COMPUTER ARCHITECTURE ASSIGNMENT - 2

We have implemented the following programs:

- 1. Finding the factorial of a number
- 2. Sorting an array using selection sort
- 3. Searching a number in an array using Binary Search

FACTORIAL OF A NUMBER

INTRODUCTION:

Factorial of a number is defined as the product of all positive integers less than or equal to a given positive integer (Factorial of 0 is defined as 1).

Our program calculates the factorial of the given number. Assumption:

The input is a non-negative number.

PROGRAM EXPLANATION:

In our program we have four labels: main, check, loop and exit.

main: Stores the necessary values of the input, memory address of input, the value of the iterator \$s1 and stores the value 1 in \$s2 which will be used to compute factorial. **check:** Check if the iterator is less than our number or not and branch accordingly.

loop: Multiplies the current product value(\$s2) to the iterator \$s1 and then increments the iterator.

exit: Stores the result of factorial in data memory and exits from the program.

Instructions Used:

- Load Word (lw)
- Store Word (sw)
- Add immediate (addi)
- Multiply (mul)
- Branch on not equal (bne)
- Jump (j)

ASSEMBLY CODE:

```
main:
 2
3
4
5
                 addi $t0, $0, 10
addi $t1, $0, 268500992
sw $t0, 0($t1)
lw $s0, 0($t1)
addi $s1, $0, 1
addi $s2, $0, 1
                                                                         #t0 = n = 10
                                                                        #t0 = n = 10

#t1 = memory address where n will be stored

#M[t1] = t0 = 10

#s0 = M[t1] = n = 10

#s1 = i = 1
 6
 7
 8 9
                                                                        #s2 = fact(n) = 1
                 j check
 10
      check:
                            #checks the loop condition
11
12
13
                 bne $s1, $s0, loop
                                                            #if (i!=n) jump to loop
14
                 mul $s2, $s2, $s0
                                                            #else fact = fact * n because the loop function will execute only till (n-1)
 15
                 j exit
17
      loop:
                 mul $s2, $s2, $s1
addi $s1, $s1, 1
                                                            # fact = fact * i
18
19
                                                            \# i = i+1
                 j check
20
21
22
23
24
25
26
      exit:
                 addi $t0, $0, 268501024
sw $s2, 0($t0)
addi $v0,$0,10
                                                            #t0 = address where the result will be stored
                                                            \#M[t0] = s2
                                                            # return 0
                 syscall
```

TEXT SEGMENT:

3kpt	Basic	Address	Code		Source
	addi \$8,\$0,10	4194304	0x2008000a 3:	addi \$t0, \$0, 10	#t0 = n = 10
	lui \$1,4097	4194308	0x3c011001 4:	addi \$t1, \$0, 268500992	<pre>#tl = memory address where n will be st</pre>
	ori \$1,\$1,0	4194312	0x34210000		
9	add \$9,\$0,\$1	4194316	0x00014820		
	sw \$8,0(\$9)	4194320	0xad280000 5:	sw \$t0, 0(\$t1)	#M[t1] = t0 = 10
	lw \$16,0(\$9)	4194324	0x8d300000 6:	lw \$s0, 0(\$t1)	#s0 = M[t1] = n = 10
9	addi \$17,\$0,1	4194328	0x20110001 7:	addi \$sl, \$0, l	#sl = i = 1
2	addi \$18,\$0,1	4194332	0x20120001 8:	addi \$s2, \$0, 1	#s2 = fact(n) = 1
2	j 4194340	4194336	0x08100009 9:	j check	
	bne \$17,\$16,2	4194340	0x1630000213:	bne \$sl, \$s0, loop	#if (i!=n) jump to loop
	mul \$18,\$18,\$16	4194344	0x7250900214:	mul \$s2, \$s2, \$s0	<pre>#else fact = fact * n because the loop function</pre>
2	j 4194364		0x0810000f 15:	j exit	
	mul \$18,\$18,\$17	4194352	0x72519002 18:	mul \$s2, \$s2, \$s1	# fact = fact * i
	addi \$17,\$17,1	4194356	0x22310001 19:	addi \$sl, \$sl, l	#i=i+1
2	j 4194340	4194360	0x08100009 20:	j check	
2	lui \$1,4097	4194364	0x3c011001 23:	addi \$t0, \$0, 268501024	#t0 = address where the result will be stored
	ori \$1,\$1,32	4194368	0x34210020		
9	add \$8,\$0,\$1		0x00014020		
22	sw \$18,0(\$8)		0xad120000 24:	sw \$s2, 0(\$t0)	#M[t0] = s2
100	addi \$2,\$0,10		0x2002000a 25:	addi \$v0,\$0,10	# return 0
	syscall	4194384	0x0000000c 26:	syscall	

• DATA SEGMENT:

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28
268500992	10	0	70100 (10)	(122)	10,00 (120)	74445 (125)	0	Value (120
268501024	3628800	0	0)	0	0	
268501056	0	0	0			0	0	
268501088	0	0	0			0	0	
268501120	0	0	0	0	0	0	0	
268501152	0	0	0	(0	0	0	
268501184	0	0	0	(0	0	0	
268501216	0	0	0	(0	0	0	
268501248	0	0	0		0	0	0	
268501280	0	0	0	(0	0	0	
268501312	0	0	0		0	0	0	
268501344	0	0	0	0	0	0	0	
268501376	0	0	0	(0	0	0	
268501408	0	0	0	(0	0	0	
268501440	0	0	0	(0	0	0	
268501472	0	0	0	0	0	0	0	

• MEMORY:

Name	Number	Value
\$zero	0	G
\$at	1	268501024
\$v0	2	10
\$v1	3	G
\$a0	4	0
\$al	5	G
\$a2	6	C
\$a3	7	G
\$t0	8	268501024
\$t1	9	268500992
\$t2	10	G
\$t3	11	G
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	
\$s0	16	10
\$sl	17	10
\$s2	18	3628800
\$s3	19	G
\$s4	20	0
\$s5	21	0
\$s6	22	G
\$s7	23	0
\$t8	24	G
\$t9	25	G
\$k0	26	0
\$k1	27	G
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	G
pc		4194388

SELECTION SORT

INTRODUCTION:

Selection sort is a sorting algorithm that works by repeatedly selecting the smallest (or largest) element from the unsorted portion of the array and moving it to the sorted portion of the array.

Our program sorts the array using the selection sort method.

PROGRAM EXPLANATION:

Our program has 5 parts: main, outer_loop, cont_out, inner_loop, cont_in and exit.

main: Stores the necessary values, the integer 4 for alignment, the size of the array, the base address of the array and the iterator for the outer_loop which is \$s0.

outer_loop: Checks if you have iterated till the end of the loop. If yes, then branches to exit, else it stores the index of the minimum value obtained through inner_loop in \$s7. It also increments the iterator for the inner_loop which is \$s2.

cont_out: This swaps the minimum element to the desired place in the array (which is the outer loop iterator).

inner_loop: It calculates the next minimum element and then goes to cont_out which then swaps it. **cont_in:** Increments the inner iterator which is \$s2 and proceeds to inner loop.

exit: Exits from the program.

