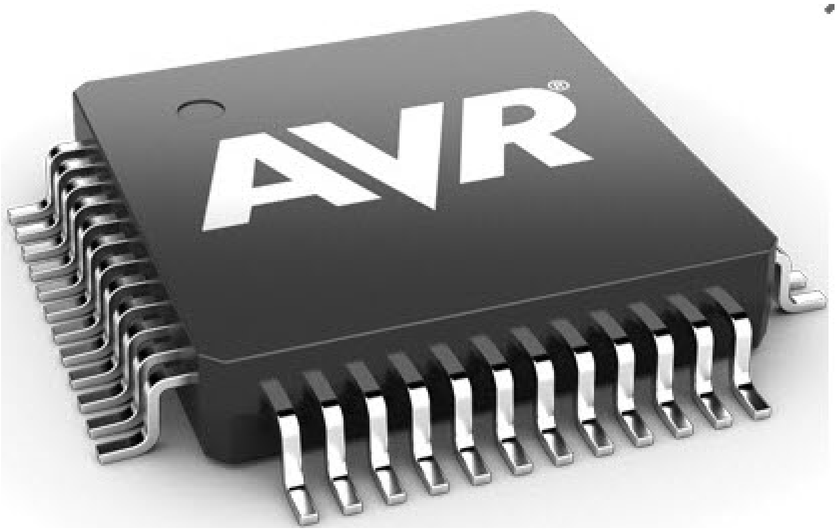
Microprocesoare si Microcontrolere

TEMA



Popa Nicolae Dănuț

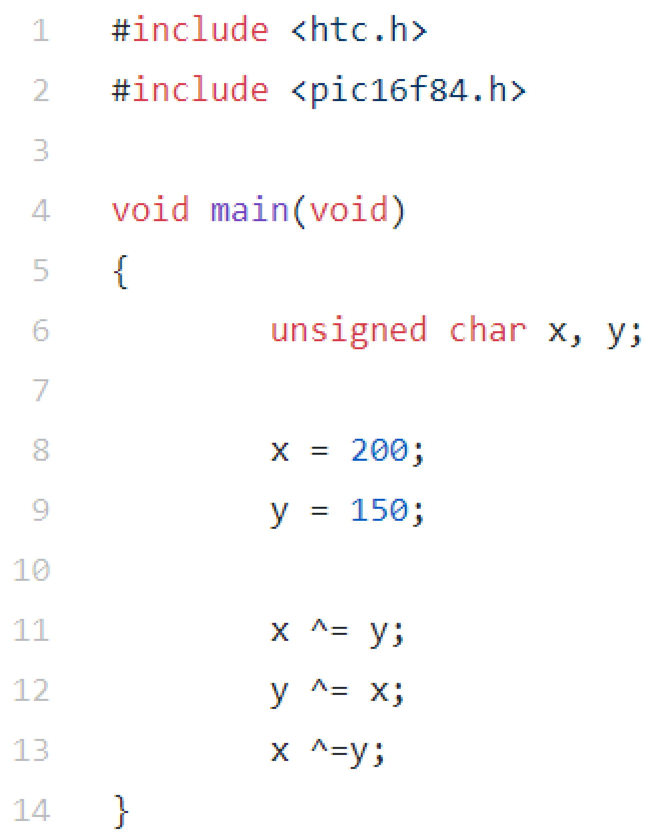
Grupa 5402

12.10.2017

Enunț:

