```
HW 2
A. J a. cladd $to, $zero, $zero
                                     # 4=0
                                     # store i in as
           sd
                 $ as, 7000 ($ to)
           addi $t1, $zero, 100
                                     # $ = 100
                                     # get i
    loup: Id
                 $ a3, 7000 ($ to)
          delt
                $ t2, $t1, $ a3
                                    #if 100 < i
                 $ t2  szero. EXIT # when 100ct, EXIT
          bne
          dsll
                 $t2, $03, 3
                                   # t2= i * 8
               $t3, $t2, 3000
                                   # t3 = B+ j*4
         daddi
          ld
                $ a1, 0($t2)
                                   # t3= 8[i]
                                   # az = C
          Id
                 $ az, 5000 ($to)
        dadd
                $t2, $a1, $az
                                  # t2 = Bli] + C
                                   # Cw = A + 1 * 4
        daddi
               $ Cw, $ t2, 1000
         scl
                                  # A[i] = B[i] + C
                $ t3, 0 ($00)
                                  # +++
        daddi $ a3,$a3, 1
         scl
                                  # store i
                $ as, 7000 ($to)
   EXIT:
  instructions: 3+13×10/+3=1319
   memory-data references: 1+5×10 = 50b
   code size in bytes: 16 × 32/8 = 64
                                    # i=0
             $0x0, %rax
b. mov g
                                   # base pointer=0
             $000, %rbp
     morg
                                      # store i
             % rax, Dx 1658 (% rbp)
                                      # get i
# v = 101
             Ox1658 (%rbp), % rax
loop: mova
             $0x65 , % rax
     cmpg
              EXIT
                                      # if i:/ol, EX17
     je
             %rax, %rbx
                                      # rbx = 7
     mov
                                      # rbx = xx 8
             $0x3, 9, rbx
                                      # 13[4]
             Ox 608(% rbx), %rcx
     mova
                                      # get C
# Bi]+C
     mova
             Ox 1385 (% rbp). %rdx
     add
             %rdx, %cdx
             % rax, % rbx
                                      # rbx = i
     mov
             $0x3, %rbx
     she
                                      # rbx=i*8
             % rcx, Ox3e8c% rbx)
                                      HALi] + Bli]+C
     morg
             garax, Brbx
                                      # rbx=i
                                      # 1++
      add
             Sorl, Torbx
              %rbx, 0x (bs 81%rbp)
                                      # sawe i
              900
  instructions: 3+15x10+3=152
  memory-data references: 1+5×121= tob
  code size in pytes: 18 x 32/8 = 72
```

1.18 4	· La ala.	1 00	lator	11	n40rW -	1.00/ /		
A.18 a. S			mulator:	Memory-me	will the state of	Load-store:		
	wh B	Load	B	Add A, 1		Load RI, B		
	ush C	Acla	C *	Add B.		Load Rz, C		*
	dd *		AO	Sub D. A		Add Rs, RI.		<u>) ^ </u>
7-	op A	O Add	C 0 1			Store A, R3		
P	ush A	Store	BOA			Add RI, R3.		*
	ush C		A *			store B, RI		
A	dd to	Nega	fe .	A=C+C	5	ub R4, R3.	RI O	*
	9p B		DA	D= A-B	5	Store D, R4	A	
	wh B			b=DfA				
	wh A							
	eb *							
P.	D D							
h A	Tualue,	is bounded t	rom momord	after housing	been laur	dod ance : 1		
W. ~	vume i	s much !	Torri menusa	aguer naving	(In	h Acourtleto	r 11	w Marke
TI	17.	1	<i>I</i>	1/- 1/	>tue	ek, Aummun	, memor	y-memory O 4. Leed store
/ne	resurt o	t one instru	ution re pas	sed to anoth	er instructu	on as an ope,	rana:	0
1		1			Stack, Accu	unulator, Memo	y-memori	f, Loud store
Stol	age with	run the pr	o Cessor: X			Memory-memory 16/8 = 3 = 1 16/8 = 3 = 1 16/8 = 2 = 24 tor Memory 14		
_sto1	age in	memory:	**		. /			
C	0	U	9	Stack Acc	unu lator	Memory-memory	Local-	store
instr	ution bytes	s be fetched	16/8 × 12	= 24 16/8.	×8=16	1618,3=+	1618×8	r=16
dati	a be trans	terred from l	to meniory it	18 = 9 = 18 16	18×4=8	1618 = 9 = 18	1618 * 5	==10
totei	1 memory	e traffic	24 + 18 =	42 16-	8=24	6+18=24	16+10	= 2b
50	Accumulat	or and Men	1 ord - Me mord	agre must e	Hicient.			
А.			0	Storek	Accumula	tor Memory	- พ.ค.ค.มาน	Loud-store
Just	mustion b	ufor he fetch	had b	110x1>= 7L	1418 × 8 = 6	4 64/8 3	= >4	6418×8=64
dala	1 L	yes ve tera	a Ho man hi Ord	1418 08 - 72	1/18 x H = 2	32 64/8 %]	-7,	64/8×J=40
1. L	r re tru	usiferred from	1 100 menusy	1172-11-8	111 123 - 8	1 1412	01	64 + 40 = 104
tvu	ii miru	ing tradice)"	BT (20 100	641322	b 24+721=	16	047 60-107