Instructions Used:

- Load Word (lw) Store Word (sw) Add immediate (addi) Multiply (mul) Add (add)
- Branch on greater than equal (bge)Jump (j)

ASSEMBLY CODE:

```
1 .data
2
           arr: .word 23,31,-9,180,99
3 .text
4
5 addi $s5,$zero,4
                                             #store value of 4 to be used later for alignment
                                             #store size of array in $t0
6 addi $t0,$0,5
                                            #store base address of array
7
   la $t1,arr
8
9 #selection sorting code
                                           #counter for outer loop (i=0)
               add $s0,$zero,$zero
10
               addi $s1,$t0,-1
                                            # store len-1 in $s1
11
                                           #check if i>=len-1 then break
12 outer_loop: bge $s0,$s1, exit
                                           #min_pos=i
13
               add $s7,$s0,$zero
                                           #counter for inner loop , j=i+1
14
               addi $s2,$s0,1
               j inner_loop
15
16
               #swap min-pos to place
17
18 cont_out : mul $t3,$s5,$s0
                                            #i*4 for alignment
                                            #store value of index i
               add $t3,$t3,$t1
19
               lw $t4,0($t3)
                                            #tmp= arr[i]
20
21
22
               mul $t6,$s5,$s7
                                            #min pos*4 for alignment
                                           #store min pos address val
23
               add $t6,$t6,$t1
               lw $t5,0($t6)
                                           #tmp2= arr[min_pos]
24
25
               sw $t5,0($t3)
                                           #arr[i] = tmp2 = arr[min pos]
26
27
               sw $t4,0($t6)
                                           #arr[min_pos] = tmp = arr[i]
28
                                           #1++
29
               addi $s0,$s0,1
30
               j outer_loop
31
32 inner_loop: bge $s2,$t0,cont_out
                                           #j>=n , break
33
                                            #j*4 for alignment
34
               mul $t3,$s5, $s2
35
               add $t3,$t3,$t1
                                            #store index j address
               lw $t4,0($t3)
                                            #tmp= arr[j]
36
37
                                            #min_pos*4 for alignment
               mul $t6,$s5,$s7
38
39
               add $t6,$t6,$t1
                                            #store min pos address val
40
               lw $t5,0($t6)
                                           #tmp2= arr[min_pos]
41
                                          #if(arr[j]>=arr[min_pos], no change
               bge $t4,$t5,cont in
42
43
               add $s7,$s2,$zero
                                                #min pos=j;
44
  cont_in:
45
               addi $s2,$s2,1
                                            #1++
46
               j inner_loop
47
48 exit:
               addi $v0,$0,10
                                             #exit program
49
               syscall
50
```

TEXT SEGMENT:

ot Basic	Address	Code			Source
addi \$21,\$0,4	4194304	0x20150004 5:	addi \$s5,\$z	ero,4	#store value of 4 to be used later for alig
addi \$8,\$0,5	4194308	0x20080005 6:	addi \$t0,\$0	,5	#store size of array in \$t0
lui \$1,4097	4194312	0x3c0l100l 7:	la \$tl,arr		#store base address of array
ori \$9,\$1,0	4194316	0x34290000			***
add \$16,\$0,\$0	4194320	0x0000802010:		add \$s0,\$zero,\$zero	#counter for outer loop (i=0)
addi \$17,\$8,-1	4194324	0x2111fffff 11:		addi \$s1,\$t0,-1	# store len-l in \$sl
slt \$1,\$16,\$17	4194328	0x0211082a 12:	outer loop:	bge \$s0,\$sl, exit	#check if i>=len-1 then break
beq \$1,\$0,26	4194332	0x1020001a	1984 1188	10 Total	
add \$23,\$16,\$0	4194336	0x0200b82013:		add \$s7,\$s0,\$zero	#min_pos=i
addi \$18,\$16,1	4194340	0x22120001 14:		addi \$s2,\$s0,1	#counter for inner loop , j=i+l
j 4194388	4194344	0x08100015 15:		j inner_loop	
mul \$11,\$21,\$16	4194348	0x72b0580218:	cont out :	mul \$t3,\$s5,\$s0	#i*4 for alignment
add \$11,\$11,\$9	4194352	0x0169582019:	3000	add \$t3,\$t3,\$t1	#store value of index i
lw \$12,0(\$11)	4194356	0x8d6c0000 20:		lw \$t4,0(\$t3)	#tmp= arr[i]
mul \$14,\$21,\$23	4194360	0x72b77002 22:		mul \$t6,\$s5,\$s7	#min pos*4 for alignment
add \$14,\$14,\$9	4194364	0x01c97020 23:		add \$t6,\$t6,\$t1	#store min pos address val
lw \$13,0(\$14)	4194368	0x8dcd0000 24:		lw \$t5,0(\$t6)	#tmp2= arr[min_pos]
sw \$13,0(\$11)	4194372	0xad6d0000 26:		sw \$t5,0(\$t3)	<pre>#arr[i] = tmp2 = arr[min_pos]</pre>
sw \$12,0(\$14)	4194376	0xadcc0000 27:		sw \$t4,0(\$t6)	#arr[min_pos] = tmp = arr[i]
addi \$16,\$16,1	4194380	0x22100001 29:		addi \$s0,\$s0,1	#1++
j 4194328	4194384	0x08100006 30:		j outer_loop	
slt \$1,\$18,\$8	4194388	0x0248082a 32:	inner_loop:	bge \$s2,\$t0,cont_out	#j>=n , break
beq \$1,\$0,-12	4194392	0x1020fff4			
mul \$11,\$21,\$18	4194396	0x72b25802 34:		mul \$t3,\$s5, \$s2	#j*4 for alignment
add \$11,\$11,\$9	4194400	0x01695820 35:		add \$t3,\$t3,\$t1	#store index j address
lw \$12,0(\$11)	4194404	0x8d6c0000 36:		lw \$t4,0(\$t3)	#tmp= arr[j]
mul \$14,\$21,\$23	4194408	0x72b77002 38:		mul \$t6,\$s5,\$s7	#min_pos*4 for alignment
add \$14,\$14,\$9	4194412	0x01c97020 39:		add \$t6,\$t6,\$t1	#store min_pos address val
lw \$13,0(\$14)	4194416	0x8dcd0000 40:		lw \$t5,0(\$t6)	#tmp2= arr[min_pos]
slt \$1,\$12,\$13	4194420	0x018d082a 42:		bge \$t4,\$t5,cont_in	#if(arr[j]>=arr[min_pos], no change
beq \$1,\$0,1	4194424	0x10200001			
add \$23,\$18,\$0	4194428	0x0240b820 43:		add \$s7,\$s2,\$zero	#min_pos=j;
addi \$18,\$18,1	4194432	0x22520001 45:	cont_in:	addi \$s2,\$s2,1	#j++
j 4194388	4194436	0x08100015 46:	1000	j inner_loop	84.0
addi \$2,\$0,10	4194440	0x2002000a 48:	exit:	addi \$v0,\$0,10	#exit program
syscall	4194444	0x0000000c 49:		syscall	- 20 AN

• DATA SEGMENT:

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+2
268500992	-9	23	31	99	180	0	0	
268501024	0	0	0	0	0	0	0	
268501056	G	0	0	G	0	0	0	
268501088	0	0	0	0	0	0	0	
268501120	0	0	Θ	0	0	0	0	
268501152	0	0	0	0	0	0	0	
268501184	9	0	0	G	0	0	0	
268501216	0	0	0	0	0	0	0	
268501248	9	0	0	G	0	0	0	
268501280	0	0	0	0	0	0	0	
268501312	0	0	0	0	0	0	0	
268501344	0	0	0	0	0	0	0	
268501376	9	0	0	9	0	0	0	
268501408	0	0	0	0	0	0	0	
268501440	0	0	0	0	0	0	0	
268501472	0	0	0	0	0	0	0	

• MEMORY:

Name	Number	Value
\$zero	0	G
\$at	1	G
\$v0	2	10
\$v1	3	G
\$a0	4	(
\$al	5	(
\$a2	6	(
\$a3	7	(
\$t0	8	5
\$t1	9	268500992
\$t2	10	C
\$t3	11	268501004
\$t4	12	180
\$t5	13	99
\$t6	14	268501008
\$t7	15	(
\$s0	16	4
\$sl	17	4
\$s2	18	5
\$s3	19	(
\$s4	20	(
\$s5	21	4
\$s6	22	(
\$s7	23	4
\$t8	24	(
\$t9	25	(
\$k0	26	(
\$k1	27	(
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	(
\$ra	31	(
рс		4194448

BINARY SEARCH

• INTRODUCTION:

Binary Search is a searching algorithm which is used in a sorted array by repeatedly dividing the search interval in half.

Our program searches for a particular number in an array of integers using binary search. If the number exists, the result is the corresponding index in the array; if not, the result is -1.