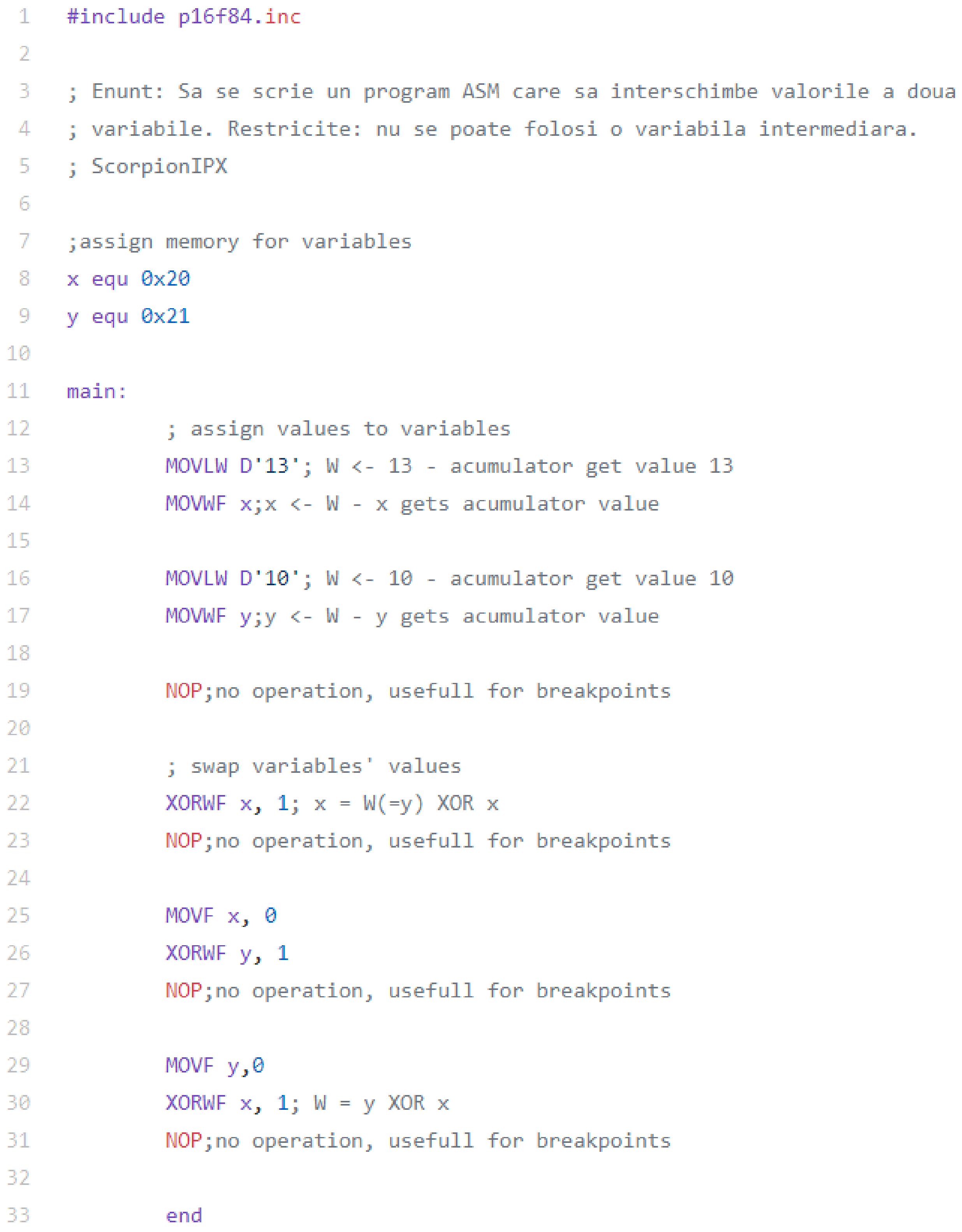
Să se scrie un program care interschimbă valorile a două variabile. Restricții: nu este permisă utilizarea unei variabile ajutătoare.

Implementare C – varianta 1:



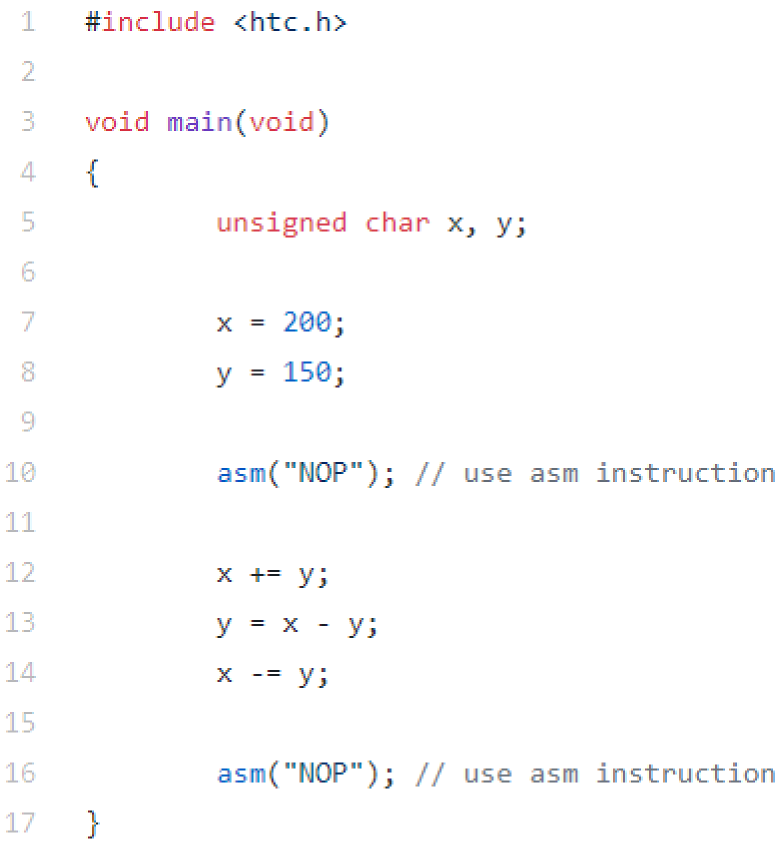
```c
#include <htc.h>
#include <pic16f84.h>

void main(void)
{
        unsigned char x, y;

        x = 200;
        y = 150;

        x ^= y;
        y ^= x;
        x ^=y;
}
```

