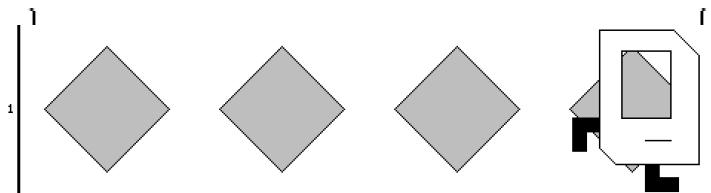
Place Beeper Line

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

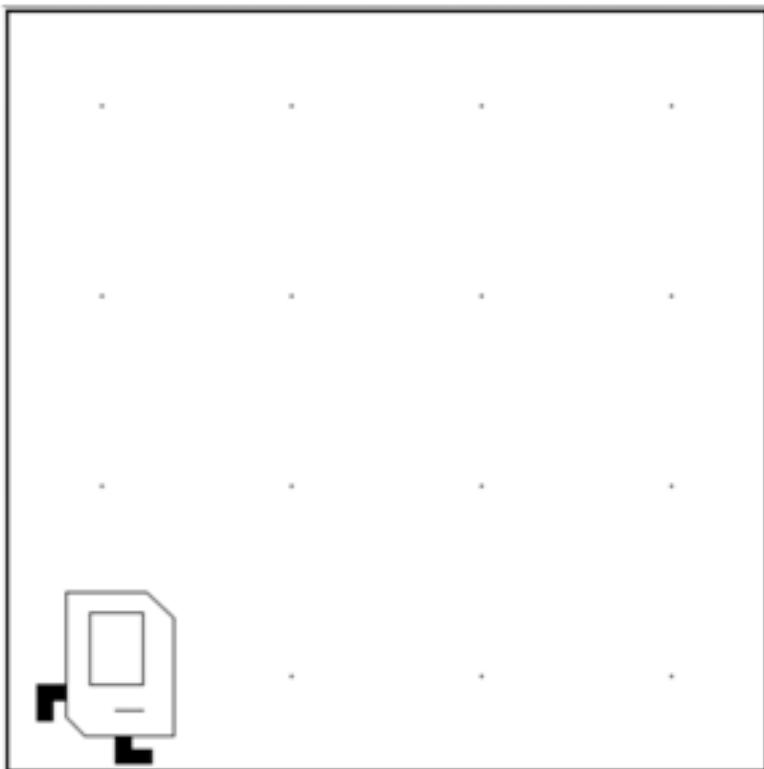
    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra put beeper
        putBeeper();
    }
}
```

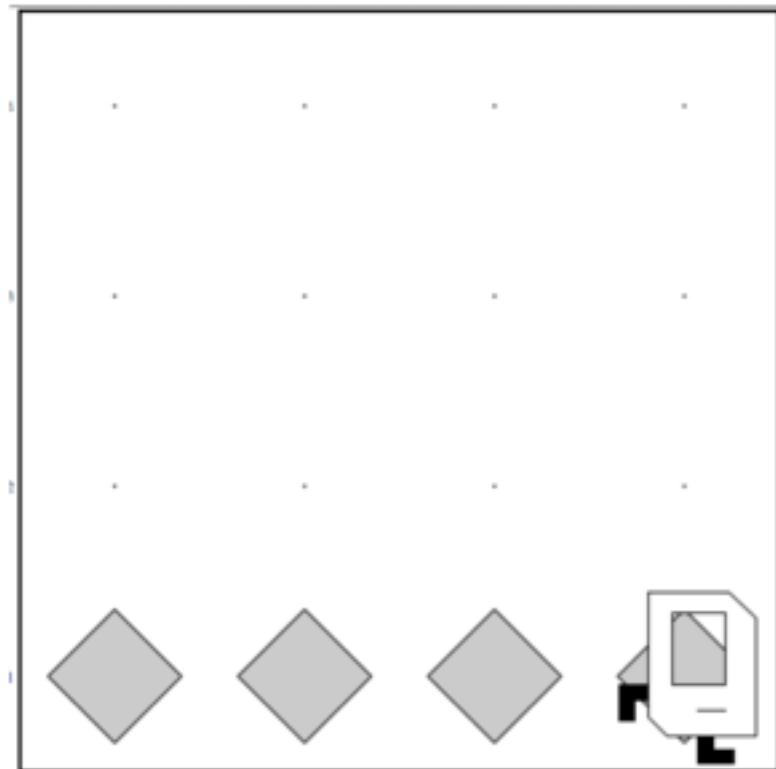


Ligne des bips

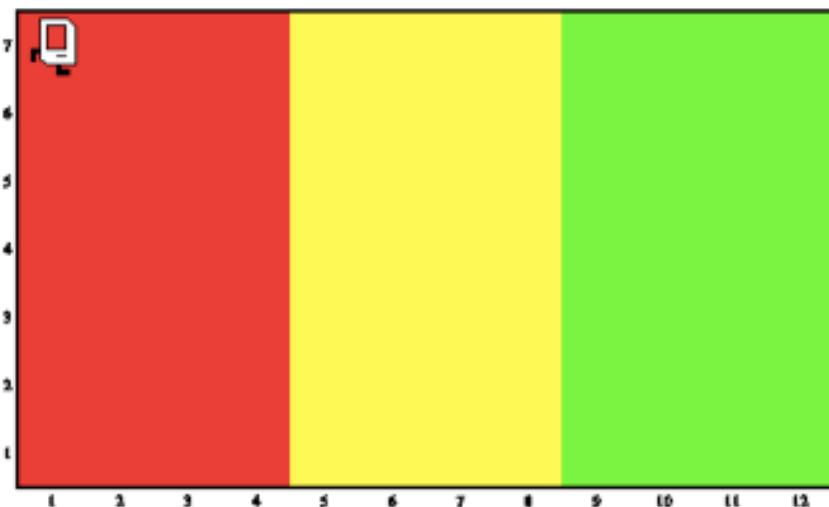
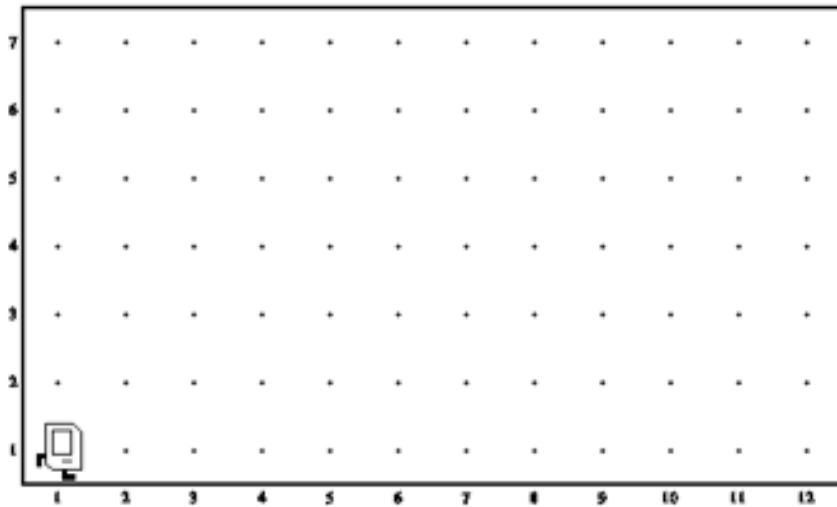
Avant



Après



Pratique: Drapeau Guinean



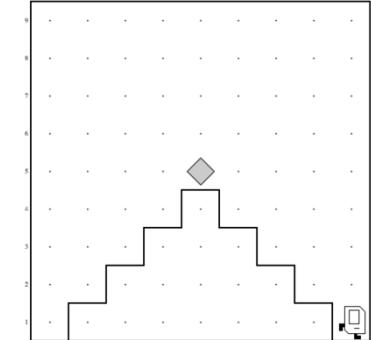
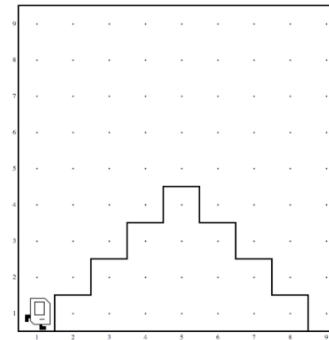
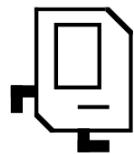
Peinture

Karel a quelques commandes dans sa manche. Elle peut peindre le coin sur lequel elle se tient en suivant les instructions

