Computer Science Department
Teodor C. Przymusinski

Programming Languages

Midterm Exam
February 25, 2005
(100 pts.)

Last name:	*First name:*	*Social Security:*

1. (80pts) (a) (50 pts.) Write a complete SIMPLESEM code for the following program using the C3 language paradigm. (b) (30pts.) Manually run the resulting code and enter all the data in the data segment. Enter your code and data in the table provided below. Use pencil so that you can easily make necessary corrections. Use back sides of pages for scratch notes.

```
int m = 4;  int n = 16;

main()
{
   int m = 100
   if (n > 10  & m > 83) m = f()
   else m = 0
}

int f()
{
   int k = 3, int n = 4
   if (m == 1) return 4
    else
     {   n  = m--  + k
         return ( n * f())
     }
}
```

SIMPLESEM CODE SEGMENT FOR PROBLEM #1

Nr	Code	Comments
0.	set 0, 4	set current
1.	set 1, 7	set free
2.	set 2, 4	m = 4
3.	set 3, 16	n = 16
4.	set 6, 100	m = 100
5.	jumpt 14, D[3]<=10 or D[6]<=83	
6.	set 1, D[1] + 1	calling sequence
7.	set D[1], 8 + 4	
8.	set D[1]+1, D[0]	
9.	set 0, D[1]	
10.	set 1, D[1] + 4	
11.	jump 16	
12.	set 6, D[D[1]]	m = f()
13.	jump 15	
14.	set 6, 0	m = 6
15.	halt	
16.	set D[0]+2, 3	k = 3
17.	set D[0]+3, 4	n = 4
18.	jumpt 23, D[2] <> 1	
19.	set D[0]-1, 4	return 4
20.	set 1, D[0]-1	
21.	set 0, D[D[0]+1]	
22.	jump D[D[1]+1]	
23.	set D[0]+3, D[2] + D[D[0]+2]	n = m + k
24.	set 2, D[2]-1	m--
25.	set 1, D[1] + 1	calling sequence
26.	set D[1], 27 + 4	
27.	set D[1]+1, D[0]	
28.	set 0, D[1]	
29.	set 1, D[1] + 4	
30.	jump 16	
31.	set D[0]-1, D[D[0]+2]*D[D[1]]	return n*f()
32.	jump 20	
33.		
34.		
35.		
36.		
37.		
38.		
39.		
40.		
41.		
42.		
43.		
44.		
45.		

2. (20 pts.) [PROLOG] Suppose that we are given an ordering between elements specified by the predefined predicate `lesseq(A,B)`, meaning that A is less than or equal to B. Write a Prolog predicate `sort(List, SortedList)` which sorts the list `List` into a non-decreasing list `SortedList`, using the above ordering.

There are many ways to solve this problem. Two of them are presented here:

```
bubblesort( List, Sorted ) :-
swap( List, List1 ),!, bubblesort( List1, Sorted ).

bubblesort( Sorted, Sorted ).

swap( [ X, Y | Rest ], [ Y, X | Rest ] ) :- lesseq(Y,X).

swap( [ Z | Rest ], [ Z | Rest1 ] ) :-
swap( Rest, Rest1 ).
```

or, alternatively:

```
order([], []).
order([A|B], L) :- merge(B, [A], L).

merge([], L, L).
merge([A|B], [C|D] , L) :- lesseq(A,C), !, merge(B, [A|[C|D]], L).
merge([A|B], [C|D] , L) :- order([A|D],E), merge(B, [C |E], L).
```

SIMPLESEM DATA SEGMENT FOR PROBLEM #1

Nr.	Contents	Comments
0)	4/8/13/18/23/18/13/8/4	CURRENT
1)	7/8/12/13/17/18/22/23/27/22/17/12/7	FREE
2)	4/3/2/1	m (global)
3)	16	n
4)		R.P. Main
5)		D.L. Main
6)	100/108	m (main)
7)	108	3*36=108
8)	12	
9)	4	
10)	3	
11)	4/7	
12)	36	3*12=36
13)	31	
14)	8	
15)	3	
16)	4/6	
17)	12	3*4=12
18)	31	
19)	13	
20)	3	
21)	4/5	
22)	4	
23)	31	
24)	18	
25)	3	
26)	4	