PROGRAM EXPLANATION:

In this program, we have six parts: main, loop, else, found, not found and exit.

main: Stores the necessary values, i.e., size of the array, the number to be searched (key), base address of array and the length of the array. We also declare lb and ub which is essential to perform the binary search. (NOTE: the boundaries of the array that we are working with at one point is defined by lower-bound and upper-bound).

loop: Checks if the lower-bound (lb) has exceeded upper-bound(ub) or not. If it has, our search is over. If not, it calculates the midpoint of the part of the array we are currently working with and stores the address of it in \$t4. If the element at index midpoint is equal to our key, we have found the value and we branch to found. Else if the element is less than key, we update upperbound to be (midpoint-1). Otherwise, we branch to the else label.

else: It changes the lower bound to (mid + 1) and goes back to loop for the next iteration.

found: Stores the index at which the element is found.

not found: If the number is not there in the array, it stores the result as -1. **exit:**

Exits from the program.

Instructions Used:

- Load Word (lw) Store
 Word (sw) Load address
 (la) Add immediate (addi)
 Add (add)
- Shift right logical (srl)
- Shift left logical (sll)
 Branch on equal (beq)
 Branch on less than (blt)
 Branch on greater than (bgt)
 Jump (j)

ASSEMBLY CODE:

```
#binary search
 2
    .data
 3
             arr: .word 11, 20, 34, 45, 56 #sorted array
 4
     .text
 5
 6
    addi $s0,$zero,5
                                    #store size of array in $s0
    addi $s1,$zero,45
                                    #number to be found
 7
    la $t1,arr
                                    #store base address of array
 8
 9
    addi $t7,$zero,0
                                    #lowerbound=0
 10
    addi $t6,$s0,-1
                                    #upperbound=size-1
11
12
13
    loop:
             bgt $t7,$t6,not_found #if lowerbound>upperbound, search is over
14
             add $t5,$t7,$t6
                                    #mid = lb+ub
                                    #mid = lb+ub/2
15
             srl $t5,$t5,1
             sll $t4,$t5,2
                                    #mid*4 for alignment
16
             add $t4,$t4,$t1
                                    #store address of arr[mid]
17
18
             lw $t3, 0($t4)
                                    #$t3 = arr[mid]
19
             beq $t3,$s1, found
                                    #arr[mid] = key
20
             blt $t3,$s1,else
                                    #if arr[mid]<key
                                    #ub= mid-1
21
             addi $t6,$t5,-1
22
             j loop
23
24
    else:
            addi $t7,$t5,1
                                    #lb=mid+1
25
             j loop
26
27
     found: addi $t0,$0,268501024 #t0 = address where the result will be stored
                                    \#M[t0] = t5 = index of element found
28
             sw $t5,0($t0)
29
             j exit
 30
31
    not found:
32
             addi $t5,$zero,-1
                                    #to store -1 if no. not found
33
             addi $t0,$0,268501024 #t0 = address where the result will be stored
34
             sw $t5,0($t0)
                                    \#M[t0] = t5
35
36
    exit:
37
             addi $v0,$0,10
                                    #exit program
38
             syscall
```

TEXT SEGMENT:

pt	Address	Code	Basic				Source	
	4194304	0x20100005	ddi \$16,\$0,5	6:	addi \$s	0,\$zero,5	#store size of array in \$s0	
			ddi \$17,\$0,45			1,\$zero,45	#number to be found	
		0x3c0110017		8:	la \$t1.	err	#store base address of array	
	4194316	0x34290000	ri \$9.\$1.0				**************************************	
			ddi \$15,\$0,0	10:	addi \$t	7,\$zero,0	#lowerbound=0	
			ddi \$14,\$16,-1			3,\$s0,-1	#upperbound=size-1	
			lt \$1,\$14,\$15				#if lowerbound>upperbound, search is over	
1		0x14200011 b				-0-1		
			dd \$13,\$15,\$14	14:		add \$t5,\$t7,\$t6	#mid = lb+ub	
			rl \$13,\$13,1	15:		srl \$t5,\$t5,1	#mid = \b+ub/2	
	4194344	0x000d6080 s	11 \$12,\$13,2	16:		sll \$t4,\$t5,2	#mid*4 for alignment	
			dd \$12,\$12,\$9	17:		add \$t4,\$t4,\$t1	#store address of arr[mid]	
	4194352	0x8d8b00001	w \$11,0(\$12)	18:		lw \$t3, 0(\$t4)	#\$t3 = arr[mid]	
			eq \$11,\$17,6	19:		beg \$t3,\$sl, found	#arr[mid] = key	
			lt \$1.\$11.\$17	20:		blt \$t3,\$sl,else	#if arr[mid] <key< td=""><td></td></key<>	
	4194364	0x14200002 b	ne \$1,\$0,2			,		
	4194368	0x2laeffffa	ddi \$14.\$131	21:		addi \$t6,\$t5,-1	#ub= mid-1	
	4194372	0x08100006	4194328	22:		i loop	300/10003000	
	4194376	0x21af0001	ddi \$15.\$13.1	24:	else:	addi \$t7,\$t5.1	#lb=mid+l	
	4194380	0x08100006	4194328	25:		j loop		
		0x3c011001		27:	found:		#t0 = address where the result will be stored	
	4194388	0x34210020 d	ri \$1.\$1.32			* ***		
	4194392	0x00014020 a	dd \$8,\$0,\$1					
	4194396	0xad0d00000	w \$13,0(\$8)	28:		sw \$t5,0(\$t0)	#M(t0) = t5 = index of element found	
	4194400	0x0810001e	4194424	29:		j exit		
			ddi \$13,\$0,-1	32:		addi \$t5,\$zero,-1	#to store -1 if no. not found	
		0x3c011001		33:			#t0 = address where the result will be stored	
	4194412	0x34210020 d	ri \$1,\$1,32			100 00000		
		0x00014020 a						
	4194420	0xad0d00000	w \$13,0(\$8)	34:		sw \$t5,0(\$t0)	#M[t0] = t5	
			ddi \$2,\$0,10	37:		addi \$v0.\$0.10	#exit program	
		0x0000000c		38:		syscall	Supplemental Desire Monage	

• DATA SEGMENT:

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28
268500992	11	20	34	45	56	0	0	
268501024	3	0	0	0	9	0	0	
268501056	0	0	0	0	9	0	0	
268501088	0	0	0	0	0	0	0	
268501120	0	0	0	0	0	0	0	
268501152	0	0	0	0	0	0	0	
268501184	0	0	0	0	0	0	0	
268501216	0	0	0	0	9	0	0	
268501248	0	0	0	0	0	0	0	
268501.280	0	0	0	0	0	0	0	
268501312	0	0	0	0	0	0	0	
268501344	0	0	0	0	0	0	0	
268501376	0	Θ	Θ	0	9	0	0	
268501408	0	0	0	0	O	0	0	
268501440	0	0	0	0	9	0	0	
268501472	0	0	0	0	0	0	0	

• MEMORY:

Registers	Coproc 1	Coproc 0		_	
Nam	е	Number		Value	
\$zero			0	0	
\$at	11		1	268501024	
\$v0			2	10	
\$v1	11		3	O	
\$a0			4	0	
\$al	11		5	0	
\$a2			6	0	
\$a3	11		7	O	
\$t0			8	268501024	
\$t1	11		9	268500992	
\$t2			10	0	
\$t3	11		11	45	
\$t4			12	268501004	
\$t5			13	3	
\$t6			14	4	
\$t7			15	3 5	
\$s0			16	5	
\$sl	11		17	45	
\$s2			18	0	
\$s3	11		19	0	
\$s4			20	O	
\$s5	13		21	0	
\$s6			22	O	
\$s7	13		23	0	
\$t8			24	0	
\$t9	13		25	0	
\$k0			26	O	
\$k1			27	0	
\$gp			28	268468224	
\$sp			29	2147479548	
\$fp			30	0	
\$ra	14		31	0	
рс				4194432	

ASSUMPTIONS IN PROCESSOR:

- 1) We have defined the ALUOp of the bne instruction as 11.
- 2) We have defined the ALU control input code of srl as '101' and sll as '110'.

