Pour avoir le bytecode d'un fichier .class:
`javap -c nomDeFichier`
-c : pour avoir le détail des méthodes
nomDeFichier : un fichier .class (ne pas ajouter l'extension)

Pour compiler un fichier .j (jasmin) en .class:
`java -jar jasmin.jar nomDeFichier.j`

Pour exécuter ce fichier .class sur une JVM
`java nomDeFichier` (ne pas ajouter l'extension)

Squelette le plus simple d'un fichier .j
```
.class public Maclasse
.super java/lang/Object
.method public static main([Ljava/lang/String;)V
   .limit stack 99  ; Il ne faut pas mettre 99, mais la valeur minimum nécessaire
   .limit locals 99 ; Uniquement si on utilise des variables
   ; VOTRE CODE ICI
   return
.end method
```

Exemple pour afficher un entier :
```
getstatic java/lang/System/out Ljava/io/PrintStream;
bipush 4 ; Par exemple
invokevirtual java/io/PrintStream/println(I)V
```

Exemple pour afficher la lettre A:
```
getstatic java/lang/System/out Ljava/io/PrintStream;
ldc 65 ; équivalent de «A» en unicode
invokevirtual java/io/PrintStream/println(C)V
```

Exemple pour afficher un «coucou»
```
getstatic java/lang/System/out Ljava/io/PrintStream;
ldc "coucou"
invokevirtual java/io/PrintStream/println(Ljava/lang/String;)V
```

Définition d'une méthode générique (à faire en dehors du main):
```
.method public static nomMethod(type_paramètres)type_résultat
   .limit stack 99
   .limit locals 99
    ; VOTRE CODE
   return
.end method
```
`type_résultat` et `type_paramètres` sont de type: F, V, I , FF... (si aucun paramètre d'entrée, mettre **`()`**)
Par exemple, la méthode **`rajoute(II)I`** va prendre 2 entiers en paramètres (que l'on accédera dans la méthode grâce à `iload 0` et `iload 1`) et renverra un entier (avec `ireturn`).

Appel de la méthode
Si besoin, placer les paramètres au sommet de la pile
`invokestatic nomClass/nomMethod(type_paramètres) type_résultat`
Le résultat (s'il y en a un) est stocké au sommet de la pile, en remplacement des paramètres. Dans l'exemple précédent, il faudra 2 entiers sur la pile, puis on fait un **`invokestatic Maclasse/rajoute(II)I`**, ce qui supprimera ces deux entiers de la pile et mettra un autre (le retour de la méthode) sur la pile.