UCS1512 – Microprocessors Lab

16 BIT ARITHMETIC OPERATIONS

Exp no : 3 Name: Sreedhar V

Date : 11-09-2020 Reg no: 185001161

AIM:

To program and execute the string manipulation like moving a string of bytes,comparing 2 strings of bytes,searching a byte in a string and moving a string without using string instructions in 8086 using an emulator.

Moving a string of bytes:

Algorithm:

- Program is set to run from any specified memory position.
- Move the address of data segment to register DS .
- Move the address of extra segment to register ES .
- Move the offset of the source and dest to SI and DI respectively .
- Initialize register CX to size of the source (No. Of bytes).
- Clear the direction flag using CLD instruction.
- Repetitively move the each byte of source to destination using rep movsb.
- Terminate the program.

Program:

CODE	COMMENT
Program for moving a string of bytes:	
assume code: cs,ds:data,es:extra data segment source db 11h,12h,13h,14h data ends	Data segment is initialized source is initialized and set to 11h,12h,13h,14h
extra segment dest db ? extra ends	Extra segment is initialized dest is initialized

code segment
org 0100h
start : mov ax,data
mov ds,ax
 	
mov ax,data
mov es,ax
	Code segment begins
	Originating address is set to 0100h
	Address of the data and extra are transferred to AX , from AX transferred to DS and ES respectively

mov si,offset source mov di,offset dest mov cx,0004h cld	Offset of the source and dest are transferred to SI and DI respectively Initialize CX to 0004h,since string of 4 bytes is used Clear the direction flag
rep movsb	Move each byte of string from SI to DI till CX becomes 0
mov ah,4ch int 21h	Program terminates
code ends ends start	

Unassembled code:

```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip  0, Progra...

D:\>debug 3a.exe
-u
076C:0000 B86A07        MOV     AX,076A
076C:0003 8ED8          MOV     DS,AX
076C:0005 B86B07        MOV     AX,076B
076C:0008 8EC0          MOV     ES,AX
076C:000A BE0000        MOV     SI,0000
076C:000D BF0000        MOV     DI,0000
076C:0010 B90400        MOV     CX,0004
076C:0013 FC            CLD
076C:0014 F3            REPZ
076C:0015 A4            MOVSB
076C:0016 B44C          MOV     AH,4C
076C:0018 CD21          INT     21
076C:001A 46            INC     SI
076C:001B FE7709        PUSH    [BX+09]
076C:001E 8946FE        MOV     [BP-02],AX
-_
```

Sample input and output:

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
D:\>debug 3a.exe
-e 076a:0000
076A:0000  11.11   12.12   13.13   14.14

-d 076a:0000
076A:0000  11 12 13 14 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0020  B8 6A 07 8E D8 B8 6B 07-8E C0 BE 00 00 BF 00 00   .j....k.........
076A:0030  B9 04 00 FC F3 A4 B4 4C-CD 21 46 FE 77 09 89 46   .......L.!F.w..F
076A:0040  FE 8A 46 F9 88 46 F8 FE-46 F9 EB C9 8A 5E F8 B7   ..F..F..F....^..
076A:0050  00 8A 87 48 2F D0 D8 73-17 E8 B6 00 8A 5E F8 B7   ...H/..s.....^..
076A:0060  00 8A 87 48 2F D0 D8 73-07 53 B0 01 50 E8 73 01   ...H/..s.S..P.s.
076A:0070  A0 B6 2C 3A 46 F8 74 7E-C7 46 FA 00 00 8A 46 F8   ..,:F.t~.F....F.
-
```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
-g

Program terminated normally
-d 076b:0000
076B:0000  11 12 13 14 00 00 00 00-00 00 00 00 00 00 00 00   ................
076B:0010  B8 6A 07 8E D8 B8 6B 07-8E C0 BE 00 00 BF 00 00   .j....k.........
076B:0020  B9 04 00 FC F3 A4 B4 4C-CD 21 46 FE 77 09 89 46   .......L.!F.w..F
076B:0030  FE 8A 46 F9 88 46 F8 FE-46 F9 EB C9 8A 5E F8 B7   ..F..F..F....^..
076B:0040  00 8A 87 48 2F D0 D8 73-17 E8 B6 00 8A 5E F8 B7   ...H/..s.....^..
076B:0050  00 8A 87 48 2F D0 D8 73-07 53 B0 01 50 E8 73 01   ...H/..s.S..P.s.
076B:0060  A0 B6 2C 3A 46 F8 74 7E-C7 46 FA 00 00 8A 46 F8   ..,:F.t~.F....F.
076B:0070  B4 00 B1 05 D3 E0 03 06-B4 2C 89 46 FC 8A 1E B6   .........,.F....
-
-
-
-
-
-
-
-
-
-
-
-
-
```