RESULT (FROM PROCESSOR)

FACTORIAL

Teorigia/Pion - Teorigia/Pion / Colombia: Section Colombia: Sect
Register file: [0: 0, 1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 10, 9: 0, 10: 0, 11: 0, 12: 0, 13: 0, 14: 0, 15: 0, 16: 0, 17: 0, 18: 0, 19: 0, 20: 0, 21: 0, 22: 0, 23: 0, 26: 0, 27: 0, 28: 268468224, 29: 2147479548, 30: 0, 31: 0]
Data Remory: (266569992: 0, 265501024: 0)
PC: 4394366
Register file: (0: 0, 1: 265500992, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 10, 9: 0, 10: 0, 11: 0, 12: 0, 13: 0, 14: 0, 15: 0, 16: 0, 17: 0, 18: 0, 19: 0, 20: 0, 21: 0, 22: 0, 23: 0, 24: 0, 25: 0, 26: 0, 27: 0, 28: 268468224, 29: 2147479548, 30: 0, 31: 0}
Outa Minory: {268551024: 0}
PC: 4180112
Register file: {0: 0, 1: 260500992, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 10, 9: 0, 10: 0, 11: 0, 12: 0, 13: 0, 14: 0, 15: 0, 16: 0, 17: 0, 18: 0, 19: 0, 20: 0, 21: 0, 22: 0, 21: 0, 24: 0, 25: 0, 26: 0, 27: 0, 28: 26806224, 29: 2147479548, 30: 0, 31: 0}
Data Minory: (2005/01/02): 0, 2005/01/02: 0)
K: 4194316
Register (File: (6) 6, 1: 268669992, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 10, 9: 26869992, 10: 0, 11: 0, 12: 0, 13: 0, 14: 0, 15: 0, 16: 0, 17: 0, 18: 0, 19: 0, 20: 0, 21: 0, 22: 0, 23: 0, 20: 0, 27: 0, 28: 26046220, 20: 2107779040, 30: 0, 31: 0)
Data Meaory: { 1266569992: 0, 266501624: 0}
PC: 4194330
Register file: {0: 0, 1: 266560992, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 10, 9: 26656992, 10: 0, 11: 0, 12: 0, 13: 0, 14: 0, 15: 0, 16: 0, 17: 0, 18: 0, 19: 0, 20: 0, 21: 0, 22: 0, 21: 0, 22: 0, 23: 0, 26: 0, 27: 0, 28: 266460224, 29: 2147479548, 30: 0, 31: 0}
Data Monory: { {cnsserver: 10, 2nsseries+: 0}
K: 1993N
Register File: (0: 0, 1: 268500992, 2: 0, 5: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 10, 9: 26850992, 10: 0, 11: 0, 12: 0, 13: 0, 14: 0, 15: 0, 16: 10, 17: 0, 16: 0, 19: 0, 20: 0, 21: 0, 22: 0, 23: 0, 26: 0, 27: 0, 28: 26860224, 29: 2187479540, 30: 0, 31: 0}
Data Memory: {266501624: 0}
K: 439428
Register file: (0: 0, 1: 2685869972, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 10, 9: 268586997, 10: 0, 11: 0, 12: 0, 13: 0, 14: 0, 15: 0, 16: 10, 17: 1, 18: 0, 19: 0, 20: 0, 21: 0, 22: 0, 23: 0, 24: 0, 25: 0, 26: 0, 27: 0, 28: 268468724, 29: 2147479548, 30: 0, 31: 0]
Data Resory: {268506992: 18, 268501824: 0}
PC: 4194332
Register file: (6: 6, 1: 288580992, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 10, 9: 268580992, 10: 0, 11: 0, 12: 0, 13: 0, 14: 0, 15: 0, 16: 10, 17: 1, 18: 1, 19: 0, 20: 0, 21: 0, 22: 0, 23: 0, 24: 0, 25: 0, 26: 0, 27: 0, 28: 269468224, 29: 2147479948, 30: 0, 31: 0)
Data Memory: {2665899992: 18, 268591024: 0}
pi: 4194336
Register file: (6: 6, 1: 266586992, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 0: 10, 9: 26656992, 10: 0, 11: 0, 12: 0, 13: 0, 14: 0, 15: 0, 16: 10, 17: 1, 10: 1, 19: 0, 20: 0, 21: 0, 22: 0, 23: 0, 24: 0, 25: 0, 26: 0, 27: 0, 20: 260468224, 29: 2107079548, 30: 0, 31: 0)
Tata Rescry: {208598992: 18, 288501024: 0}
IC: 4194349
Register file: (61:4) 1-12 (14
Data Recorp: [266569992: 18, 266501024: 0]
PC: 4194052
Register file:
(6: 0, 1: 285500992, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 10, 9: 265500992, 10: 0, 11: 0, 12: 0, 13: 0, 14: 0, 15: 0, 16: 10, 17: 1, 18: 1, 19: 0, 20: 0, 21: 0, 22: 0, 23: 0, 24: 0, 25: 0, 27: 0, 28: 269468224, 29: 2147479548, 30: 0, 31: 0) Data Rescry: (266500992: 19, 265501024: 0)
EC: 4194356
(6: e, 1:266569992, 2: e, 3: e, 4: e, 5: e, 6: e, 7: e, 8: 10, 9: 266569992, 10: e, 11: e, 12: e, 13: e, 14: e, 15: e, 16: 16, 17: 2, 18: 1, 19: e, 20: e, 21: e, 22: e, 23: e, 24: e, 25: e, 26: e, 27: e, 28: 268469224, 29: 2187079548, 30: e, 31: e) Data Reservey: (6: e, 1: 265669992, 12: e, 21: e, 22: e, 23: e, 24: e, 25: e, 26: e, 27: e, 28: 26
(266566922: 18, 266501024: 0)

```
92, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 6: 10, 9: 26896992, 10: 0, 11: 0, 12: 0, 13: 0, 14: 0, 15: 0, 16: 10, 17: 2, 18: 1, 19: 0, 20: 0, 21: 0, 22: 0, 24: 0, 25: 0, 24: 0, 25: 0, 26: 0, 27: 0, 28: 269468224, 29: 21:
   ata Memory:
{268500992: 10, 268501024: 0}
  Nepster file: (#0 6) 1: 8: 85: 86: 87: 6: 87: 6: 87: 6: 87: 88: 10, 92: 2885600972, 106: 0, 112: 0, 122: 0, 123: 0, 194: 0, 122: 5885600972, 2: 0, 23: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 24: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25: 0, 25
 PC: 4194352
  Register file: (6: 6, 1: 2-86000002, 2: 6, 3: 6, 4: 6, 5: 6, 6: 8, 7: 6, 8: 10, 6: 8, 7: 6, 8: 10, 9: 268560002, 10: 6, 11: 0, 12: 6, 13: 6, 14: 6, 15: 0, 16: 10, 17: 2, 18: 2, 19: 6, 20: 6, 21: 6, 22: 6, 23: 6, 23: 6, 26: 6, 27: 6, 28: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 26, 20: 2
 Register file: (6: 6; 1: 6: 86000002, 2: 6, 3: 9, 6: 6, 5: 6, 6: 8, 7: 6, 8: 10, 9: 2685000002, 10: 0, 11: 0, 12: 0, 13: 0, 16: 10, 17: 3, 18: 2, 19: 0, 20: 0, 21: 0, 20: 0, 21: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20: 0, 20:
  Data Monory:
{268500992: 10, 268501024: 0}
  Hepster file: (00, 0, 1: 6, 3: 6, 4: 6, 5: 0, 6: 0, 7: 0, 8: 10, 9: 268566992, 10: 0, 11: 0, 12: 0, 13: 6, 14: 0, 15: 0, 16: 10, 17: 3, 18: 2, 17: 0, 20: 0, 21: 0, 22: 0, 23: 0, 24: 0, 25: 0, 26: 0, 27: 0, 28: 26846224, 20: 2107479548, 30: 0, 31: 0)
Register file: (6: 6, 1: 3.64699992, 2: 6, 3: 0, 4: 0, 5: 6, 6: 0, 7: 0, 8: 10, 9: 265569992, 10: 8, 11: 9, 12: 0, 13: 0, 14: 0, 15: 0, 16: 10, 17: 0, 15: 6, 19: 0, 20: 0, 21: 0, 22: 0, 23: 0, 20: 0, 27: 0, 20: 269468220, 29: 22477799103, 30: 0, 31: 0)
 Hepister (Cit):
  PC: 4194352
 Register file: [6] (8, 12-60569997, 2: 6, 3: 6, 4: 6, 5: 6, 6: 6, 7: 6, 8: 16, 9: 26656997, 16: 6, 11: 6, 12: 8, 13: 9, 14: 9, 15: 9, 16: 10, 17: 4, 18: 24, 19: 6, 20: 9, 21: 0, 22: 0, 26: 0, 27: 0, 26: 0, 26: 0, 27: 0, 28: 26666224, 29: 2170779548, 39: 6, 31: 0)
  PC: 4194356
 Register (1/16):
 Oata Memory:
{268500992: 10, 268501024: 0}
 egistor file:
{e: e, 1: 268560992, 2: e, 3: e, 4: 0, 5: e, 6: e, 7: e, 8: 10, 9: 268560992, 10: e, 11: e, 12: e, 13: e, 14: e, 15: e, 16: 10, 17: 6, 18: 120, 19: e, 20: e, 21: e, 22: e, 23: e, 24: e, 25: e, 26: e, 27: e, 28: e
  negistor file: [19. 4]: 6, 18. 4; 6, 5; 8, 6; 6, 7; 6, 8; 18, 9; 268560997, 16; 6, 11; 6, 12; 6, 11; 6, 12; 6, 15; 6, 16; 19, 17; 6, 16; 19, 19; 6, 26; 6, 21; 6, 22; 6, 21; 6, 22; 6, 21; 6, 24; 6, 25; 6, 26; 6, 27; 6, 28; 26866224, 29; 21479
  egister file: (0° 6, 1: 6, 4: 0, 5: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 10, 9: 268500992, 18: 0, 11: 0, 12: 0, 13: 0, 14: 0, 15: 0, 16: 10, 17: 0, 18: 120, 19: 0, 20: 0, 21: 0, 22: 0, 23: 0, 24: 0, 25: 0, 26: 0, 27: 0, 28: 26
```

```
ata Memory:
{268500992: 10, 268501024: 0}
  Data Memory:
{268588992: 10, 268581824: 8}
 PC: 4194360
 PC: 4194352
  Negister file: [80, 51: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61: 80, 61
  Data Memory:
{268500992: 10, 268501024: 0}
 Register file:
 Register file: (6: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1: 9. 1:
 Supplier file:
(0) 6, 1: 364669992, 2; 6, 3; 6, 4; 6, 5; 6, 6; 0, 7; 6, 8; 10, 9; 264669992, 10; 6, 11; 6, 12; 0, 13; 6, 14; 0, 15; 6, 16; 16, 17; 6, 18; 40326, 19; 0, 20; 0, 21; 6, 22; 0, 23; 6, 24; 6, 25; 0, 26; 6, 27; 6, 28; 26466224, 29; 21/07099688, 20; 0, 21; 6)
    lata Memory:
{268500992: 10, 268501024: 0}
 Register (fals: (00 9, 11: 286800002), 21: 6, 31: 9, 41: 6, 51: 6, 61: 0, 77: 0, 81: 13, 97: 268800002, 107: 0, 81: 13, 91: 268800002, 107: 0, 81: 13, 91: 268800002, 107: 0, 81: 201: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107: 0, 107:
 Negister (file: (dic: 4, 2, 3; 6, 4; 6, 5; 6, 6; 6, 7; 6, 8; 10, 7; 6, 8; 10, 9; 265500002, 10: 6, 11: 6, 12: 0, 13: 6, 14: 0, 15: 0, 16: 10, 17: 9, 18: 40320, 19: 0, 20: 0, 21: 0, 22: 0, 23: 0, 24: 0, 26: 0, 27: 0, 28: 268468224, 29: 214
 PC: 4194344
Register (file: (6), 4, 12 (20099992, 2: 6, 2: 6, 2: 7: 6, 4: 6; 5: 6, 6: 6, 7: 6, 8: 19, 6: 2009992, 16: 6, 11: 0, 12: 0, 13: 0, 14: 6, 15: 6, 16: 10, 17: 10, 15: 0, 16: 10, 17: 10, 15: 0, 16: 10, 17: 10, 15: 0, 16: 10, 17: 10, 15: 10, 16: 10, 17: 10, 16: 10, 17: 10, 16: 10, 17: 10, 16: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 17: 10, 1
  PC: 4194348
```

