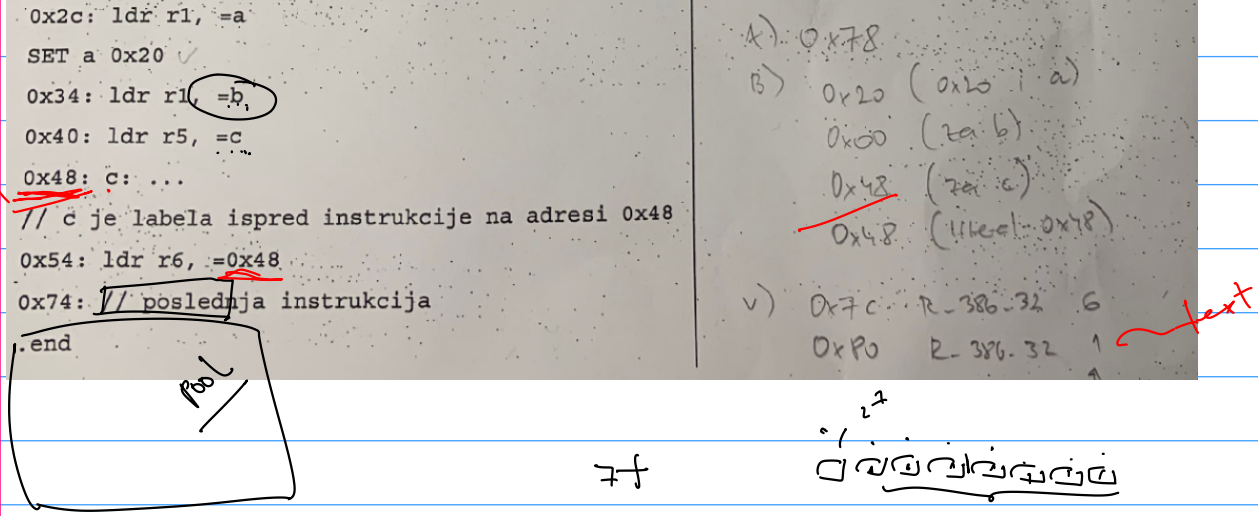
3.[10] Посматра се 32-битни процесор који нема подршку за непосредно адресирање за било коју 32-битну константу. Један од начина адресирања које процесор подржава је регистарско индиректно са померајем, с тим да се померај специфицира 8-битном означеном вредношћу и рачуна се у односу на другу инструкцију после посматране инструкције (све инструкције су величине 32 бита, а адресибилна јединица је бајт). Као замена за недостајуће непосредно адресирање, асемблер пружа подршку програмеру у виду литералних константи и базена литерала. Асемблер има подршку и за симболичке литерале у виду директиве SETA <simbol> <vrednost> која уводи нову литералну константу видљиву у читавом програму. Једном дефинисана вредност симболичког литерала се не може редефинисати. Асемблер оптимизује базен литерала тако да се у њему не понављају вредности ако није неопходно. За сегмент кода дат у наставку дати одговоре на постављена питања. Ради краћег записивања, дате су само релевантне линије кода, а бројчаним лабелама испред линије наведене су адресе на којима ће се налазити превод посматране инструкције (између наведених инструкција се налазе и друге инструкције које нису од значаја за задатак).