Result:

Moving a string of bytes is executed and verified using an emulator.

Comparing 2 strings of bytes:

Algorithm:

- Program is set to run from any specified memory position.
- Move the address of data segment to register DS .
- Move the address of extra segment to register ES .
- Move the offset of the source and dest to SI and DI respectively .
- Initialize register CX to size of the source (No. Of bytes)
- Clear the direction flag using CLD instruction.
- Increment the CX register.
- Repeat until the bytes of source and dest strings are equal.
- Store the value of CX to status variable (Indicating the index of mismatch).
- Terminate the program.

Program:

CODE	COMMENT
Program for Comparing 2 strings of bytes:	
assume code: cs,ds:data,es:extra data segment source db 10h,20h,30h,40h status dw 0000h data ends	Data segment is initialized source is initialized and set to 10h,20h,30h,40h status is initialized and set to 0000h
extra segment dest db 10h,22h,30h,40h extra ends	Extra segment is initialized dest is initialized and set to 10h,22h,30h,40h
code segment org 0100h start : mov ax,data mov ds,ax	Code segment begins Originating address is set to 0100h Address of the data and extra are transferred to AX , from AX transferred to DS and ES respectively

mov ax,data
mov es,ax
 	
mov si,offset source
mov di,offset dest
mov cx,0004h
inc cx
cld
	Offset of the source and dest are transferred to SI and DI respectively
	Initialize CX to 0004h,since string of 4 bytes is used
	Increment CX
	Clear the direction flag

