Q1:

C = 10;

while(C > 0)

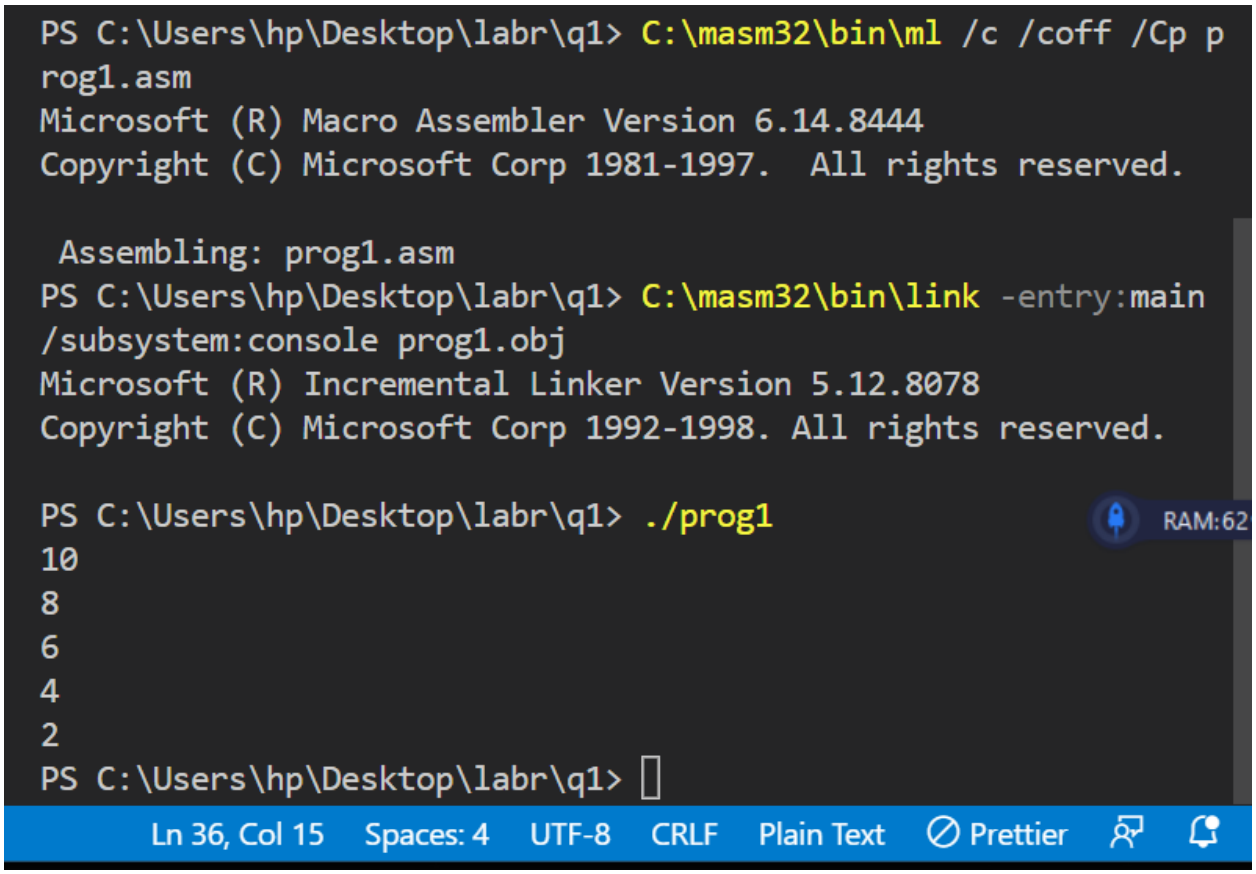
{

print(C);

C = C - 2;

}

output:



Q2:

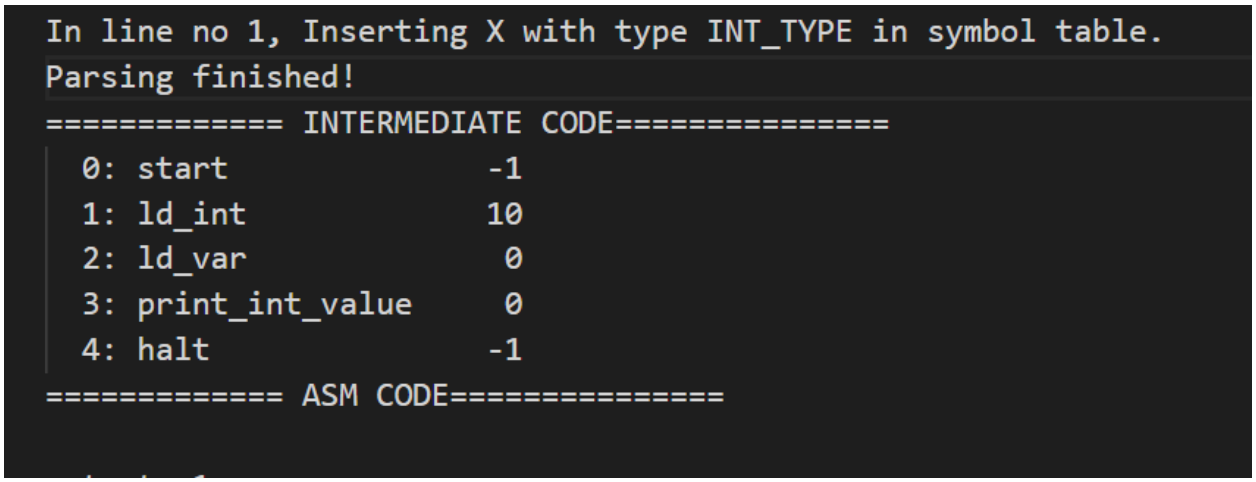
let X as integer = IN();

let Y as integer = 10 - X;

OUT(Y);

Output:

a)



b)

```
============ ASM CODE===============

;start -1
.686
.model flat, c
include C:\masm32\include\msvcrt.inc
includelib C:\masm32\lib\msvcrt.lib

.stack 100h
printf PROTO arg1:Ptr Byte, printlist:VARARG
scanf PROTO arg2:Ptr Byte, inputlist:VARARG

.data
output_integer_msg_format byte "%d", 0Ah, 0
output_string_msg_format byte "%s", 0Ah, 0
```

```
input_integer_format byte "%d",0

number sdword ?

.code

main proc
        push ebp
        mov ebp, esp
        sub ebp, 100
        mov ebx, ebp
        add ebx, 4

;ld_int 10
        mov eax, 10
        mov dword ptr [ebx], eax
        add ebx, 4


;ld_var 0
        mov eax, [ebp-0]
        mov dword ptr [ebx], eax
        add ebx, 4


;print_int_value 0
        push eax
        push ebx
        push ecx
        push edx
        push [ebp-0]
        push [ebp+4]
        push [ebp+8]
        push ebp
        mov eax, [ebp-0]
        INVOKE printf, ADDR output_integer_msg_format, eax
        pop ebp
        pop [ebp+8]
        pop [ebp+4]
        pop [ebp-0]
```

```
        pop edx
        pop ecx
        pop ebx
        pop eax

;halt -1
        add ebp, 100
        mov esp, ebp
        pop ebp
        ret
main endp
end
```