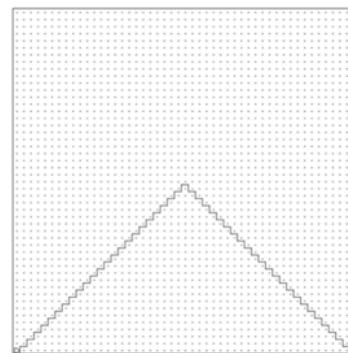
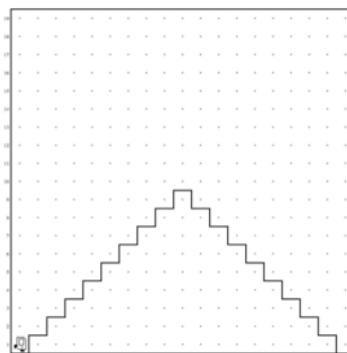
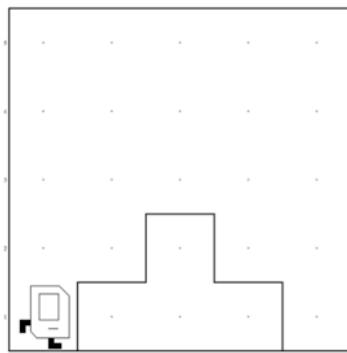
```
paintCorner(color);
```

La valeur entre parenthèses, appelée un argument dans la terminologie des langages de programmation, peut être l'un des suivants: **RED ORANGE GREEN CYAN BLUE MAGENTA** et plus...

Pratique: Karel de Montagne et de Manguïer



Travailler dans
n'importe
quelle monde....



(Conseil: Utiliser des while loops!)

Que se passe-t-il si vous ne souhaitez effectuer qu'une seule loop en fonction d'une condition?

If/Else Déclaration

If Déclaration

