PA #3
DALTON TARDE
824357515

Q1. What data-structures linker will use and what will be the values of all the variables and data-structures at the end of pass 1 of linkage editor?

Location	Label	Operation	Operand	Rough work	
	SUM	START	00000		
		EXTDEF	NUM		
		EXTREF	RSSUMI, RSSUMF		
	FIRST	STL	RETADR		
		LDA	#1	Load #1 = A	
		STA	FLAG	store 1 @ 29	
		LDA	NUM	A = 20	
		+JSUB	FNSUMI	FNSUMI = 20	
		+JSUB	FNSUMF	FNSUMF = 20	
		+LDA	RSSUMI	A = 19	
		+COMP	RSSUMF	19 → 210	
		JEQ	EXMAIN	jmp	
		CLEAR	A	Clear A	
		STA	FLAG	stort 1 @ 0x0029	
	EXMAIN	J	@RETADR		
	RETADR	RESW	1		
	FLAG	RESW	1		
	NUM	WORD	20		
		END	FIRST		
	FNSUMI	START	0		
.SUBROUTINE THAT SUMS FIRST 20 INTS BY ADDING THEM					
		EXTDEF	RSSUMI		
		EXTREF	NUM		
		+LDX	NUM	load 20 = x	
		LDS	#1	load 1 = S	
		CLEAR	A	Clear A	
	LOOP	COMPR	X,S	20 \ 1	
		JLT	EXSUB1		
		ADDR	X,A	n = 20	
		SUBR	S,X	x = 20-- = 19	
		J	LOOP		
		STA	RSSUMI	Store 19 @ 0x0050	
	EXSUB1	RSUB			
	RSSUMI	RESW	1		
		END			
	FNSUMF	START	0		
.SUBROUTINE THAT SUMS FIRST 20 INTS BY FORMULA					
		EXTDEF	RSSUMF		
		EXTREF	NUM		
		+LDA	NUM	A = 20	
		ADD	#1	A = 21	
		MUL	NUM	A = 420	
		DIV	#2	A = 210	
		STA	RSSUMF	sture 210 @ 0068	
		RSUB			
	RSSUMF	RESW	1		
		END			

Q2.

LINE	Label	Operation	Operand
10		START	0
20	SWAP	MACRO	&A,&B,&C
30		ABC	‘&C’
40		LDA	‘&A’
50		STA	*+15
60		LDA	‘&B’
70		STA	‘&A’
80		LDA	*+6
90		STA	‘&B’
100		RESW	1
110		MEND	
120			
130	ABC	MACRO	&A
140	$LOOP	TD	=X’&A’
150		JEQ	*-3
160		RD	=X’&A’
170		STCH	*+3
180		RESB	1
190		MEND	
		…	
220		SWAP	INT1,INT2,IN
		…	
270		ABC	

What is the value of different variables and data structures when macro processor processed upto :

1) Line 110
2) Line 180
3) Line 220
4) Line 270

1) Line 110 - Closes MACRO (line 20) through MEND directive

2) Line 180 - RESB assigned a value of 1, data structure Passes Pointer by reference

3) Line 220 - Swap Calls back to MACRO, Functioning as memory Points for HL pair Storage

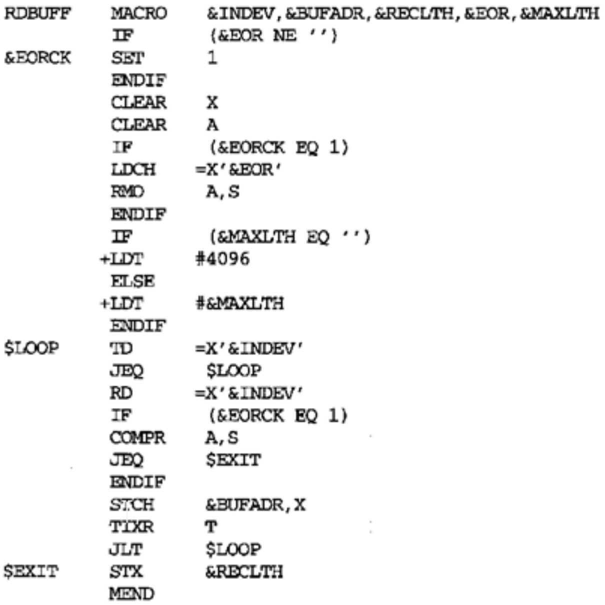
4) Line 270 - ABC is declared (non-operation)

Q3.

1. **The Macro definition is shown as follows. Please expand two consecutive macro invocations: (3 points)**

RDBUFF F1, BUFFER, LENGTH, 00, 1024

RDBUFF 0F, PDATA, SIZE, , 4096



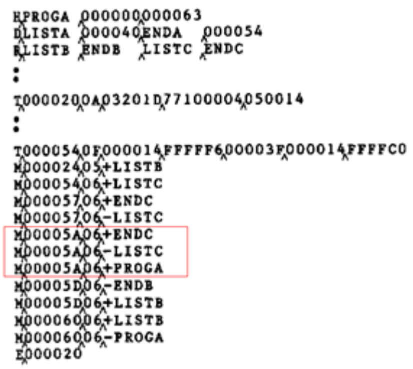
```
RDBUFF    MACRO    &INDEV,&BUFADR,&RECLTH,&EOR,&MAXLTH
          IF       (&EOR NE '')
&EORCK    SET      1
          ENDIF
          CLEAR    X
          CLEAR    A
          IF       (&EORCK EQ 1)
          LDCH     =X'&EOR'
          RMO      A,S
          ENDIF
          IF       (&MAXLTH EQ '')
         +LDT      #4096
          ELSE
         +LDT      #&MAXLTH
          ENDIF
$LOOP     TD       =X'&INDEV'
          JEQ      $LOOP
          RD       =X'&INDEV'
          IF       (&EORCK EQ 1)
          COMPR    A,S
          JEQ      $EXIT
          ENDIF
          STCH     &BUFADR,X
          TIXR     T
          JLT      $LOOP
$EXIT     STX      &RECLTH
          MEND
```

```
RDBUFF F1, BUFFER, LENGTH, 00, 1024
       EOR CK = 1
       CLEAR X
       CLEAR A
       LDCH = X '00'
       RMO A, S
$LOOP  TD = X 'F1'
       JEQ = $LOOP
       RD = X'F1'
       COMPR A, S
       JEQ $EXIT
       STCH BUFFER, X
       TIXR T
       JLT $LOOP
$EXIT  STX LENGTH
```

