public static void main(String a[]) {
    int nMMSsTotal = 34;
    int nMMSsLeftOver = nMMSsTotal;
    int nPeople = 6;
    int nMMSsEach = 0;
    while( nMMSsLeftOver >= nPeople ) {
        nMMSsLeftOver = nMMSsLeftOver - nPeople;
        nMMSsEach = nMMSsEach + 1;
    }
}
Java (Virtual) Machine Code

0: bipush 34
2: istore_1
3: iload_1
4: istore_2
5: bipush 6
7: istore_3
8: iconst_0
9: istore_4
11: iload_2
12: iload_3
13: if_icmplt 29
16: iload_2
17: iload_3
18: isub
19: istore_2
20: iload_4
22: iconst_1
23: iadd
24: istore_4
26: goto 11
29: getstatic #2; //..Print..
public static void main(String a[]) {
    int nMMsTotal = 34;
    int nMMsLeftOver = nMMsTotal;
    int nPeople = 6;
    int nMMsEach = 0;
    while( nMMsLeftOver >= nPeople ) {
        nMMsLeftOver = nMMsLeftOver - nPeople;
        nMMsEach = nMMsEach + 1;
    }
}
Look at the Line Number Table

public static void main(java.lang.String[]);

LineNumberTable:
  line 5: 0
  line 6: 3
  line 7: 5
  line 8: 8
  line 9: 11
  line 11: 16
  line 12: 20
  line 14: 29
  line 20: 87

3 public static void main(String a[])
4 {           0: bipush 34
  5   int nMMsTotal = 34;

2:  istore_1

The javap program (for “Java-Print”, from the Java Development Kit) reads this information from inside a .class file and prints it so it's easier for humans to understand.
Look at the Local Variable Table

3 public static void main(String a[])
4 {
5   int nMMsTotal = 34;

LocalVariableTable:
Start  Length  Slot  Name       Signature
0      88      0    a       [Ljava/lang/String;
3      85      1    nMMsTotal       I
5      83      2    nMMsLeftOver       I
8      80      3    nPeople       I
11      77      4    nMMsEach       I

The Java compiler decided to use Slot 1 for local variable named nMMsTotal
How JVM does nMMsTotal=34;

0:   bipush 34   Push literal 34 on top of the stack.

2:   istore_1   Pop what's on top of the stack off and STORE it in slot number 1, which the Java compiler had chosen to use for variable named nMMsTotal.
What's this stack thing about?

- What do you think?
Now for the next line of Java code

```
public static void main(String a[])
{
  int nMMsTotal = 34;
  int nMMsLeftOver = nMMsTotal;

  bipush 34
  istore_1
  iload_1
  istore_2
```

LocalVariableTable:

<table>
<thead>
<tr>
<th>Slot</th>
<th>Name</th>
<th>Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>nMMsTotal</td>
<td>I</td>
</tr>
<tr>
<td>2</td>
<td>nMMsLeftOver</td>
<td>I</td>
</tr>
<tr>
<td>3</td>
<td>nPeople</td>
<td>I</td>
</tr>
</tbody>
</table>

LineNumberTable:

- line 5: 0
- line 6: 3
- line 7: 5
- line 8: 8
- line 9: 11
- line 11: 16
- line 12: 20
- line 14: 29
- line 20: 87
Lines 7 and 8 are just like 5

```java
public static void main(String a[])
{
    int nMMsTotal = 34;
    int nMMsLeftOver = nMMsTotal;
    int nPeople = 6;
    int nMMsEach = 0;
```

```
0:   bipush 34
2:   istore_1
3:   iload_1
4:   istore_2
5:   bipush 6
7:   istore_3
8:   iconst_0
9:   istore 4
```
And now for the loop!

```
9  while( nMMsLeftOver >= nPeople )
10  {
11     nMMsLeftOver = nMMsLeftOver – nPeople;
12     nMMsEach = nMMsEach + 1;
13  }
14  System.out....

11: iload_2
12: iload_3
13: if_icmplt 29
16: iload_2
17: iload_3
18: isub
19: istore_2
20: iload_4
22: iconst_1
23: iadd
24: istore_4
26: goto 11
29: getstatic #2; //..Print..
```
And now for the loop!

```java
9    while( nMMsLeftOver >= nPeople )
10    {
11        nMMsLeftOver = nMMsLeftOver – nPeople;
12        nMMsEach = nMMsEach + 1;
13    }
14    System.out....

11:  iload_2
12:  iload_3
13:  if_icmplt 29
16:  iload_2
17:  iload_3
18:  isub
19:  istore_2
20:  iload_4
22:  icnst_1
23:  iadd
24:  istore_4
26:  goto 11
29:  getstatic #2; //..Print..
```
And now for the loop!

```java
9    while( nMMsLeftOver >= nPeople )
10    {
11       nMMsLeftOver = nMMsLeftOver – nPeople;
12       nMMsEach = nMMsEach + 1;
13    }
14    System.out....
```

```assembly
11:  iload_2
12:  iload_3
13:  if_icmplt  29
16:  iload_2
17:  iload_3
18:  isub
19:  istore_2
20:  iload  4
22:  iconst_1
23:  iadd
24:  istore  4
26:  goto   11
29:  getstatic #2; //..Print..
```
And now for the loop!

9    while( nMMsLeftOver >= nPeople )
10    {
11        nMMsLeftOver = nMMsLeftOver – nPeople;
12        nMMsEach = nMMsEach + 1;
13    }
14    System.out....

11:  iload_2
12:  iload_3
13:  if_icmplt 29
16:  iload_2
17:  iload_3
18:  isub
19:  istore_2
20:  iload  4
22:  iconst_1
23:  iadd
24:  istore  4
26:  goto   11
29:  getstatic #2;  //..Print..
QUIZ What is each instruction for?

11_________________________
12_________________________
13_________________________
16_________________________
17_________________________
18_________________________
19_________________________
20_________________________
22_________________________
23_________________________
24_________________________
26_________________________