

Recursive method tracing  
Stack based approach to  
2009 AP CS A Multiple Choice #39

By: Brandon Horn  
[apcstutoring.com](http://apcstutoring.com)

`recur` makes only 1 recursive call so it is not necessary to track line numbers.  
`recur` does maintain a local variable `dig` which must be tracked for each call.  
The initial call is `recur(32)` which is abbreviated `r(32)`.

`r(32)`

`r(32)` stores "2" as `dig` and calls `r(10)`.

`r(10)`

`r(32) dig = "2"`

`r(10)` stores "1" as `dig` and calls `r(3)`.

`r(3)`

`r(10)` `dig` = "1"

`r(32)` `dig` = "2"

r(1)

r(3) dig = "0"

r(10) dig = "1"

r(32) dig = "2"

`r(1)` stores "1" as `dig` and returns "1".

~~`r(1) dig = "1"`~~ returns "1"

`r(3) dig = "0"`

`r(10) dig = "1"`

`r(32) dig = "2"`

r(3) returns "01".

~~r(1) dig = "1"~~ returns "1"

~~r(3) dig = "0"~~ returns "01"

r(10) dig = "1"

r(32) dig = "2"

```
r(1) dig = "1" returns "1"  
r(3) dig = "0" returns "01"  
r(10) dig = "1" returns "101"  
r(32) dig = "2"
```



```
r(1) dig = "1" returns "1"  
r(3) dig = "0" returns "01"  
r(10) dig = "1" returns "101"  
r(32) dig = "2" returns "2101"
```