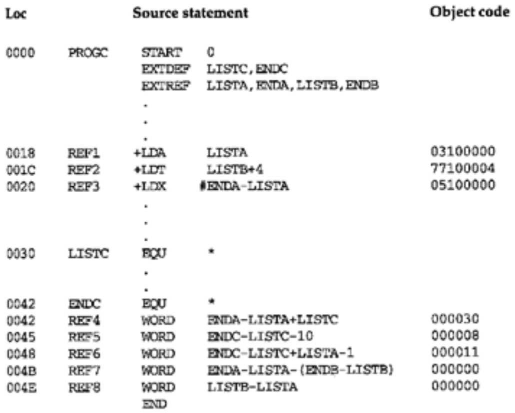
```
RDBUFF OF, PDATA, SIZE, , 4096
       CLEAR X
       CLEAR A
       +LDT 4096
       ELSE
       +:LDT 4096
$LOOP  TD = X 'OF'
       JEQ $LOOP
       RD = X 'OF'
       STCH PDATA, X
       TIXR T
       JLT $LOOP
$EXIT  STX SIZE
```

Q4.

1. **ProgA is shown in the following Object Program. We need to determine the <u>content</u> at the offset of "00005A" after linking and loading. This content is a word (3 bytes). Assume ProgA is loaded to address 0x4000, and ProgC is loaded to 0x40E2. We also coincidentally found the source code of ProgC, which could be logically useful for solving this problem. What could be the content at the offset of "00005A" of ProgA after linking and loading? (3 points)**



```
H^PROGA ^000000^000063
D^LISTA ^000040^ENDA  ^000054
R^LISTB ^ENDB  ^LISTC ^ENDC
:

T^000020^0A^03201D^77100004^050014
:

T^000054^0F^000014^FFFFF6^00003F^000014^FFFFC0
M^000024^05^+LISTB
M^000054^06^+LISTC
M^000057^06^+ENDC
M^000057^06^-LISTC
M^00005A^06^+ENDC
M^00005A^06^-LISTC
M^00005A^06^+PROGA
M^00005D^06^-ENDB
M^00005D^06^+LISTB
M^000060^06^+LISTB
M^000060^06^-PROGA
E^000020
```



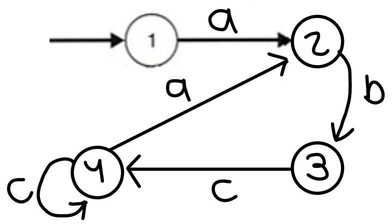
```
Loc             Source statement                          Object code

0000    PROGC    START    0
                 EXTDEF   LISTC,ENDC
                 EXTREF   LISTA,ENDA,LISTB,ENDB
                 .
                 .
                 .
0018    REF1    +LDA      LISTA                           03100000
001C    REF2    +LDT      LISTB+4                         77100004
0020    REF3    +LDX     #ENDA-LISTA                      05100000
                 .
                 .
                 .
0030    LISTC    EQU      *
                 .
                 .
0042    ENDC     EQU      *
0042    REF4     WORD     ENDA-LISTA+LISTC                000030
0045    REF5     WORD     ENDC-LISTC-10                   000008
0048    REF6     WORD     ENDC-LISTC+LISTA-1              000011
004B    REF7     WORD     ENDA-LISTA-(ENDB-LISTB)         000000
004E    REF8     WORD     LISTB-LISTA                     000000
                 END
```

The content at the offset of "00005A" is $\boxed{0x4012}$

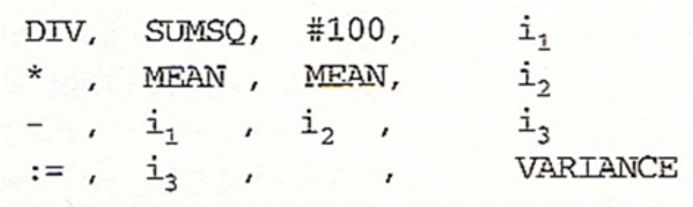
$$
\begin{array}{rl}
 & 0x4124 \quad (ENDC) \\
- & 0x4112 \quad (LISTC) \\
+ & \underline{0x4000} \quad (PROGA) \\
 & 0x4012
\end{array}
$$

1. Given the tabular representation of finite automaton, draw/finish the figure (diagram) representation of the finite automaton with state transition. 1 – initial state, 4 – final state. (2 points)

State	a	b	c
1	2		
2		3	
3			4
4	2		4



2. Given the quadruples, write the high level language in one statement, removing all intermediate variables of i1, i2, and i3. (2 points)



```
DIV,  SUMSQ,  #100,    i_1
*  ,  MEAN ,  MEAN,    i_2
-  ,  i_1  ,  i_2  ,   i_3
:= ,  i_3  ,       ,   VARIANCE
```

VARIANCE

=SUMSQ/100 – MEAN×MEAN