

Q#1 Program to print an Array using Loop

section .data

array db 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

length db 10

msg db 0x0D, 0x0A, 'Array Element: ', 0

section .text

global _start

_start:

mov cx, [length] ; Load the length of the array into CX

mov si, 0 ; Initialize index to 0

print_array:

mov al, [array + si] ; Load the current array element into AL

add al, 48 ; Convert to ASCII

mov ah, 0x0E ; BIOS function to print character

int 0x10 ; Print the character

; Print new line

mov ah, 0x0E

mov al, 0x0D ; Carriage return

int 0x10

mov al, 0x0A ; Line feed

int 0x10

inc si ; Move to the next element

loop print_array ; Loop until CX is 0

```
    mov ah, 0x4C    ; Exit the program
    int 0x21
```

Q#2 Program to input string and print it

```
section .data
```

```
    msg db 'Enter a string: $'
    buffer db 50, 0
```

```
section .text
```

```
    global _start
```

```
_start:
```

```
    mov ah, 0x09
    lea dx, [msg]
    int 0x21
```

```
    mov ah, 0x0A
    lea dx, [buffer]
    int 0x21
```

```
    mov ah, 0x09
    lea dx, [buffer + 1]
    int 0x21
```

```
    mov ah, 0x4C
    int 0x21
```

Q#3 Code by your idea using array, jump and string

```
section .data
```

```
strings db 'Hello, World!$'
strings2 db 'Assembly Language$'
strings3 db 'Jump Instruction Example$'
prompt_msg db 'Select a string (0-2): $'
invalid_msg db 'Invalid selection.$'
buffer db 1, 0
```

```
section .text
```

```
global _start
```

```
_start:
```

```
mov ah, 0x09
```

```
lea dx, [prompt_msg]
```

```
int 0x21
```

```
mov ah, 0x01
```

```
int 0x21
```

```
sub al, 48
```

```
cmp al, 0
```

```
je print_string1
```

```
cmp al, 1
```

```
je print_string2
```

```
cmp al, 2
```

```
je print_string3
```

```
jmp print_invalid
```

```
print_string1:
```

```
mov ah, 0x09
lea dx, [strings]
int 0x21
jmp exit
```

print_string2:

```
mov ah, 0x09
lea dx, [strings2]
int 0x21
jmp exit
```

print_string3:

```
mov ah, 0x09
lea dx, [strings3]
int 0x21
jmp exit
```

print_invalid:

```
mov ah, 0x09
lea dx, [invalid_msg]
int 0x21
```

exit:

```
mov ah, 0x4C
int 0x21
```