

Step: 0	Multiplicand: 11110010	Product: 00000000	00000101	0
Step: 1	Multiplicand: 11110010	Product: 00000111	00000010	1
Step: 2	Multiplicand: 11110010	Product: 01111100	10000001	0
Step: 3	Multiplicand: 11110010	Product: 01000101	01000000	1
Step: 4	Multiplicand: 11110010	Product: 00011011	10100000	0
Step: 5	Multiplicand: 11110010	Product: 00001101	11010000	0
Step: 6	Multiplicand: 11110010	Product: 00000110	11101000	0
Step: 7	Multiplicand: 11110010	Product: 00000011	01110100	0
Step: 8	Multiplicand: 11110010	Product: 00000001	10111010	0

Product: 10111010

Decimal Result: -70

Initial Values: A: 0000 Q: 0110 M: 0100

step: 1 Left Shift and Subtract: A: 1100

A: 1100 Q: 110_ - MSB=1

A: 0000 Q: 1100 - A=A+M

step: 2 Left Shift and Subtract: A: 1101

A: 1101 Q: 100_ - MSB=1

A: 0001 Q: 1000 - A=A+M

step: 3 Left Shift and Subtract: A: 1111

A: 1111 Q: 000_ - MSB=1

A: 0011 Q: 0000 - A=A+M

step: 4 Left Shift and Subtract: A: 0010

A: 0010 Q: 000_ Successful

A: 0010 Q: 0001 -No Restoration

Quotient(Q): 0001 Remainder(A): 0010

First Fit

```
enter the number of memory blocks4
enter the number of processes4
enter the size of block100
enter the size of block200
enter the size of block300
enter the size of block400
enter the size of process220
enter the size of process330
enter the size of process440
enter the size of process500
Process No. Process Size    Block no.
1           220           3
2           330           4
3           440          Not Allocated
4           500          Not Allocated
```

Best Fit

```
enter the number of memory blocks4
enter the number of processes4
enter the size of block100
enter the size of block200
enter the size of block300
enter the size of block400
enter the size of process500
enter the size of process110
enter the size of process220
enter the size of process390
Process No. Process Size    Block no.
1           500          Not Allocated
2           110           2
3           220           3
4           390           4
```

Worst Fit

```
enter the number of memory blocks4
enter the number of processes4
enter the size of block100
enter the size of block200
enter the size of block300
enter the size of block400
enter the size of process110
enter the size of process220
enter the size of process330
enter the size of process400
Process No. Process Size    Block no.
1           110           4
2           220           3
3           330          Not Allocated
4           400          Not Allocated
```

length of pages sequence: 10
pages sequence: 1 2 3 1 2 4 4 1 8 9
Enter the number of frames: 3

For page 1: Page fault occurred. Page 1 replaces page -1 in memory.
Current Page Frames: 1 - -
For page 2: Page fault occurred. Page 2 replaces page -1 in memory.
Current Page Frames: 1 2 -
For page 3: Page fault occurred. Page 3 replaces page -1 in memory.
Current Page Frames: 1 2 3
For page 1: Page hit occurred. No replacement needed.
Current Page Frames: 1 2 3
For page 2: Page hit occurred. No replacement needed.
Current Page Frames: 1 2 3
For page 4: Page fault occurred. Page 4 replaces page 1 in memory.
Current Page Frames: 4 2 3
For page 4: Page hit occurred. No replacement needed.
Current Page Frames: 4 2 3
For page 1: Page fault occurred. Page 1 replaces page 2 in memory.
Current Page Frames: 4 1 3
For page 8: Page fault occurred. Page 8 replaces page 3 in memory.
Current Page Frames: 4 1 8
For page 9: Page fault occurred. Page 9 replaces page 4 in memory.
Current Page Frames: 9 1 8

