

**Source File:** lab20.asm  
**Input:** Standard Input  
**Output:** Standard Output  
**Value:** 2

Write a program to shift the elements of an integer array forward one position. The value at the end of the array must wrap around to the first position. For example, the array {10, 20, 30, 40} would be transformed into {40, 10, 20, 30}. Do not copy the elements to any other array. Use the following initialized variables:

```
1 SECTION .data
2 myArray times 40 dd 0
3 length equ ($ - myArray) / 4
```

Begin by using the `loop` instruction with indirect or indexed addressing to initialize the elements of the integer array from standard input. Use a second `loop` instruction with indirect or indexed addressing to shift the elements of the integer array as described above.

Use the following code segment as a guide to display the hexadecimal contents of an area of memory. Use this code segment to display the contents of the array three times—before the array is initialized, after the elements of the array have been initialized, and after the elements of the array have been shifted.

```
1 mov     esi, myArray    ; esi = starting offset
2 mov     ecx, length     ; ecx = # of units
3 mov     ebx, 4          ; ebx = unit size (1=byte, 2=word, 4=dword)
4 call    DumpMem
```

(The `DumpMem` procedure is explained in Section 5.4.)

Add an identification section as shown below to the top of your source file.

```
    ; Your name
    ; CS 3304
    ; Lab 20
```

A sample execution sequence is shown in Figure 1. To use the `Makefile` as distributed in class, add a target of `lab20` to `targetsAsmLanguage`.

```

1  newuser@csunix ~/3304/20> cp /usr/local/3304/data/20/* .
2  newuser@csunix ~/3304/20> make lab20
3  nasm -f elf32 -l lab20.lst -o lab20.o lab20.asm -I/usr/local/3304/include/ -I.
4  ld -m elf_i386 --dynamic-linker /lib/ld-linux.so.2 -o lab20 lab20.o \
5  /usr/local/3304/src/Along32.o -lc
6  newuser@csunix ~/3304/20> ../irvine_test.sh lab20 01.dat
7
8  Dump of offset 0804A010
9  -----
10 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
11 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
12 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
13 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
14 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
15
16 Dump of offset 0804A010
17 -----
18 FFFFFFFF FFFFFFFE FFFFFFFC FFFFFFF8 FFFFFFF0 FFFFFFFE0 FFFFFFFC0 FFFFFFF80
19 FFFFFFF0 FFFFFFFE0 FFFFFFFC0 FFFFFFF800 FFFFFFF00 FFFFFFFE00 FFFFFFFC00 FFFFFFF8000
20 FFFF0000 FFFE0000 80000001 80000000 00000000 00000001 00000002 00000004
21 00000008 00000010 00000020 00000040 00000080 00000100 00000200 00000400
22 00000800 00001000 00002000 00004000 00008000 00010000 00020000 7FFFFFFF
23
24 Dump of offset 0804A010
25 -----
26 7FFFFFFF FFFFFFFF FFFFFFFE FFFFFFFC FFFFFFF8 FFFFFFF0 FFFFFFFE0 FFFFFFFC0
27 FFFFFFF80 FFFFFFF00 FFFFFFFE00 FFFFFFFC00 FFFFFFF800 FFFFFFF00 FFFFFFFE00 FFFFFFFC000
28 FFFF8000 FFFF0000 FFFE0000 80000001 80000000 00000000 00000001 00000002
29 00000004 00000008 00000010 00000020 00000040 00000080 00000100 00000200
30 00000400 00000800 00001000 00002000 00004000 00008000 00010000 00020000
31 newuser@csunix ~/3304/20>

```

Figure 1. Commands to Assemble, Link, & Run Lab 20