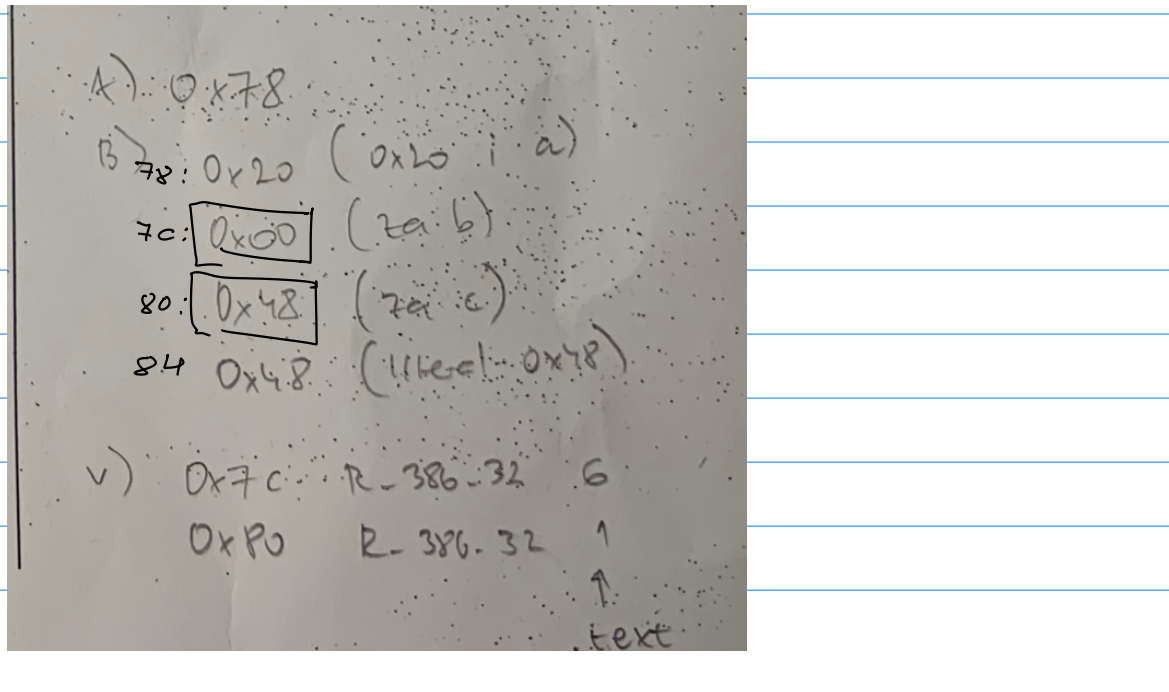
```
.public b
// b nije definisan u posmatranom fajlu
.text
0x20: ldr r1, =0x20
// upotreba literalne konstante
// (efektivno u r1 ce biti upisano 0x20)
0x2c: ldr r1, =a
SET a 0x20
0x34: ldr r1, =b
0x40: ldr r5, =c
0x48: c: ...
// c je labela ispred instrukcije na adresi 0x48
0x54: ldr r6, =0x48
0x74: // poslednja instrukcija
.end
```

а) На којој адреси се налази базен литерала?

б) Приказати табелу литерала након завршеног рада асемблера.

в) Под претпоставком да су симболи a, b и c у табели симбола на редним бројевима 5, 6 и 7, респективно, навести релокационе записе који ће настати од наведених инструкција.





```
.public b
// b nije definisan u posmatranom fajlu
.text
0x20: ldr r1, =0x20
// upotreba literalne konstante
// (efektivno u r1 ce biti upisano 0x20)
0x2c: ldr r1, =a
SET a 0x20
0x34: ldr r1, =b
0x40: ldr r5, =c
0x48: c: ...
// c je labela ispred instrukcije na adresi 0x48
0x54: ldr r6, =0x48
0x74: // poslednja instrukcija
.end
```

PC: = 0x28

0x78
− 0x28
0x50

PC: = 34

0x78
− 0x34
0x44

8
1000

5

0x78: 20 00 00 00
0x7c | 00 00 00 00 |
0x80 | 48 00 00 00 |
0x84 48 00 00 00

vrednost simbola b

ABS32

ABS32

.rela.text

0x7c , ABS32 , b

0x80 , ABS32 , .text .

3.[10] Посматра се једнопролазни асемблер са подршком за литералне и симболичке литералне константе. За сваку симболичку литералну константу је могуће везати највише једну вредност у читавом фајлу, а сам симбол је могуће користити и пре дефиниције. Асемблер оптимизује базен литерала елиминишући дуплирање исте вредности кад год је то могуће. На улаз асемблера се доводи код дат са десне стране. Све инструкције су 32-битне, а LDR инструкција користи операциони код 0x123 (највиших 12 битова инструкције), за којим следи редни број одредишног регистра (0-15), затим тип адресирања у 4 бита (регистарско индиректно са померајем – вредност 0xB), затим редни број регистра који се користи за адресирање података (ако се користи) и на крају 8-битни померај, уколико се користи. Регистар *PC* је са редним бројем 15, а у случају *PC* релативног адресирања адреса се рачуна у односу на следећу инструкцију.

```
.data
   .long 0x20
A: .long 0x30

.text
   LDR R0, =B
B: LDR R1, =A
   LDR R5, =D
   LDR R2, =C
   LDR R3, =4
C .SETA 4
   LDR R4, =C
D .EQU A
.end
```

Cook patching

а) [3] Под претпоставком да ће преправљање кода да се ради на крају пролаза кроз код, приказати табелу обраћања унапред, заједно са објашњењима евентуално потребних измена у структури података табеле обраћања унапред.