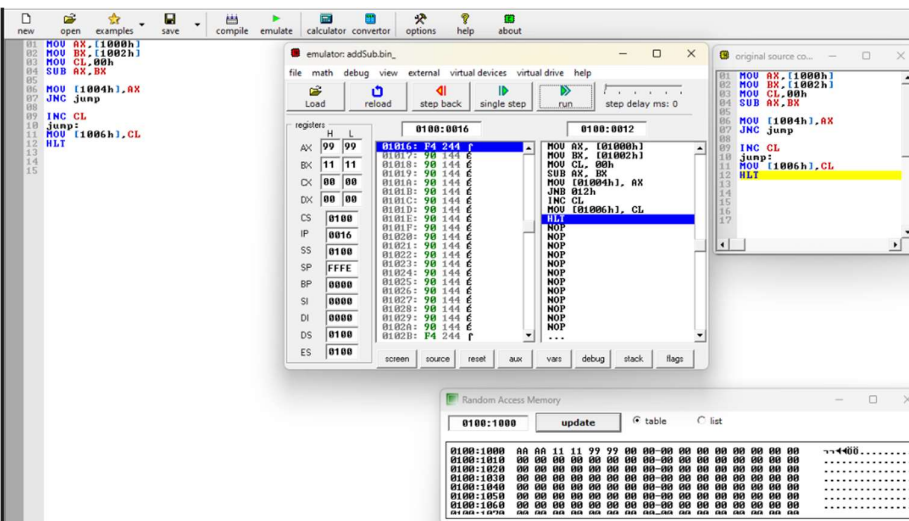
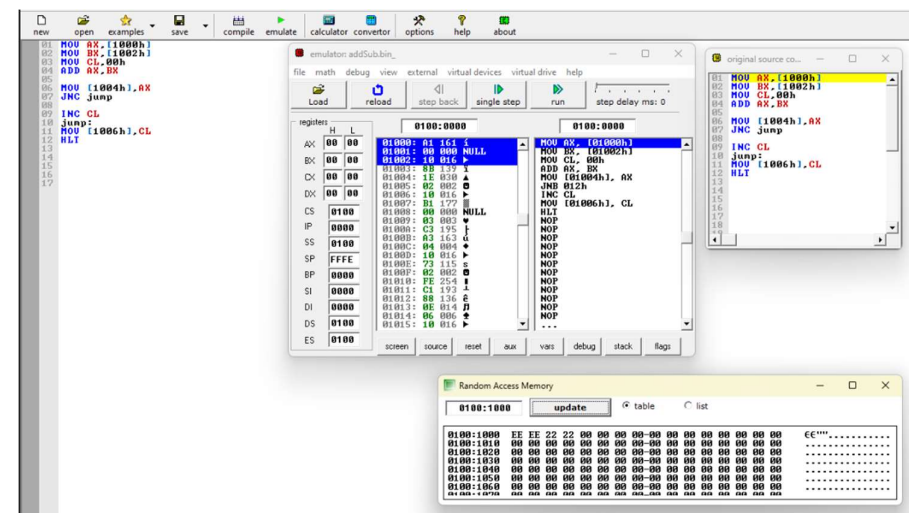
For page 9: Page fault occurred. Page 9 replaces page 4 in memory.
Current Page Frames: 9 1 8
Total number of page faults: 7

Page fault ratio: 0.70

Enter no of frames:
3
Enter no of page references:
10
Enter page references:
1 2 3 2 4 5 6 4 8 1
F1 F2 F3
1 - -
1 2 -
1 2 3
1 2 3
4 2 3
4 2 5
4 6 5
4 6 5
4 6 8
4 1 8
Number of Page Faults:8
Number of Page Hits:2

Stream	Frame1	Frame2	Frame3
7	7	-	-
0	7	0	-
1	7	0	1
2	2	0	1
0	2	0	1
3	2	0	3
0	2	0	3
4	2	4	3
2	2	4	3
3	2	4	3
0	2	0	3
3	2	0	3
2	2	0	3
1	2	0	1
2	2	0	1
0	2	0	1
1	2	0	1
7	7	0	1
0	7	0	1
1	7	0	1

Hits: 11
Misses: 9



emulator: ascDes.exe

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers H L

Register	H	L
AX	00	00
BX	00	00
CX	00	36
DX	00	00
CS	0711	
IP	0000	
SS	0710	
SP	0000	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0711:0026 0711:0000