repe cmpsb	Compare each byte of string from SI and DI till CX becomes 0

mov status,cx
mov ah,4ch
int 21h
	Transfer data from CX to status
	
	Program terminates

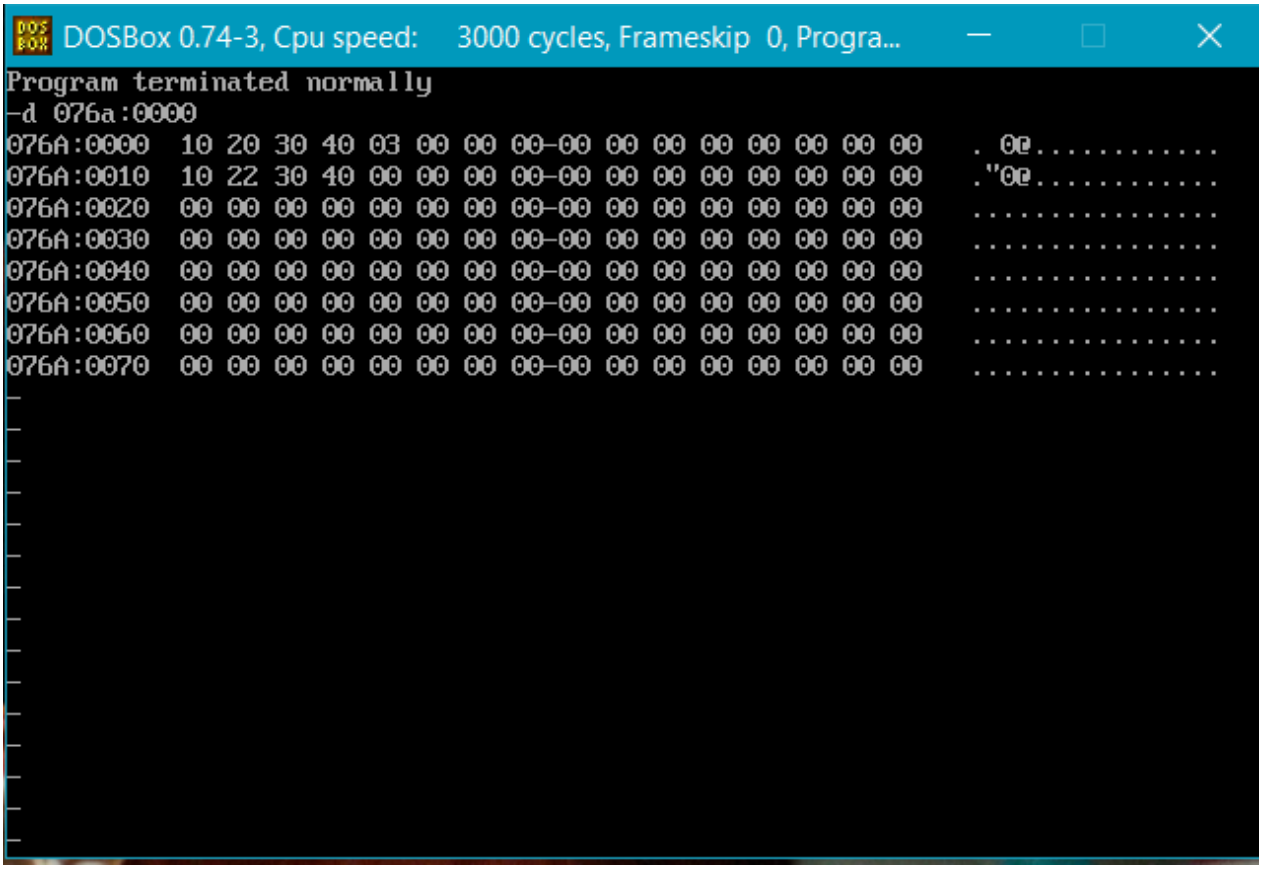
code ends ends start	

Unassembled code:

```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Progra...
D:\>debug 3b.exe
-u
076C:0100 B86A07        MOV     AX,076A
076C:0103 8ED8          MOV     DS,AX
076C:0105 B86B07        MOV     AX,076B
076C:0108 8EC0          MOV     ES,AX
076C:010A BE0000        MOV     SI,0000
076C:010D BF0000        MOV     DI,0000
076C:0110 B90400        MOV     CX,0004
076C:0113 41            INC     CX
076C:0114 FC            CLD
076C:0115 F3            REPZ
076C:0116 A6            CMPSB
076C:0117 890E0400      MOV     [0004],CX
076C:011B B44C          MOV     AH,4C
076C:011D CD21          INT     21
076C:011F FF7201        PUSH    [BP+SI+01]
-
-
-
-
-
-
-
-
```

Execution:

```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Progra...
-d 076a:0000
076A:0000  10 20 30 40 00 00 00 00-00 00 00 00 00 00 00 00   . 0@............
076A:0010  10 22 30 40 00 00 00 00-00 00 00 00 00 00 00 00   ."0@............
076A:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
-d 076b:0000
076B:0000  10 22 30 40 00 00 00 00-00 00 00 00 00 00 00 00   ."0@............
076B:0010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076B:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076B:0030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076B:0040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076B:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076B:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076B:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
-
-
-
-
-
-
-_
```



Result:

Comparing 2 strings of bytes is executed and verified using an emulator.

Searching a byte in a string:

Algorithm:

- Program is set to run from any specified memory position.
- Move the address of data segment to register DS .
- Move the address of extra segment to register ES .
- Move the source to Accumulator Register AL.
- Move the offset of the dest to DI .
- Increment CX register.
- Clear the direction flag using CLD instruction.
- Initialize register CX to size of the source (No. Of bytes).
- Repeat until the source byte and DI are not equal.
- Store the value of CX to status variable (Indicating the index of match).
- Terminate the program.

