

Many Maxes in Array

Create code (in JJ or Java) to operate on an array named `arr` of `n` ints (say percents, having many repeated elements) and prints out the position of all of the possibly many maximum values. ReUse; you need not do declarations.

ReUse `maxArray` routine which finds the maximum of an array

```
max = maxArray (arr, n);
for (int p = 0; p < n; p++)
    if (max == arr[p])
        System.out.println (p);
```

Many Rounds in Time Class

For the `Time` instant class given in the lecture notes (with two parts: hour, mins) create methods to round off the time (to the nearest hour) in one of the following ways.

For example, 6:20 rounds to 6:00, 12:30 rounds to 13:00, and 23:40 rounds to 0:00.

Recall from Homework solutions, involving classes and methods

```
Function rounded (none) ofType int is public
    Box result ofType int
    -- Does: return the closest hour as an int!
    If (mins <= 30) then
        Set result = hour
    Else -- next hour, but not 25!
        Set result = (hour + 1) % 24
    EndIf
EndFunction rounded
```

Function `rounded (none) ofType Time` is an alternate

Routine `roundTime (none)` is also possible (mutable)

```
Routine round (none) -- roundTime above
-- Does change, or mutate, the object Time
-- Sets mins to 0, the hours to the nearest
```

```
    If (mins <= 30) then
        Inc hour by 1
    EndIf
    Set mins = 0
    If (hour > 23) then
        Set hour = 0
    EndIf
EndRoutine round
```

```
Function rounded (none) ofClass Time
-- Does return a new object of type Time
-- Does not change the original (immutable)
    Box result ofClass Time
    Boxes m, h ofType int
    If (mins <= 30) then
        Set h = hour
    Else
        Set h = hour + 1
        Set h = hour % 24
    EndIf
    New result ofClass Time with (h,0)
EndFunction rounded
```