```
import stanford.karel.*;  
  
public class IfExample extends SuperKarel {  
  
    public void run() {  
  
        // exemple d'un if déclaration  
        if(condition) {  
            code à executer si la condition est  
            vraie  
        }  
  
    }  
  
}
```

If Déclaration

```
import stanford.karel.*;  
  
public class IfExample extends CSBridgeStudent{  
  
    public void run() {  
  
        // exemple d'un if déclaration  
        if(tuEsUnElève()) {  
            levezLaMain();  
        }  
    }  
}
```

Assumez-vous comme Karel et exécutez ce programme!

If Déclaration

```
import stanford.karel.*;  
  
public class IfExample extends SuperKarel{  
  
    public void run() {  
        safeMove();  
    }  
  
    private void safeMove() {  
        if(frontIsClear()) {  
            move();  
        }  
    }  
}  
}
```

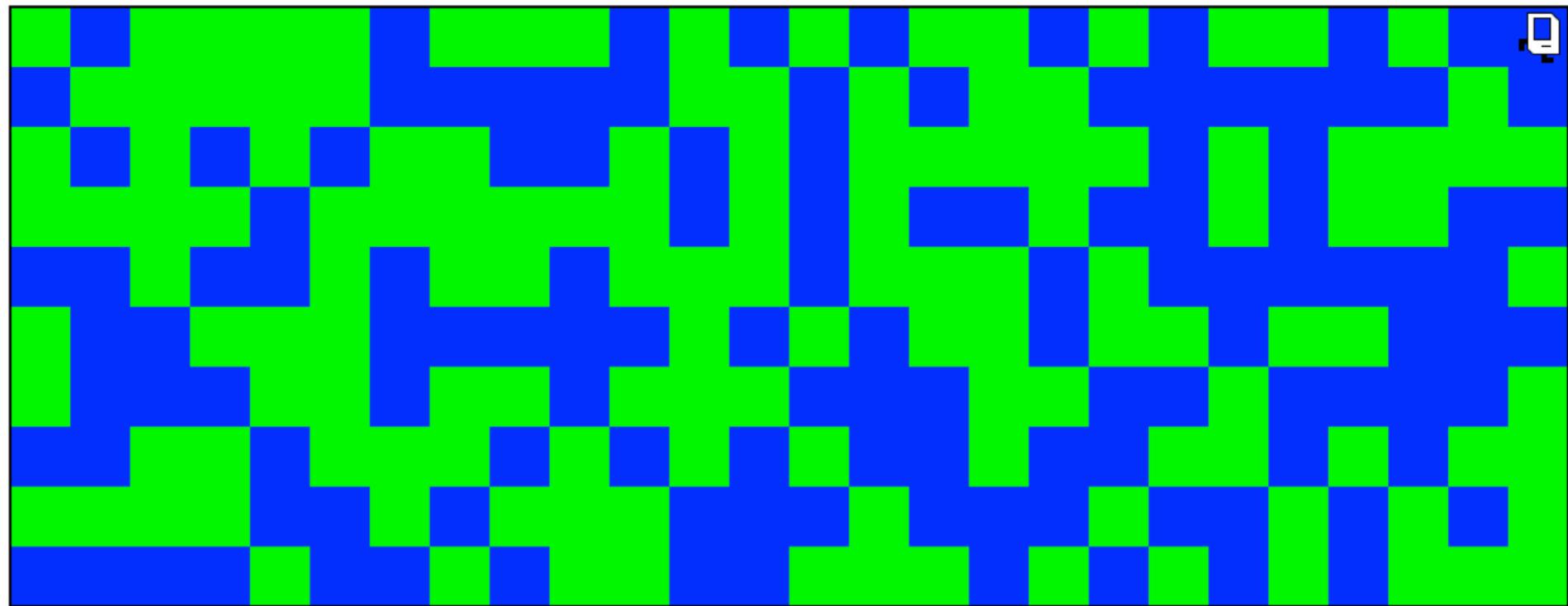
Nous pouvons également spécifier ce qui se passera si la condition est faux



Karel se rend à l'armée:

- S'il y a un bip, prenez-le
- S'il n'y a pas de bip, mettez-en un

Peintre au Hasard



Utiliser un if Déclaration!