Program:

CODE	COMMENT
Program for Searching a byte in a string:	
assume code: cs,ds:data,es:extra data segment source db 10h status dw 0000h data ends	Data segment is initialized source is initialized and set to 10h status is initialized and set to 0000h
extra segment dest db 30h,40h,10h,50h extra ends	Extra segment is initialized dest is initialized and set to 30h,40h,10h,50h
code segment org 0100h start : mov ax,data mov ds,ax	Code segment begins Originating address is set to 0100h Address of the data and extra are transferred to AX , from AX transferred to DS and ES respectively
mov ax,data mov es,ax	

mov al,source
mov di,offset dest
mov cx,0004h
inc cx
cld
	Value of source is transferred to AL
	Offset of the dest is transferred to DI
	Move data from source to AL register
	Initialize CX to 0004h,since string of 4 bytes is used
	Increment CX
	Clear the direction flag

repne scasb	Compare the byte of string from AL and content in DI until CX becomes 0

mov status,cx
mov ah,4ch
int 21h
	Transfer data from CX to status
	
	Program terminates

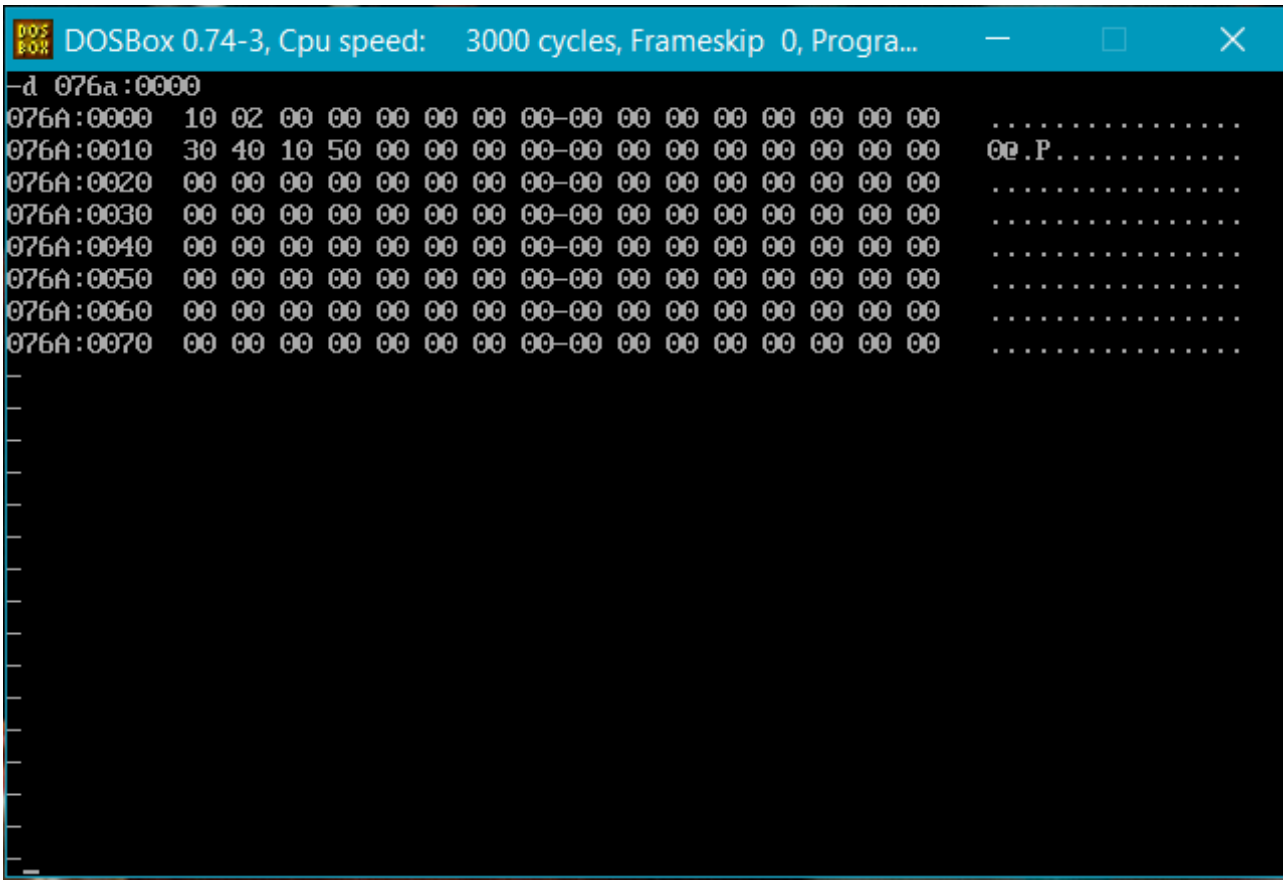
code ends ends start	

Unassembled code:

```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip  0, Progra...
D:\>debug 3c.exe
-u
076C:0100 B86A07        MOV     AX,076A
076C:0103 8ED8          MOV     DS,AX
076C:0105 B86B07        MOV     AX,076B
076C:0108 8EC0          MOV     ES,AX
076C:010A A00000        MOV     AL,[0000]
076C:010D BF0000        MOV     DI,0000
076C:0110 B90400        MOV     CX,0004
076C:0113 41            INC     CX
076C:0114 FC            CLD
076C:0115 F2            REPNZ
076C:0116 AE            SCASB
076C:0117 890E0100      MOV     [0001],CX
076C:011B B44C          MOV     AH,4C
076C:011D CD21          INT     21
076C:011F FF7201        PUSH    [BP+SI+01]
-
-
-
-
-
-
-
-
```

Execution:

```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip  0, Progra...
-d 076a:0000
076A:0000  10 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0010  30 40 10 50 00 00 00 00-00 00 00 00 00 00 00 00   0@.P............
076A:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
-d 076b:0000
076B:0000  30 40 10 50 00 00 00 00-00 00 00 00 00 00 00 00   0@.P............
076B:0010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076B:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076B:0030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076B:0040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076B:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076B:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076B:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
-
-
-
-
-
-
-
-_
```



Result:

Searching a byte in a string is executed and verified using an emulator.