RC: 4151186
Register file: (8: 0, 1: 266500992, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 10, 9: 266500992, 10: 0, 11: 0, 12: 0, 13: 0, 14: 0, 15: 0, 16: 10, 17: 10, 18: 362800, 19: 0, 20: 0, 21: 0, 22: 0, 21: 0, 20: 0, 26: 0, 27: 0, 28: 26660224, 29: 2107079540, 30: 0, 31: 0)
Data Mesapy: (588569992: 10, 268591024: 0)
PC: 113/1068
Register file: [8: 0, 1: 268901024, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 10, 9: 26890992, 10: 0, 11: 0, 12: 0, 13: 0, 14: 0, 15: 0, 16: 10, 17: 10, 18: 362800, 19: 0, 20: 0, 21: 0, 22: 0, 21: 0, 28: 0, 26: 0, 27: 0, 28: 2686024, 29: 214979948, 30: 0, 13: 0]
Data Mesery: (36866992: 10, 268691024: 0)
PC: 4394972
Register file: {6: 0, 1: 268501624, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 268501824, 9: 268501824, 9: 268501822, 10: 0, 11: 0, 12: 0, 13: 0, 14: 0, 15: 0, 16: 10, 17: 10, 18: 3638800, 19: 0, 20: 0, 21: 0, 22: 0, 24: 0, 25: 0, 26: 0, 27: 0, 28: 268468224, 29: 2147479548, 30: 0, 31: 0)
Data Mesery: (288686992: 10, 28861024: 0)
PC: 4194776
Register file: (6: 0, 1: 0.0001022, 1: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 265001224, 9: 265000022, 10: 0, 11: 0, 12: 0, 13: 0, 14: 0, 15: 0, 16: 15, 17: 15, 15: 0.22000, 19: 0, 20: 0, 21: 0, 21: 0, 21: 0, 25: 0, 26: 0, 27: 0, 31: 26000224, 39: 2147179508, 30: 0, 31: 0)
Data Memory: (248000992: 10, 2480618294: 3423880)
DC: 139/388
Register file: [8: 0, 1: 268501024, 2: 10, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 0: 268501024, 9: 268500922, 10: 0, 11: 0, 12: 0, 13: 0, 14: 0, 15: 0, 16: 10, 17: 10, 10: 3628000, 19: 0, 20: 0, 21: 0, 21: 0, 21: 0, 25: 0, 26: 0, 27: 0, 20: 26840224, 29: 2147079588, 30: 0, 31: 0]
Data Mesory: (268000992: 10, 268501024: 362860)
DC: 1139188
Register file: [0: 0, 1: 268501024, 2: 10, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 6: 268501024, 9: 268500902, 10: 0, 11: 0, 12: 0, 13: 0, 14: 0, 15: 0, 16: 10, 17: 10, 18: 3628000, 19: 0, 20: 0, 21: 0, 22: 0, 24: 0, 25: 0, 26: 0, 27: 0, 28: 26840224, 29: 2147479548, 30: 0, 31: 0}
Data Remory: (268069992: 18, 2680618294: 3628889)
Hemilt: (265806992: 10, 265801824: 362800)

