ASSEMBLY - LODS INSTRUCTION

In cryptography, a Caesar cipher is one of the simplest known encryption techniques. In this method, each letter in the data to be encrypted is replaced by a letter some fixed number of positions down the alphabet.

In this example, let us encrypt a data by simply replacing each alphabet in it with a shift of two alphabets, so **a** will be substituted by **c**, **b** with **d** and so on.

We use LODS to load the original string 'password' into the memory.

```
section .text
   global _start         ;must be declared for using gcc

_start:                  ;tell linker entry point
   mov    ecx, len
   mov    esi, s1
   mov    edi, s2

loop_here:
   lodsb
   add al, 02
   stosb
   loop    loop_here
   cld
   rep     movsb

   mov     edx,20        ;message length
   mov     ecx,s2        ;message to write
   mov     ebx,1         ;file descriptor (stdout)
   mov     eax,4         ;system call number (sys_write)
   int     0x80          ;call kernel

   mov     eax,1         ;system call number (sys_exit)
   int     0x80          ;call kernel

section .data
s1 db 'password', 0 ;source
len equ $-s1

section .bss
s2 resb 10               ;destination
```

When the above code is compiled and executed, it produces the following result:

```
rcuuyqtf
```