б) [4] Приказати садржај базена литерала у формату *адреса:садржај*. Водити рачуна да се вредности у базену литерала појављују по редоследу додавања у исти, што се ради у најранијем могућем тренутку.

в) [3] Приказати табелу релокација. Уместо редних бројева симбола навести називе симбола који се користе у релокационом запису.

0, 1, - - -, f

reg PC

00 00 00 00
12 3D E/F

18 − 4 = 14
1C − 8

Асемблерска линија	Адреса	Садржај							
.long 0x20	0	20	00	00	00		} .data		} .data
.long 0x30	4	30	00	00	00				
ldr R0, =B	0	12	30	9f	14	0	} .text		
ldr R1, =A	4	12	31	9f	14	1			
ldr R5, =D	8	12	35	9f	10	1			
ldr R2, =C	C	12	32	9f	10	2			
ldr R3, =4	10	12	33	9f	C	2			
ldr R4, =C	14	12	34	9f	8	2			
	18	04	00	00	00		0	} litpool	
	1C	04	00	00	00		1		
	20	04	00	00	00		2		

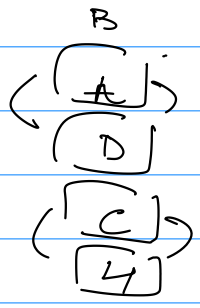
```
.data
    .long 0x20
A:  .long 0x30

.text
    LDR R0, =B
B:  LDR R1, =A
    LDR R5, =D
    LDR R2, =C
    LDR R3, =4
C   .SETA 4
    LDR R4, =C
D   .EQU A
.end
```

Симбол	Секција	Вредност	Видљивост
A	.data	4	e
B	.text	4	l
C	ABS	4	e
D	.data	4	e

БАЗЕН ЛИТЕРАЛА

Индекс	Померај
0	0
1	4
2	8
3	C
4	10

Вредност:



Референцирање БАЗЕНА ЛИТЕРАЛА.

Померај	Индекс у базену литерала	
3	0	
7	1	1
6	2	
f	3	
13	4	2
17	4	

literal pool

(0) 0 : .text + 4

(1) 4 : .data + 4

(2) 8 : 4

relocations for literal pool.

18 , ABS32 , .text

1C , ABS32 , .data

3.[10] Посматра се асемблер процесора код којег су све инструкције ширине 32 бита. Адресни простор је величине 4GB, а адресибилна јединица је бајт. У посматраном процесору не постоји подршка за непосредно адресирање, већ асемблер пружа подршку за литералне константе у нумеричком и симболичком облику. Константе се учитавају користећи LDA dstREG, adrREG[pomjeraj] која у регистар наведен као први операнд уписује вриједност прочитану из меморије са адресе која се добија када се саберу adrREG регистар и помјерај наведен у инструкцији. За помјерај су у инструкцији предвиђена 8 битова чији садржај се посматра као означена вредност у другом комплементу двојке. У току извршавања једне инструкције у регистру PC се налази адреса следеће инструкције. Асемблер оптимизује базен литерала тако да се у њему једна вредност појављује највише једном, уколико је то могуће. Приказати помјераје у наведеним инструкцијама, као и садржај базена литерала на крају рада асемблера. Напомена: наведени су само делови асемблерског кода, а бројеви дати на почетку реда представљају редни број инструкције.

```
.text
1. LDA R0, =56
...
5. LDA R1, =A
...
7. LDA R5, =B
8. LDA R6, B
A SETA 56  ;ova linija ne sadrzi instrukciju, zbog cega nema redni broj
...
15. B:  .LONG 5
...
20. ... ; poslednja instrukcija u text sekciji
.data
5C: .LONG 24
.end
```

БАЗЕН

Асемблерска линија	Адреса	Садржај							
lda R0, =56	0	?	?	?	4C				
	...								
lda R1, =A	10	?	?	?	3C				
	...								
lda R5, =B	18	?	?	?	38				
lda R6, B	1C	?	?	?	18				
	...								
B: .long 5	38	05	00	00	00				
	...								
	4C	??	??	??	??				
	50	38	00	00	00				
	54	38	00	00	00				

litpool (56)
litpool (56)
litpool (13)
B

Симбол	Секција	Вредност	Видљивост
B	.text	38	L
A	ABS	56	L
C	.data	10	L

ЕЛФ релокациони записи .text секције

Офсет	Тип	Симбол
54	ABS 32	.text