BINARY SEARCH

```
equatur (16/4)
(6 % 5 : 0 % 3 : 0 % 3 : 0 % 10 : 0 5 : 0 % 6 : 0 7 : 0 % 6 : 0 7 : 0 % 6 : 0 % 2000000002, 20 : 0 % 12 : 0 % 12 : 0 % 12 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 : 0 % 15 :
  ata Hemory:
{268500902: 11, 268500906: 20, 268501000: 34, 268501004: 45, 268501008: 56, 268501024: 0}
  reporter false. 10: 6, 2: 6, 3: 6, 4: 6, 5: 6, 6: 6, 7: 6, 8: 8, 9: 200500092, 10: 6, 11: 6, 12: 6, 12: 4, 14: 4, 15: 6, 16: 5, 17: 45, 16: 6, 19: 6, 20: 6, 21: 6, 22: 6, 23: 6, 24: 6, 25: 6, 26: 6, 27: 6, 28: 200400214, 28: 2147475048, 30: 6, 31: 61
  EQ. CENTROL :-

10 T. A. C. P. C. A. C. P. C. E. P. C. E. P. C. P. E. P. C. P. E. P. C. P.
  ata Memory:
{268500072: 11, 268500076: 20, 268501000: 34, 268501004: 45, 268501008: 56, 268501024: 0}
PC: 4194344
PC: 4194348
   Register file: (6: 6, 1: 6, 2: 6, 3: 6, 4: 6, 5: 9, 6: 6, 7: 6, 8: 6, 9: 268560992, 10: 0, 11: 34, 12: 268561090, 13: 2, 14: 4, 15: 6, 16: 5, 17: 45, 18: 6, 19: 6, 20: 6, 21: 0, 21: 0, 21: 0, 21: 0, 21: 0, 21: 0, 27: 6, 26: 6, 27: 6, 28: 26846024, 29: 2142479848, 30: 0, 31: 0)
   egister file: (e. 6, 1: 6, 2: 6, 3: 6, 4: 6, 5: 6, 6: 6, 7: 6, 8: 6, 9: 268566992, 18: 9, 11: 34, 12: 268561990, 13: 2, 14: 4, 15: 6, 16: 5, 17: 45, 18: 6, 19: 0, 20: 0, 21: 0, 22: 0, 23: 0, 24: 0, 25: 6, 26: 6, 27: 6, 28: 268568224, 29: 21*7479548, 30: 0, 31: 0)
 Data Memory:
{268508992: 11, 268508996: 20, 268501000: 34, 268501004: 45, 268501008: 56, 268501024: 0}
 PC: 4194360
  Data Memory:
{268500992: 11, 768500996: 20, 768501080: 34, 768501084: 45, 768501088: 56, 768501024: 0}
Register (falce: (8) e. 8, 12, 2, 12, 3, 15, 4, 41, 6, 51, 6, 61, 67, 8, 81, 6, 92, 264, 200, 202, 101, 0, 111, 34, 121, 266861800, 312, 2, 341, 4, 151, 6, 161, 5, 171, 45, 101, 6, 101, 6, 201, 6, 211, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 221, 6, 2
   egitter (file: [6: 6, 1: 1, 2: 6, 3: 6, 4: 6, 5: 6, 6: 6, 7: 6, 6: 0, 7: 6, 6: 0, 7: 26556992, 16: 0, 11: 34, 12: 268561993, 13: 0, 13: 4, 16: 3, 16: 5, 17: 45, 18: 6, 19: 6, 20: 6, 21: 6, 22: 6, 22: 6, 22: 6, 22: 6, 22: 6, 22: 6, 23: 6, 22: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23: 6, 23
  ata Memory:
{268508992: 11, 268508996: 20, 268501000: 34, 268501004: 45, 268501008: 56, 268501024: 0}
Register file: (9: 0. 1: 1. 2: 0. 3: 0. 4: 0. 5: 0. 6: 0. 7: 0. 8: 0. 9: 268509992. 10: 0. 11: 34. 12: 268501000. 31: 2. 14: 4. 15: 3. 16: 5. 17: 45. 18: 0. 19: 0. 20: 0. 20: 0. 21: 0. 22: 0. 23: 0. 24: 0. 25: 0. 26: 0. 27: 0. 26: 26: 26860224, 29: 2147479548, 30: 0. 31: 0]
  Oata Memory:
{268500992: 11, 268500996: 20, 268501000: 34, 268501004: 45, 268501008: 56, 268501024: 0}
Register file: (6:0, 1:0, 2:0, 3:0, 4:0, 5:0, 6:0, 7:0, 8:0, 9:268500992, 10:0, 11:34, 12:268501000, 13:2, 14:4, 15:3, 16:5, 17:45, 18:0, 19:0, 28:0, 28:0, 21:0, 22:0, 23:0, 24:0, 25:0, 26:0, 27:0, 28:268646224, 29:21897
Data Memory:
{268500992: 11, 268500996: 28, 268501000: 34, 268501004: 45, 268501008: 56, 268501024: 0}
Registre file: (6: 6, 1: 6, 2: 6, 2: 6, 4: 9, 5: 6, 6: 9, 7: 6, 8: 9, 9: 26:500992, 10: 0, 11: 34, 32: 266601000, 13: 7, 14: 4, 15: 3, 16: 6, 17: 45, 18: 6, 19: 6, 20: 0, 21: 0, 22: 0, 21: 0, 22: 0, 24: 0, 25: 0, 26: 0, 27: 0, 20: 26: 0, 27: 0, 20: 266601224, 29: 21474791046, 30
Data Menory:
{768500992: 11, 268500996: 20, 268501000: 34, 268501004: 45, 268501008: 56, 268501024: 0}
 Data Memory:
{268500992: 11, 268500996: 20, 268501000: 34, 268501004: 45, 268501008: 56, 268501024: 0}
PC: 4194348
```

PC: 4194352
Hepister file:
(6) 0, 11 0, 21 0, 31 0,
Data Newsry: (2665689992: 11, 268508996: 20, 268501900: 34, 268501900: 45, 268501902: 6, 268501924: 0)
PC: 4194356
Register file: (8: 0, 1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 0: 0, 9: 268569992, 10: 0, 11: 45, 12: 268561004, 13: 3, 14: 4, 15: 3, 16: 5, 17: 45, 10: 0, 19: 0, 20: 0, 21: 0, 22: 0, 23: 0, 24: 0, 25: 0, 26: 0, 27: 0, 28: 268460224, 29: 2147879548, 30: 0, 31: 0)
Data Resory: (26560999: 11, 26500999: 20, 26501000: 34, 266501000: 56, 266501024: 0)
PC: (159138)
Register file: [8: 0, 1: 268599992, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 0, 9: 268599992, 10: 0, 11: 45, 12: 268591004, 13: 3, 14: 4, 15: 3, 16: 5, 17: 45, 18: 0, 19: 0, 20: 0, 21: 0, 22: 0, 23: 0, 24: 0, 25: 0, 26: 0, 27: 0, 28: 268463224, 29: 2147479548, 30: 0, 31: 0}
Data Resory: (26560990: 28, 26561900: 34, 26561900: 45, 26561900: 66, 26561924: 0)
PC: 4154188
Register file: - (0: 0, 1: 268501024, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 0, 9: 26850002, 10: 0, 11: 45, 12: 268501004, 13: 3, 14: 4, 15: 3, 16: 5, 17: 45, 18: 0, 19: 0, 20: 0, 21: 0, 22: 0, 23: 0, 24: 0, 25: 0, 26: 0, 27: 0, 28: 26946324, 29: 2147479548, 30: 0, 31: 0)
Data Mesory: (265000996; 20, 265001000; 34, 265001000; 45, 266501204; 0)
PC: 4194392
Register file: [0: 0, 1: 268501624, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 268501624, 9: 26850992, 10: 0, 11: 45, 12: 268501684, 13: 3, 14: 4, 15: 3, 16: 5, 17: 45, 18: 0, 19: 0, 20: 0, 21: 0, 22: 0, 22: 0, 20: 0, 26: 0, 27: 0, 28: 26846224, 29: 2149479548, 30: 0, 21: 0]
Data Rewory: (265500990: 28, 265501900: 34, 265501900: 45, 265501900: 56, 265501924: 0)
PC: 4194196
Register file: (6: 0, 1: 268501024, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 268501024, 9: 268501024, 9: 268501024, 13: 3, 14: 4, 15: 3, 16: 5, 17: 45, 18: 0, 19: 0, 20: 0, 21: 0, 22: 0, 23: 0, 24: 0, 25: 0, 26: 0, 27: 0, 28: 268468224, 29: 2147479548, 39: 0, 31: 0)
Outs Monory: (246504996: 28, 265501908: 34, 266501908: 56, 266501908: 56, 266501908: 3}
PC: 1129408
Register file: - (0: 0, 1: 268501024, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 268501024, 9: 268501024, 9: 268501029, 10: 0, 11: 45, 12: 268501099, 13: 3, 14: 4, 15: 3, 16: 5, 17: 45, 18: 0, 19: 0, 20: 0, 21: 0, 22: 0, 23: 0, 24: 0, 25: 0, 26: 0, 27: 0, 28: 268468224, 29: 2147479548, 38: 0, 31: 0
Data Runory: (268506992: 11, 268500996: 20, 268501000: 34, 268501000: 56, 268501024: 3)
PC: 41594124
Register file: - (0: 0, 1: 268501024, 2: 10, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 268501024, 9: 268500024, 9: 2485000024, 9: 2485000024, 9: 2485000024, 9: 2485000024, 9: 2485000024, 9: 2485000024, 9: 2485000024, 9: 2485000024, 9: 2485000024, 9: 2485000024, 9: 2485000024, 9: 2485000024, 9: 24850000024, 9: 248500000024, 9: 2485000000000000000000000000000000000000
Outa Renory: (268506992: 11, 268500996: 20, 268501000: 30, 268501000: 56, 268501020: 3)
OC: 11591128
Hugister file: (6: 0, 1: 268601024, 2: 10, 3: 6, 4: 0, 5: 0, 6: 0, 7: 0, 8: 268601024, 9: 268500992, 10: 0, 11: 45, 12: 268501004, 13: 3, 14: 4, 15: 3, 16: 5, 17: 45, 18: 0, 19: 0, 20: 0, 21: 0, 22: 0, 23: 0, 24: 0, 25: 0, 26: 0, 27: 0, 28: 26806024, 29: 2147479648, 30: 0, 31:
Outa Menory: [266560992: 11, 266560996: 20, 266561006: 50, 266501006: 56, 266561008: 56, 266561008: 3}
Result:

SELECTION SORT

PC: 4194304										
Data Memory: {268500992:	23,	268500996:	31,	268501000:	-9,	268501004	: 180,	268501008	: 99}	
PC: 4194308 Data Memory: {268500992:	23,	268500996:	31,	268501000:	-9,	268501004	: 180,	268501008	: 99}	
PC: 4194312 Data Memory: {268500992:		268500996:		268501000:		268501004	: 180,	268501008	: 99}	
PC: 4194316 Data Memory: {268500992:	23,	268500996:	31,	268501000:	-9,	268501004	: 180,	268501008	: 99}	
PC: 4194320 Data Memory: {268500992:										
PC: 4194324 Data Memory: {268500992:										
PC: 4194328 Data Memory: {268500992:										
PC: 4194332 Data Memory: {268500992:										
PC: 4194336 Data Memory:										
{268500992: PC: 4194340 Data Memory:										
{268500992: PC: 4194344 Data Memory:										
{268500992: 										
{268500992: 										
{268500992: 										
{268500992: PC: 4194400 Data Memory:										
{268500992: PC: 4194404 Data Memory:										
{268500992: PC: 4194408 Data Memory:										
{268500992: PC: 4194412 Data Memory:										
268500992: 										
PC: 4194420 Data Memory: {268500992:										
PC: 4194424 Data Memory: {268500992:										
PC: 4194432 Data Memory: {268500992:										
PC: 4194436 Data Memory: {268500992:										
PC: 4194388 Data Memory: {268500992:										
PC: 4194392 Data Memory: {268500992:										
PC: 4194396 Data Henory: {268500992:	23,	268500996:	31,	268501000:	-9,	268501004:	180,	268591998:	99}	
PC: 4194400 Data Hemory: {268500992:		268500996:		268501000:		268501004:	180,	268501008:	99}	
PC: 4194404 Data Memory: {268500992:		268500996:		268501000:		268591994:	180,	268591998:	99}	
PC: 4194408 Data Memory: {268500992:		268500996:		268501000:		268501004:	180,	268501008:	99}	
PC: 4194412 Data Memory: {268500992:	23,	268500996:	31,	268501000:	-9,	268501004:	180,	268501008:	99}	
PC: 4194416 Data Memory: {268500992:	23,	268500996:	31,	268501000:	-9,	268591094:	180,	268591998:	99}	
PC: 4194420 Data Memory: {268500992:	23,	268500996:	31,	268501000:	-9,	268501004:	180,	268501008:	99}	

```
Data Memory:
{268500992: 23, 268500996: 31, 268501000: -9, 268501004: 180, 268501008: 99}
PC: 4194428
 Data Hemory:
{268500992: 23, 268500996: 31, 268501000: -9, 268501004: 180, 268501008: 99}
PC: 4194432
Data Memory:
{268500992: 23, 268500996: 31, 268501000: -9, 268501004: 180, 268501008: 99}
PC: 4194436
Data Memory:
{268500992: 23, 268500996: 31, 268501000: -9, 268501004: 180, 268501008: 99}
 Data Memory:
{268500992: 23, 268500996: 31, 268501000: -9, 268501004: 180, 268501008: 99}
PC: 4194392
Data Memory:
{2685089992: 23, 268508996: 31, 268501000: -9, 268501004: 180, 268501008: 99}
PC: 4194396
Data Memory:
{268500992: 23, 268500996: 31, 268501000: -9, 268501004: 180, 268501008: 99}
PC: 4194400
Data Memory:
{268500992: 23, 268500996: 31, 268501000: -9, 268501004: 180, 268501008: 99}
PC: 4194404
 Oata Memory:
{268500992: 23, 268500996: 31, 268501000: -9, 268501004: 180, 268501008: 99}
PC: 4194408
Data Hemory:
{268509992: 23, 268500996: 31, 268501000: -9, 268501004: 180, 268501008: 99}
PC: 4194412
Data Memory:
{268500992: 23, 268500996: 31, 268501000: -9, 268501004: 180, 268501008: 99}
PC: 4194416
 PC: 113-110:
Johan Memory:
{268500992: 23, 268500996: 31, 268501000: -9, 268501004: 180, 268501008: 99}
PC: 4194420
Data Memory:
{268500992: 23, 268500996: 31, 268501000: -9, 268501004: 180, 268501008: 99}
PC: 4194424
Data Memory:
{268500992: 23, 268500996: 31, 268501000: -9, 268501004: 180, 268501008: 99}
PC: 4194432
Data Hemory:
{268500992: 23, 268500996: 31, 268501000: -9, 268501004: 180, 268501008: 99}
PC: 4194436
Data Memory:
{268500992: 23, 268500996: 31, 268501000: -9, 268501004: 180, 268501008: 99}
 Data Memory:
{268500992: 23, 268500996: 31, 268501000: -9, 268501004: 180, 268501008: 99}
 PC: 4194360
 PC: 4134300
Data Memory:
{268500992: 23, 268500996: 31, 268501008: -9, 268501004: 180, 268501008: 99}
 PC: 4194364
Data Hemory:
{268500992: 23, 268500996: 31, 268501000: -9, 268501004: 180, 268501008: 99}
 PC: 4194368
 P.: 4134506
Data Henory:
[268500992: 23, 268500996: 31, 268501000: -9, 268501004: 180, 268501008: 99]
  Oata Memory:
{268500992: -9, 268500996: 31, 268501000: -9, 268501004: 180, 268501008: 99}
 PC: 4194376
 Pc: 4124570
Data Henory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
 PC: 4194380
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
 PC: 4194384
  Data Hemory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
 PC: 4194328
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
 PC: 4194332
 Data Memory:
{268509992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
 DC: 4194336
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
 PC: 4194340
 P.: 4134346
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
 PC: 4194344
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
 PC: 4194388
 Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
 PC: 4194392
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
 PC: 4194396
 PC: 133362
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
```

```
с. изоннее
                -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194404
Data Memory:
{268500992:
                -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194412
Data Hemory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194420
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194424
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194428
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194432
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194436
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194388
Data Hemory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
D: 4194392
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194396
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
DC: 4194400
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194404
pc. -1394104
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194412
Data Memory:
[268560992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194416
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194420
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194424
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
Data Memory:
{268500992: -9, 268508996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194436
PC: 1134730
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194388
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194392
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194396
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194400
RC.-11-17-00
Data Hemory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194404
AC: 41294094
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194408
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194412
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194416
PC: 4124410
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194420
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194424
Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
```