07136: 90 144 E MOV AX, 00710h
 07137: 90 144 E MOV DS, AX
 07138: 90 144 E MOV CH, 04h
 07139: 90 144 E MOV CL, 04h
 0713A: 90 144 E MOV SI, 00000h
 0713B: 90 144 E MOV AL, [SI]
 0713C: 90 144 E MOV BL, [SI] + 01h
 0713D: 90 144 E JNB 010h
 0713E: 90 144 E MOV DL, [SI] + 01h
 0713F: 90 144 E XCHG [SI], DL
 07140: 90 144 E NOP
 07141: 90 144 E INC SI
 07142: 90 144 E DEC CL
 07143: 90 144 E JNE 0Ch
 07144: 90 144 E DEC CH
 07145: 90 144 E JNE 07h
 07146: 90 144 E NOP
 07147: 90 144 E NOP
 07148: 90 144 E NOP
 07149: 90 144 E NOP
 0714A: F4 244 F JNC 010h
 0714B: 00 000 NULL

0711:003A 0711:003A

0714B: 90 144 E MOV AX, 00710h
 0714C: 90 144 E MOV DS, AX
 0714D: 90 144 E MOV CH, 04h
 0714E: 90 144 E MOV CL, 04h
 0714F: 90 144 E MOV SI, 00000h
 07150: 90 144 E MOV AL, [SI]
 07151: 90 144 E MOV BL, [SI] + 01h
 07152: 90 144 E JNB 010h
 07153: 90 144 E MOV DL, [SI] + 01h
 07154: 90 144 E XCHG [SI], DL
 07155: 90 144 E NOP
 07156: 90 144 E INC SI
 07157: 90 144 E DEC CL
 07158: 90 144 E JNE 0Ch
 07159: 90 144 E DEC CH
 0715A: 90 144 E JNE 07h
 0715B: 90 144 E NOP
 0715C: 90 144 E NOP
 0715D: 90 144 E NOP

DATA SEGMENT
 STRING1 DB 99h,12h,56h,4
 DATA ENDS

CODE SEGMENT
 ASSUME CS:CODE,DS:DATA
 START:MOV AX,DATA
 MOV DS,AX
 MOV CH,04h
 UP2:MOV CL,04h
 LEA SI,STRING1
 UP1:MOV AL,[SI]
 MOV BL,[SI+1]
 CMP AL,BL
 JNC DOWN
 MOV DL,[SI+1]
 XCHG [SI],DL
 MOV [SI+1],DL
 DOWN:INC SI
 DEC CL
 JNZ UP1
 DEC CH
 JNZ UP2

variables

size: byte elements: 5

edit show as: hex

STRING1 12h, 36h, 45h, 56h, 99h

CF 1 ZF 1 SF 0 OF 0 AF 0 PF 1 IF 1 DF 0

analyse

registers H L

Register	H	L
AX	00	00
BX	00	00
CX	00	36
DX	00	00
CS	0711	
IP	0000	
SS	0710	
SP	0000	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0711:0000 0711:0000

07110: B8 184 1 MOV AX, 00710h
 07111: 10 016 J MOV DS, AX
 07112: 07 007 BEEP MOV CH, 04h
 07113: 8E 142 MOV CL, 04h
 07114: D8 216 MOV SI, 00000h
 07115: B5 181 MOV AL, [SI]
 07116: 04 004 MOV BL, [SI] + 01h
 07117: B1 177 CMP AL, BL
 07118: 04 004 JNB 010h
 07119: DE 198 MOV DL, [SI] + 01h
 0711A: 00 000 XCHG [SI], DL
 0711B: 00 000 MOV [SI] + 01h, DL
 0711C: 8A 138 INC SI
 0711D: 04 004 DEC CL
 0711E: 8A 138 JNE 0Ch
 0711F: 5C 092 DEC CH
 07120: 01 001 JNE 07h
 07121: 3A 058 NOP
 07122: C3 195 NOP
 07123: 73 115 NOP
 07124: 08 008 BACK NOP
 07125: 8A 138

DATA SEGMENT
 STRING1 DB 99h,12h,56h
 DATA ENDS

CODE SEGMENT
 ASSUME CS:CODE,DS:DATA
 START:MOV AX,DATA
 MOV DS,AX
 MOV CH,04h
 UP2:MOV CL,04h
 LEA SI,STRING1
 UP1:MOV AL,[SI]
 MOV BL,[SI+1]
 CMP AL,BL
 JNC DOWN
 MOV DL,[SI+1]
 XCHG [SI],DL
 MOV [SI+1],DL
 DOWN:INC SI
 DEC CL
 JNZ UP1
 DEC CH
 JNZ UP2