Moving a string without using string instruction:

Algorithm:

- Load data from opr1 to register AX (first number).
- Move the address of data segment to register DS .
- Move the offset of the source and dest to SI and DI respectively .
- Initialize register CX to size of the source (No. Of bytes).
- Clear the direction flag using CLD instruction.
- Repetitively move the each byte of source to destination using an explicit loop untill CX becomes 0 and increment SI and DI index registers each time inside the loop.
- Terminate the program.

Program:

CODE	COMMENT
Program for moving a string without using string instructions:	
assume code: cs,ds:data data segment source db 11h,12h,13h,14h dest db ? data ends	Data segment is initialized source is initialized and set to 11h,12h,13h,14h dest is initialized
code segment org 0100h start : mov ax,data mov ds,ax	Code segment begins Originating address is set to 0100h Address of the data are transferred to AX , from AX transferred to DS.
mov si,offset source mov di,offset dest mov cx,0004h cld	Offset of the source and dest are transferred to SI and DI respectively Initialize CX to 0004h,since string of 4 bytes is used Clear the direction flag
here : mov bl,[si] mov [di],bl inc si inc di loop here	Move each byte of string from SI to DI till CX becomes 0 by transferring the value in SI address to BL and from BL to DI. Increment SI and DI Loop label is here.
mov ah,4ch int 21h	Program terminates
code ends ends start	

Unassembled code:

```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip  0, Progra...
D:\>debug 3d.exe
-u
076B:0100 B86A07        MOV     AX,076A
076B:0103 8ED8          MOV     DS,AX
076B:0105 BE0000        MOV     SI,0000
076B:0108 BF0400        MOV     DI,0004
076B:010B B90400        MOV     CX,0004
076B:010E FC            CLD
076B:010F 8A1C          MOV     BL,[SI]
076B:0111 881D          MOV     [DI],BL
076B:0113 46            INC     SI
076B:0114 47            INC     DI
076B:0115 E2F8          LOOP    010F
076B:0117 B44C          MOV     AH,4C
076B:0119 CD21          INT     21
076B:011B B0FF          MOV     AL,FF
076B:011D 7701          JA      0120
076B:011F 40            INC     AX
-
-
-
-
-
-
-
```

Execution:

```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip  0, Progra...
-d 076a:0000
076A:0000  11 12 13 14 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
-g

Program terminated normally
-d 076a:0000
076A:0000  11 12 13 14 11 12 13 14-00 00 00 00 00 00 00 00   ................
076A:0010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
-
-
-
-
```

Result:

Moving a string without using string instruction is executed and verified using an emulator.