PC: 4194432 Data Memory: {268509092: -9, 268509996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194436 Data Memory: {268509092: -9, 268500996: 31, 26850100: 23, 268501004: 180, 268501008: 99}
PC: 4194388 Data Memory:
{2685080992: -9, 2685080996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
{268509992: -9, 26850996: 31, 268501000: 23, 268501004: 180, 268501008: 99} PC: 4194348 Data Menory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99} PC: 4194352 Data Memory:
{268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99} PC: 4194356
Data Memory: {268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99} PC: 4194360
Data Memory: {268508992: -9, 268508996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
Data Memory: {268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194368 Data Memory: {268500992: -9, 268500996: 31, 268501000: 23, 268501004: 180, 268501008: 99}
PC: u194372 Data Memory: {268500992: -9, 268500996: 23, 268501000: 23, 268501004: 180, 268501008: 99}
PC: 4194376 Data Memory: {268509092: -9, 268509096: 23, 268501000: 31, 268501004: 180, 268501008: 99}
PC: 4194384 Data Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194328 Data Memory:
{268569992: -9, 268569996: 23, 268591090: 31, 268561004: 180, 268591098: 99} PC: 4194332 Data Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194360
Data Memory: (268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194364
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194364 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99}
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194364 Data Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194364 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194368 Data Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194364 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194368 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194372 Data Memory:
{268508992: -9, 268508996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194364 Data Nemory: {268508992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194368 Data Nemory: {268508992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194372 Data Memory: {268508992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194376 Data Nemory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194386 Data Nemory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Nemory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194364 Data Nemory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194368 Data Nemory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194372 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194376 Data Nemory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Nemory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Nemory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Nemory: {2685000992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Nemory: {2685000992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194384 Data Nemory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194364 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194368 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194372 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194376 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380
{268509992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194364 Data Nemory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194372 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194376 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194376 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194384 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194328 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194328 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194332
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194364 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194372 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194372 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194376 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194384 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194384 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194328 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194332 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194332 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194332 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194332 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194332 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99}
{268508992: -9, 268508996: 23, 268501808: 31, 268501884: 188, 268501888: 99} PC: 4194364 Data Memory: {268508992: -9, 268508996: 23, 268501889: 31, 268501884: 188, 268581888: 99} PC: 4194372 Data Memory: {268508992: -9, 268508996: 23, 268581888: 31, 268581884: 188, 268581888: 99} PC: 4194372 Data Memory: {268508992: -9, 2685888996: 23, 268581888: 31, 268581884: 188, 268581888: 99} PC: 4194376 Data Memory: {268508992: -9, 2685888996: 23, 268581888: 31, 268581884: 188, 268581888: 99} PC: 4194388 Data Memory: {268588992: -9, 2685888996: 23, 268581888: 31, 268581884: 188, 268581888: 99} PC: 4194384 Data Memory: {268588992: -9, 2685888996: 23, 268581888: 31, 268581884: 188, 268581888: 99} PC: 4194384 Data Memory: {268588992: -9, 2685888996: 23, 268581888: 31, 268581884: 188, 268581888: 99} PC: 4194328 Data Memory: {268588992: -9, 268588996: 23, 268581888: 31, 268581884: 188, 268581888: 99} PC: 4194332 Data Memory: {268588992: -9, 268588996: 23, 268581888: 31, 268581884: 188, 268581888: 99} PC: 4194332 Data Memory: {268588992: -9, 268588996: 23, 268581888: 31, 268581884: 188, 268581888: 99} PC: 4194336 Data Memory: {268588992: -9, 268588996: 23, 268581888: 31, 268581884: 188, 268581888: 99} PC: 4194336 Data Memory: {268588992: -9, 268588996: 23, 268581888: 31, 268581884: 188, 268581888: 99} PC: 4194336 Data Memory: {268588992: -9, 268588996: 23, 268581888: 31, 268581884: 188, 268581888: 99} PC: 4194336 Data Memory: {268588992: -9, 268588996: 23, 268581888: 31, 268581884: 188, 268581888: 99}
{268508992: -9, 268508996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194364 Data Memory: {268508992: -9, 268508996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194372 Data Memory: {268508992: -9, 268508996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194372 Data Memory: {268508992: -9, 268508996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194376 Data Memory: {268508992: -9, 268508996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Memory: {268508992: -9, 268508996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194384 Data Memory: {268508992: -9, 268508996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194384 Data Memory: {268508992: -9, 268508996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194328 Data Memory: {268508992: -9, 268508996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194332 Data Memory: {268508992: -9, 268508996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194332 Data Memory: {268508992: -9, 268508996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194336 Data Memory: {268508992: -9, 268508996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194336 Data Memory: {268508992: -9, 268508996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194336 Data Memory: {268508992: -9, 268508996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194336 Data Memory: {268508992: -9, 268508996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194336 Data Memory: {268508992: -9, 268508996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194336 Data Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194364 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194376 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194376 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194376 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194384 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194328 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194332 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194332 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194336 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194336 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194340 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194340 Data Memory: {268500092: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194340 Data Memory: {268500092: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194340 Data Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194364 Data Nemory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194368 Data Nemory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194372 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194376 Data Nemory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Nemory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Nemory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194384 Data Nemory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194332 Data Nemory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194332 Data Nemory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194336 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194336 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194340 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194340 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194340 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194340 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194340 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194340 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194368 Data Nemory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194372 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194372 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194376 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194384 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194332 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194336 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194336 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194336 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194340 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194340 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194340 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194388 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194388 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194388 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194388 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194388 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194388
{268560992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194364 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194376 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194376 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194376 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194384 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194328 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194332 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194336 Data Memory: {2685000992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194336 Data Memory: {2685000992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194340 Data Memory: {2685000992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194340 Data Memory: {2685000992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194340 Data Memory: {2685000992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194340 Data Memory: {2685000992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194340 Data Memory: {2685000992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194340 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194340 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194396
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194156 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194156 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194376 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194376 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194380 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194384 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194328 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194328 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194332 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194336 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194336 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194340 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194384 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194384 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194388 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194392 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194392 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194392 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99} PC: 4194392 Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99}

```
Data Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99}
PC: 4194404
Data Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99}
PC: 4194408
Data Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99}
PC: 4194412
Data Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99}
PC: 4194416
Data Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99}
PC: 4194420
PC. 4777720
Data Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99}
PC: 4194424
Data Memory
 {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99}
PC: 4194428
Data Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99}
PC: 4194432
Data Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99}
Data Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99}
PC: 4194388
 Nata Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99}
Data Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99}
PC: 4194348
Data Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99}
PC: 4194352
Data Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99}
PC: 4194356
Data Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99}
PC: 4194368
Data Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 180, 268501008: 99}
Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 99, 268501008: 99}
DC - #19#376
 Data Memory:
{268509992: -9, 268500996: 23, 268501000: 31, 268501004: 99, 268501008: 180}
PC: 4194380
Data Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 99, 268501008: 180}
PC: 4194384
Data Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 99, 268501008: 180}
PC: 4194328
Data Memory: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 99, 268501008: 180}
Data Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 99, 268501008: 180}
PC: 4194440
Data Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 99, 268501008: 180}
PC: 4194444
Data Memory:
{268500992: -9, 268500996: 23, 268501000: 31, 268501004: 99, 268501008: 180}
Result: {268500992: -9, 268500996: 23, 268501000: 31, 268501004: 99, 268501008: 180}
```