

	Design suitable data structures and implement pass-I of a two-pass assembler for pseudo-machine in Java using object oriented feature.																																																																																																													
1	<pre>start 100 movr ax 05 mover bx 10 up: add ax bx movem a ='5' origin up ltorg movem b ='7' ds a 02 dc b 10 end</pre>																																																																																																													
	Implement Pass-II of two pass assembler for pseudo-machine in Java using object oriented features. The output of assignment-1 (intermediate file and symbol table) should be input for this assignment.																																																																																																													
2	<table><tr><td>(AD,1) (C,100)</td><td>SYMBOL</td><td>ADDRESS</td><td>LITERAL</td><td>ADDRESS</td></tr><tr><td>100 (IS,5) (RG,1) (C,05)</td><td>up</td><td>102</td><td>5</td><td>102</td></tr><tr><td>101 (IS,5) (RG,2) (C,10)</td><td>a</td><td>109</td><td>8</td><td>105</td></tr><tr><td>102 (S,1) (IS,2) (RG,1) (RG,2)</td><td>b</td><td>110</td><td>8</td><td>106</td></tr><tr><td>103 (IS,6) (S,2) (L,1)</td><td>c</td><td>111</td><td>7</td><td>113</td></tr><tr><td>104 (IS,4) (RG,1) (S,1)</td><td>next</td><td>102</td><td>8</td><td>114</td></tr><tr><td>105 (AD,3) (C,102)</td><td></td><td></td><td></td><td></td></tr><tr><td>102 (DL,1) (C,5)</td><td></td><td></td><td></td><td></td></tr><tr><td>103 (IS,6) (S,3) (L,2)</td><td></td><td></td><td></td><td></td></tr><tr><td>104 (IS,6) (S,4) (L,3)</td><td></td><td></td><td></td><td></td></tr><tr><td>105 (DL,1) (C,8)</td><td></td><td></td><td></td><td></td></tr><tr><td>106 (DL,1) (C,8)</td><td></td><td></td><td></td><td></td></tr><tr><td>107 (IS,6) (S,2) (L,4)</td><td></td><td></td><td></td><td></td></tr><tr><td>108 (IS,6) (S,3) (L,5)</td><td></td><td></td><td></td><td></td></tr><tr><td>109 (DL,1) (C,02)</td><td></td><td></td><td></td><td></td></tr><tr><td>110 (DL,2) (C,10)</td><td></td><td></td><td></td><td></td></tr><tr><td>111 (DL,1) (C,09)</td><td></td><td></td><td></td><td></td></tr><tr><td>112 (S,5) (AD,4) (S,1)</td><td></td><td></td><td></td><td></td></tr><tr><td>113 (AD,2)</td><td></td><td></td><td></td><td></td></tr><tr><td>(DL,1) (C,7)</td><td></td><td></td><td></td><td></td></tr><tr><td>114 (DL,1) (C,8)</td><td></td><td></td><td></td><td></td></tr></table>					(AD,1) (C,100)	SYMBOL	ADDRESS	LITERAL	ADDRESS	100 (IS,5) (RG,1) (C,05)	up	102	5	102	101 (IS,5) (RG,2) (C,10)	a	109	8	105	102 (S,1) (IS,2) (RG,1) (RG,2)	b	110	8	106	103 (IS,6) (S,2) (L,1)	c	111	7	113	104 (IS,4) (RG,1) (S,1)	next	102	8	114	105 (AD,3) (C,102)					102 (DL,1) (C,5)					103 (IS,6) (S,3) (L,2)					104 (IS,6) (S,4) (L,3)					105 (DL,1) (C,8)					106 (DL,1) (C,8)					107 (IS,6) (S,2) (L,4)					108 (IS,6) (S,3) (L,5)					109 (DL,1) (C,02)					110 (DL,2) (C,10)					111 (DL,1) (C,09)					112 (S,5) (AD,4) (S,1)					113 (AD,2)					(DL,1) (C,7)					114 (DL,1) (C,8)				
(AD,1) (C,100)	SYMBOL	ADDRESS	LITERAL	ADDRESS																																																																																																										
100 (IS,5) (RG,1) (C,05)	up	102	5	102																																																																																																										
101 (IS,5) (RG,2) (C,10)	a	109	8	105																																																																																																										
102 (S,1) (IS,2) (RG,1) (RG,2)	b	110	8	106																																																																																																										