variables

size: byte elements: 5

edit show as: hex

STRING1 99h, 56h, 45h, 36h, 12h

CF 0 ZF 0 SF 0

analyse

edit: C:\Users\vishw\OneDrive\Desktop_DJS_SEM 4\POA\codes\exp8.asm

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator convertor options help about

```
01 DATA SEGMENT
02 ARR DB 9,1,5,7,2,6,8,6,4
03 LEN DW ?
04 MIN DB ?
05 MAX DB ?
06 DATA ENDS
07
08 CODE SEGMENT
09 ASSUME DS:DATA, CS:CODE
10 START:
11 MOV AX, DATA
12 MOV DS, AX
13
14 LEA SI, ARR
15 MOV AL, [SI]
16 MOV MIN, AL
17 MOV MAX, AL
18
19 MOV CX, LEN
20 INC SI ; Move to the second element in the array
21
22 REPEAT:
23 MOV AL, [SI] ; Load the value at the current index
24 CMP AL, MIN ; Compare with MIN
25 JL UPDATE_MIN ; If less than MIN, update MIN
26 CMP AL, MAX ; Compare with MAX
27 JG UPDATE_MAX ; If greater than MAX, update MAX
28 INC SI
29 LOOP REPEAT ; Repeat until all elements are processed
30
31 UPDATE_MIN:
32 MOV MIN, AL ; Update MIN
33 JMP CONTINUE
34
35 UPDATE_MAX:
36 MOV MAX, AL ; Update MAX
37 JMP CONTINUE
38
39 CONTINUE:
40 MOV AH, 4CH
41 INT 21H
42
43 CODE ENDS
44 END START
45
46
47
48 code by Uishwa Jarsaniya, C-185
```

emulator: exp8.exe_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	4C	01
BX	00	00
CX	00	09
DX	00	00
CS	F400	
IP	0204	
SS	0710	
SP	FFFA	
BP	0000	
SI	0001	
DI	0000	
DS	0710	
ES	0700	

F400:0204

F4200:	FF	25	55	RES
F4201:	FF	25	55	RES
F4202:	CD	20	5	=
F4203:	21	03	3	!
F4204:	CF	20	7	!
F4205:	00	00	00	NULL
F4206:	00	00	00	NULL
F4207:	00	00	00	NULL
F4208:	00	00	00	NULL
F4209:	00	00	00	NULL
F420A:	00	00	00	NULL
F420B:	00	00	00	NULL
F420C:	00	00	00	NULL
F420D:	00	00	00	NULL
F420E:	00	00	00	NULL
F420F:	00	00	00	NULL
F4210:	00	00	00	NULL
F4211:	00	00	00	NULL
F4212:	00	00	00	NULL
F4213:	00	00	00	NULL
F4214:	00	00	00	NULL
F4215:	00	00	00	NULL

BIOS DI

INT 021h	IRET
ADD [BX + SI], AL	
ADD [BX + SI], AL	
ADD [BX + SI], AL	
ADD [BX + SI], AL	
ADD [BX + SI], AL	
ADD [BX + SI], AL	
ADD [BX + SI], AL	
ADD [BX + SI], AL	
ADD [BX + SI], AL	
ADD [BX + SI], AL	
ADD [BX + SI], AL	
ADD [BX + SI], AL	
ADD [BX + SI], AL	
ADD [BX + SI], AL	
ADD [BX + SI], AL	
...	

original source code

```
22 REPEAT:
23 MOV AL, [SI] ; Load the
24 CMP AL, MIN ; Compare
25 JL UPDATE_MIN ; If less
26 CMP AL, MAX ; Compare
27 JG UPDATE_MAX ; If grea
28 INC SI
29 LOOP REPEAT ; Repeat
30
31 UPDATE_MIN:
32 MOV MIN, AL ; Update
33 JMP CONTINUE
34
35 UPDATE_MAX:
36 MOV MAX, AL ; Update
37 JMP CONTINUE
38
39 CONTINUE:
40 MOV AH, 4CH
41 INT 21H
```

