2021-10-31

```
int mySqrt(int x) {
    if (x == 0) return 0;
    if (x == 1) return 1;
    int left = 0, right = x, res = 0;
    while (right - left > 1) {
        int mid = (right - left) / 2 + left;
        if (x / mid < mid) {
            right = mid;
        } else {
            left = mid;
            res = left;
        }
    }
    return res;
}</pre>
```

Coding Notes:

Binary-Search "Sgrt(x)" int mySgrt(int x)? if (x=0) return 0; if (x=1) return 1; int left=0, right=x, res=0; while (right-left >1)? int mid=(right-left)/2+left; if (x/mid < mid)? right=mid; left=mid; res=(left; return res; } return res;	int mySgrt (int x)? if (x==0) return 0; if (x==1) Yeturn 1; int left=0, right= x, Yes=0; while (right-left>1)? int mid=(right-left)/2+left; if (x/mid < mid)? right=mid; left=mid;	midz=4 midz=8	
int mySgrt (int x) { if (x==0) return 0; if (x==1) return 1; int left=0, right= x, res=0; while (right-left >1) { int mid=(right-left)/2+left; if (x/mid < mid) { right=mid; } left=mid; res=(aft; } }	int mySgrt (int x)? if (x==0) return 0; if (x==1) Yeturn 1; int left=0, right= x, Yes=0; while (right-left>1)? int mid=(right-left)/2+left; if (x/mid < mid)? right=mid; }else? left=mid;	midz=4 midz=8	
int mySgrt (int x) { if (x==0) return 0; if (x==1) return 1; int left=0, right= x, res=0; while (right-left >1) { int mid=(right-left)/2+left; if (x/mid < mid) { right=mid; } left=mid; res=(aft; } }	int mySgrt (int x)? if (x==0) return 0; if (x==1) Yeturn 1; int left=0, right= x, Yes=0; while (right-left>1)? int mid=(right-left)/2+left; if (x/mid < mid)? right=mid; }else? left=mid;	midz=4 midz=8	
int mySgrt (int x) { if (x==0) return 0; if (x==1) return 1; int left=0, right= x, res=0; while (right-left >1) { int mid=(right-left)/2+left; if (x/mid < mid) { right=mid; } left=mid; res=(aft; } }	int mySgrt (int x)? if (x==0) return 0; if (x==1) Yeturn 1; int left=0, right= x, Yes=0; while (right-left>1)? int mid=(right-left)/2+left; if (x/mid < mid)? right=mid; }else? left=mid;	1 midz=4 midz={	
if (x==0) return 0; if (x==1) return 1; int left=0, right= x, res=0; while Lright- left > 1)? int mid= Lright- left)/2 + left; if (x/mid < mid); right=mid; left=mid; res=left; }	if (x==0) return 0; if (x==1) return 1; int left=0, right=x, res=0; while (right-left>1)? int mid=(right-left)/2+left; if (x/mid < mid)? right=mid; left=mid;	mid2=4 mid,={	
if (x==0) return 0; if cx==1) return 1; int left=0. right= x, res=0; while Lright- left > 1) { int mid= Lright- left)/2 + left; if (x/mid < mid) { right=mid; } else { left = mid; res= left; }	if (x==0) return 0; if (x==1) return 1; int left=0, right=x, res=0; while (right-left>1)? int mid=(right-left)/2+left; if (x/mid < mid)? right=mid; left=mid;	/ / / / / / / / / / / / / / / / / / /	
if (x==1) return 1; int left=0, right=x, res=0; while cright-left>1) \(\) int mid=(right-left)/2+left; if (x/mid < mid) \(\) right=mid; left=mid; res=left; }	if (x==1) return 1; int left=0, right=x, res=0; while (right-left>1)? int mid=(right-left)/2+left; if (x/mid < mid)? right=mid; left=mid;	/ / / / / / / / / / / / / / / / / / /	
int left=0, right=x, res=0; while (right-left>1)? int mid=(right-left)/2+left; if(x/mid < mid)? right=mid; } else? left=mid; res=(eft; }	int left=0, right=x, res=0; while (right-left>1)? int mid=(right-left)/2+left; if(x/mid < mid)? right=mid; left=mid;	/ / / / / / / / / / / / / / / / / / /	16
while (right-left > 1) \(\) int mid= (right-left)/2 + (eft; if (x/mid < mid) \(\) right= mid; left = mid; res = (eft; \]	while (right-left > 1) { int mid=(right-left)/2+left; if (x/mid < mid) { right=mid; }else { left=mid;	midz=4 mid,={	16
int mid= (right-(oft)/2+ (oft; if (x/mid < mid) { right=mid; } else { left=mid; res=(oft; }	int mid= (right-left)/2+ left; if(x/mid < mid) { right=mid; }else { left=mid;	midz=4 midj={	16
int mid= (right-(oft)/2+ (oft; if (x/mid < mid) { right=mid; } else { left=mid; res=(oft; }	int mid= (right-left)/2+ left; if(x/mid < mid) { right=mid; }else { left=mid;	770.02- 1 770.00/5	
if (x/mid < mid) {	if (x/mid < mid) { right=mid; } else { left = mid;		
right=mid; } else{ left = mid; res = left; }	right=mid; }else{ left = mid;		
} else {	}else{ left = mid;		
left = mid; res = (eft; }	left = mid;		
res = (aft; }			
3	res = left;		
return res; }			
return res; }	3		
return res; }	}		
	return resi		
	3		