103 (IS,6) (S,2) (L,1)	c	111	7	113																																																																																																										
104 (IS,4) (RG,1) (S,1)	next	102	8	114																																																																																																										
105 (AD,3) (C,102)																																																																																																														
102 (DL,1) (C,5)																																																																																																														
103 (IS,6) (S,3) (L,2)																																																																																																														
104 (IS,6) (S,4) (L,3)																																																																																																														
105 (DL,1) (C,8)																																																																																																														
106 (DL,1) (C,8)																																																																																																														
107 (IS,6) (S,2) (L,4)																																																																																																														
108 (IS,6) (S,3) (L,5)																																																																																																														
109 (DL,1) (C,02)																																																																																																														
110 (DL,2) (C,10)																																																																																																														
111 (DL,1) (C,09)																																																																																																														
112 (S,5) (AD,4) (S,1)																																																																																																														
113 (AD,2)																																																																																																														
(DL,1) (C,7)																																																																																																														
114 (DL,1) (C,8)																																																																																																														
	Design suitable data structures and implement Pass-I of a two-pass macro-processor.																																																																																																													
3	<pre>MACRO INCR &X, &Y, &REG1 = AREG MOVER &REG1, &X ADD &REG1, &Y MOVEM &REG1, &X MEND MACRO DECR &A, &B, &REG2 = BREG MOVER &REG2, &A SUB &REG2, &B MOVEM &REG2, &A MEND START 100 READ N1 READ N2 DECR N1, N2 INCR N1, N2 STOP N1 DS 1 N2 DS 2 END</pre>																																																																																																													

The output of Pass-I (MNT, MDT and intermediate code file without any macro definitions) should be input for Pass-II. Implement Pass-II of 2 pass Macro					
4	<table><tr><td>Input.txt MACRO INCR &X, &Y, &REG1 MOVER &REG1, &X ADD &REG1, &Y MOVEM &REG1, &X MEND MACRO DECR &A, &B, &REG2 MOVER &REG2, &A SUB &REG2, &B MOVEM &REG2, &A MEND START 100 READ N1 READ N2 INCR N1, N2 DECR N1, N3 STOP N1 DS 1 N2 DS 2 N3 DS 1 END</td><td>MDT.txt INCR &X &Y &REG1 = AREG MOVER #3 #1 ADD #3 #2 MOVEM #3 #1 MEND DECR &A &B &REG2 = BREG MOVER #6 #4 SUB #5 #5 MOVEM #6 #4 MEND</td><td>MNT.txt INCR 0 3 DECR 5 3</td><td>ARG.txt &X &Y &REG1 AREG &A &B &REG2 BREG</td></tr></table>	Input.txt MACRO INCR &X, &Y, ®1 MOVER ®1, &X ADD ®1, &Y MOVEM ®1, &X MEND MACRO DECR &A, &B, ®2 MOVER ®2, &A SUB ®2, &B MOVEM ®2, &A MEND START 100 READ N1 READ N2 INCR N1, N2 DECR N1, N3 STOP N1 DS 1 N2 DS 2 N3 DS 1 END	MDT.txt INCR &X &Y ®1 = AREG MOVER #3 #1 ADD #3 #2 MOVEM #3 #1 MEND DECR &A &B ®2 = BREG MOVER #6 #4 SUB #5 #5 MOVEM #6 #4 MEND	MNT.txt INCR 0 3 DECR 5 3	ARG.txt &X &Y ®1 AREG &A &B ®2 BREG