variables

size: byte elements: 1

edit show as: hex

ARR	09h
LEN	0009h
MIN	01h
MAX	09h

Random Access Memory

0710:0000 update table list

0710:0000	09 01 05 07 02 06 08 06-04 07 00 01 07 00 00 00	.09.01.05.07.02.06.08.06-04.07.00.01.07.00.00.00
0710:0010	B8 10 07 8E D8 BE 00 00-8A 04 A2 0B 00 A2 0C 00	B8.10.07.8E.D8.BE.00.00-8A.04.A2.0B.00.A2.0C.00
0710:0020	8B 0E 09 00 46 8A 04 3A-06 0B 00 7C 09 3A 06 0C	8B.0E.09.00.46.8A.04.3A-06.0B.00.7C.09.3A.06.0C
0710:0030	00 7F 08 46 E2 EF A2 0B-00 EB 05 A2 0C 00 EB 00	00.7F.08.46.E2.EF.A2.0B-00.EB.05.A2.0C.00.EB.00
0710:0040	B4 4C CD 21 90 90 90 90-90 90 90 90 90 90	B4.4C.CD.21.90.90.90.90-90.90.90.90.90.90.90.90
0710:0050	90 90 90 90 90 90 90 90-90 90 90 90 90 90	90.90.90.90.90.90.90.90-90.90.90.90.90.90.90.90
0710:0060	00 00 00 00 00 00 00 00-00 00 00 00 00 00	00.00.00.00.00.00.00.00-00.00.00.00.00.00.00.00
0710:0070	00 00 00 00 00 00 00 00-00 00 00 00 00 00	00.00.00.00.00.00.00.00-00.00.00.00.00.00.00.00

emulator: noname.exe

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L		
AX	00	00	0710:0002	0721:0000
BX	00	00	0710:0000	0710:0000
CX	01	29	0710:0000	0710:0000
DX	00	00	0710:0000	0710:0000
IP	0000		0710:0000	0710:0000
SP	0000		0710:0000	0710:0000
BP	0000		0710:0000	0710:0000
SI	0000		0710:0000	0710:0000
DI	0000		0710:0000	0710:0000
DS	0700		0710:0000	0710:0000
ES	0700		0710:0000	0710:0000

screen source variables

size: word elements: 1

edit Load reload step back single step run step delay ms: 0

CF 0

NUM 0000h

stack segment

num du 05h

result du ?

ends

stack segment

du 128 dup(0)

ends

code segment

start:

mov ax,data

mov ds,ax

mov cx,num

mov ax,0001h

fact num

mov result,ax

code

registers

	H	L		
AX	00	78	0721:0020	0721:0020
BX	00	00	0721:0020	0721:0020
CX	00	05	0721:0020	0721:0020
DX	00	00	0721:0020	0721:0020
IP	0000		0721:0020	0721:0020
SP	0100		0721:0020	0721:0020
BP	0000		0721:0020	0721:0020
SI	0000		0721:0020	0721:0020
DI	0000		0721:0020	0721:0020
DS	0710		0721:0020	0721:0020
ES	0700		0721:0020	0721:0020

screen source reset aux vars debug stack flags

size: word elements: 1

edit show as: hex

SF 1

ZF 1

OF 0

PF 1

AF 0

IF 1

DF 0

analyse

emulator: noname.exe

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L		
AX	4C	30	F400:0204	F400:0204
BX	00	20	F400:0204	F400:0204
CX	00	39	F400:0204	F400:0204
DX	00	00	F400:0204	F400:0204
IP	0000		F400:0204	F400:0204
SP	0000		F400:0204	F400:0204
BP	0000		F400:0204	F400:0204
SI	0000		F400:0204	F400:0204
DI	0000		F400:0204	F400:0204
DS	0710		F400:0204	F400:0204
ES	0700		F400:0204	F400:0204

screen source reset aux vars debug stack flags

size: byte elements: 1

edit show as: hex

NUM1 50h

NUM2 20h

ADD_RES 70h

SUB_RES 30h

emulator: noname.exe

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L		
AX	00	00	0711:0000	0711:0000
BX	00	00	0711:0000	0711:0000
CX	00	39	0711:0000	0711:0000
DX	00	00	0711:0000	0711:0000
IP	0000		0711:0000	0711:0000
SP	0000		0711:0000	0711:0000
BP	0000		0711:0000	0711:0000
SI	0000		0711:0000	0711:0000
DI	0000		0711:0000	0711:0000
DS	0700		0711:0000	0711:0000
ES	0700		0711:0000	0711:0000

screen source reset aux vars debug stack flags

size: byte elements: 1

edit show as: hex

NUM1 50h

NUM2 20h

ADD_RES 60h

SUB_RES 60h