```
int n;

int fact()
{
     int loc;
     if (n > 1)
     {
          loc = n--;
          return loc * fact();
     }
     else return 1;
}

main()
{
     read n;

     if (n >= 0)
          print fact();
     else
          print "error"
}
```

Code Segment

0	set 0, 3	initialize CURRENT
1	set 1, 5	initialize FREE
2	set 2, read	reads the value of n
3	jump 12, D[2] < 0	tests the value of n (if statement)
4	set 1, D[1] + 1	call to fact() starts here. Make space for the return value
5	set D[1], 6 + 4	set return pointer
6	set D[1] + 1, D[0]	set dynamic link
7	set 0, D[1]	set CURRENT
8	set 1, D[1] + 3	set FREE (3 is the size of the AR of fact())
9	jump 14	jump to the first instruction of fact()
10	set write, D[D[1]]	D[1] is the address of the returned value
11	jump 13	end of if statement (skip the “else”)
12	set write, “error”	else statement
13	halt	end of the code of main
14	jumpt 25, D[2] <= 1	This is the first instruction of fact()
15	set D[0] + 2, D[2]	assigns n to loc
16	set 2, D[2] - 1	decrements n
17	set 1, D[1] + 1	recursive call to fact() starts here. Make space for the return value
18	set D[1], ip + 4	set return pointer
19	set D[1] + 1, D[0]	set dynamic link
20	set 0, D[1]	set CURRENT
21	set 1, D[1] + 3	set FREE (3 is the size of the AR of fact())
22	jump 14	jump to the first instruction of fact()
23	set D[0]–1, D[D[0]+2]* D[D[1]]	store return value
24	jump 26	end of if statement (skip the “else”)
25	set D[0] – 1, 1	else statement. Return 1
26	set 1, D[0] - 1	return from routine (set FREE)
27	set 0, D[D[0] + 1]	return from routine (set CURRENT)
28	jump D[D[1] + 1]	return from routine (jump to the stored return pointer)