

```

dynamicVar=0
counter=0
## funcOutput outputs the var in dynamic var
funcOutput () { echo "in dynamicVar: $dynamicVar"; }
## funcFiboChild calculates fibo elements recursively using the dynamicVar
## this show the dynamic scoping by using the startVar and dynamicVar in
recursive without passing the variable. Also uses the counter, which is a global
var to control the depth of recursion.
funcFiboChild(){
    sum=$((dynamicVar+startVar))
    startVar=$dynamicVar
    dynamicVar=$sum
    let counter=counter+1
    echo "recursion for $counter times"
    funcOutput
    if (($counter<10));then
        funcFiboChild
    fi
}
## funcFibo defines local var and calls funcFiboChild
funcFibo () { local startVar=1; local dynamicVar=1; funcFiboChild; }
## calls funcFibo, should call funcFiboChild for 10 times
funcFibo
## calls output directly, will output globalVar dynamicVar,which should be 0
echo "end of recursion, output the value of globals"
funcOutput

```

Pure Text version below

```

dynamicVar=0
counter=0
funcOutput () { echo "in dynamicVar: $dynamicVar"; }
funcFiboChild(){
    sum=$((dynamicVar+startVar))
    startVar=$dynamicVar
    dynamicVar=$sum
    let counter=counter+1
    echo "recursion for $counter times"
    funcOutput
    if (($counter<10));then
        funcFiboChild
    fi
}
funcFibo () { local startVar=1; local dynamicVar=1; funcFiboChild; }
funcFibo
echo "end of recursion, output the value of globals"
funcOutput

```