Input.txt MACRO INCR &X, &Y, ®1 MOVER ®1, &X ADD ®1, &Y MOVEM ®1, &X MEND MACRO DECR &A, &B, ®2 MOVER ®2, &A SUB ®2, &B MOVEM ®2, &A MEND START 100 READ N1 READ N2 INCR N1, N2 DECR N1, N3 STOP N1 DS 1 N2 DS 2 N3 DS 1 END	MDT.txt INCR &X &Y ®1 = AREG MOVER #3 #1 ADD #3 #2 MOVEM #3 #1 MEND DECR &A &B ®2 = BREG MOVER #6 #4 SUB #5 #5 MOVEM #6 #4 MEND	MNT.txt INCR 0 3 DECR 5 3	ARG.txt &X &Y ®1 AREG &A &B ®2 BREG		
5	<p>Write a program to create a Dynamic Link Library for any mathematical operation and write an application program to test it. (Java Native Interface / Use VB or VC++) .</p> <p>Design a Paper Prototyping for any Banking Website or App.</p>				
6	<p>Write a program to solve Classical Problems of Synchronization using Mutex and Semaphore (Reader Writer Problem)</p> <p>Design Paper Prototyping for any ERP system.</p>				
7	<p>Write a program to simulate CPU Scheduling Algorithms: FCFS</p> <p>Process AT BT</p> <p>P1 10 2</p> <p>P2 0 10</p> <p>P3 8 4</p> <p>P4 5 5</p> <p>Design GUI using Python for student Registration Form.(Use Text ,Label,Checkbutton,List box etc)</p>				
8	<p>Write a program to simulate CPU Scheduling Algorithms: SJF (Preemptive)</p> <p>Process AT BT</p> <p>P1 10 2</p> <p>P2 0 10</p> <p>P3 8 4</p> <p>P4 5 5</p> <p>Design Paper Prototyping for any Shopping App or Website.</p>				
9	<p>Write a program to simulate CPU Scheduling Algorithms: SJF (Non-Preemptive)</p> <p>Process AT BT</p> <p>P1 10 2</p> <p>P2 0 10</p> <p>P3 8 4</p> <p>P4 5 5</p> <p>Design GUI using Python for Login Window.</p>				

10	<p>Write a program to simulate CPU Scheduling Algorithms: Priority (Non-Preemptive). Process AT BT P1 10 2 P2 0 10 P3 8 4 P4 5 5</p> <p>Design GUI using Python for online Quiz.(Use Text ,Label,Checkbutton, etc)</p>
11	<p>Write a program to simulate CPU Scheduling Algorithms: Priority (Non-Preemptive). Process AT BT P1 10 2 P2 0 10 P3 8 4 P4 5 5</p> <p>Design GUI using Python for sign-up Window</p>
12	<p>Write a program to simulate CPU Scheduling Algorithms: Round Robin. (TQ=1 Sec) Process AT BT P1 10 2 P2 0 10 P3 8 4 P4 5 5</p> <p>Design GUI using Python for customer Feedback Form about Food in Hotel .(Use Text ,Label,Checkbutton, etc)</p>
13	<p>Write a program to simulate Memory placement strategies – best fit, first fit. Design a GUI in Python of any screen for fund Tranfer/Transaction.</p>
14	<p>Write a program to simulate Memory placement strategies – best fit, worst fit. Design a GUI in python for Patient Registration Form in Hospital .(Use Text ,Label,Checkbutton,List box etc)</p>
15	<p>Write a program to simulate Page replacement algorithm. 1. FIFO Input reference String :- 2 3 2 1 5 2 4 5 3 2 5 2 No. of frames are:- 3</p> <p>Design a GUI in python for Help screen of any App.</p>
16	<p>Write a program to simulate Page replacement algorithm. 2. OPTIMAL Input reference String :- 2 3 2 1 5 2 4 5 3 2 5 2 No. of frames are:- 3</p> <p>Design a GUI in Python for Welcome screen.</p>
17	<p>Write a program to simulate Page replacement algorithm. 3. LRU Input reference String :- 2 3 2 1 5 2 4 5 3 2 5 2 No. of frames are:- 3</p> <p>Design a GUI in python for Sports Academy Registration Form</p>
18	<p>Write a program to implement Deadlock Avoidance Algorithm Design GUI using Python for Cab/Auto Booking App</p>