```
alloc m 48                 : allocate space for double type array
func scale                 : a new function called scale
formal 8                   : allocate 8 bytes memory space for double x
localloc 4                 : allocate 5 bytes memory space for int i
bgnstmt 6                  : go to line 6
      t1 := param 0        : yield address of a parameter
      t2 := @f t1          : deference the address of t1
      t3 := 0              : define a constant value 0
      t4 := cvf t3         : convert t3 to float typt
      t5 := t2 ==f t4      : check if t2 is equal to t4
      bt t5 B1             : if t5 is true, then go to B1
      br B2                : else go to B2
label L1
bgnstmt 7
      t6 := 0              : define a constant value 0
      reti t6              : return t6
label L2:  ( bgnstmt 8??)
B1=L1
B2=L2
bgnstmt 8  (go to “bt t13 B3”)
      t7 := local 0        : yield space for local variable
      t8 := 0
      t9 := t7 =i t8       : set t7 to be 0
label L3:
      t10 := local 0       :yiled space for local variable
      t11 := @i t10        : deference int type value t10
      t12 := 6             : set t12 to be 6
      t13 := t11 <i t12    : check if t11 is less then 6
      bt t13 B3            : if t13 is true, then go to B3
      br B4                : else go to B4
label L4:
      t14 := local 0
      t15 := 1
      t16 := @i t14
      t17 := t16 +i t15    : i+1
```

```
        t18 := t14 =i t17   : i=i+1
        br B5               : go to B5
label L5:
bgnstmt 9 (go to y27:= t22 =f t26”)
        t19 := local 0
        t20 := @i t19
        t21 := global m
        t22 := t21 []f t20
        t23 := param 0
        t24 := @f t23
        t25 := @f t22
        t26 := t25 *f t24   :
        t27 := t22 =f t26   :m[i] =x * m[i]
        br B6
label L6
B3=L5
B4=L6
B5=L3
B6=L4
bgnstmt 10 (go to reti t28)
        t28 := 1
        reti t28
fend
```