Implementare ASM – variant 1:



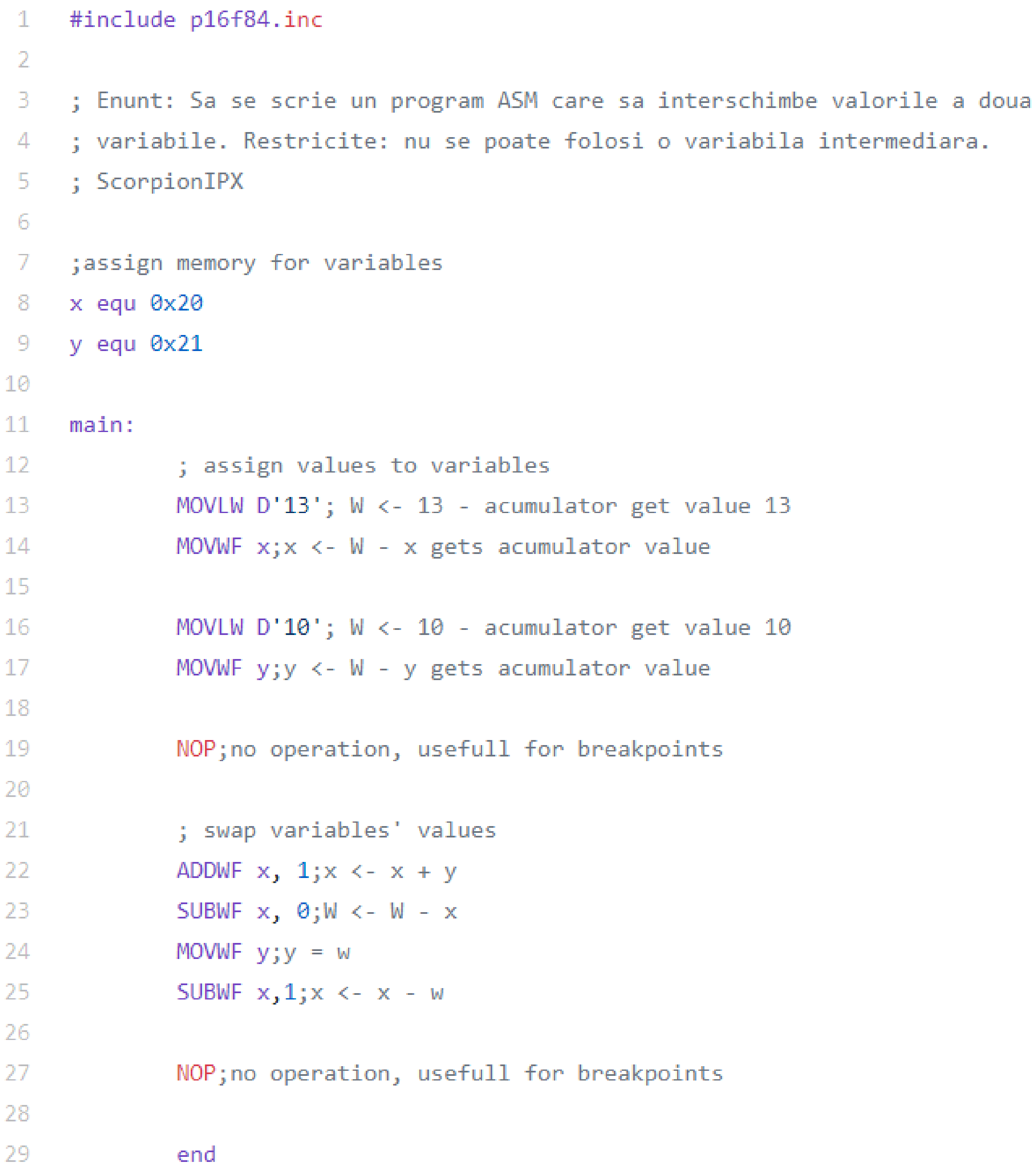
```
#include p16f84.inc

; Enunt: Sa se scrie un program ASM care sa interschimbe valorile a doua
; variabile. Restricite: nu se poate folosi o variabila intermediara.
; ScorpionIPX

;assign memory for variables
x equ 0x20
y equ 0x21

main:
        ; assign values to variables
        MOVLW D'13'; W <- 13 - acumulator get value 13
        MOVWF x;x <- W - x gets acumulator value

        MOVLW D'10'; W <- 10 - acumulator get value 10
        MOVWF y;y <- W - y gets acumulator value

        NOP;no operation, usefull for breakpoints

        ; swap variables' values
        XORWF x, 1; x = W(=y) XOR x
        NOP;no operation, usefull for breakpoints

        MOVF x, 0
        XORWF y, 1
        NOP;no operation, usefull for breakpoints

        MOVF y,0
        XORWF x, 1; W = y XOR x
        NOP;no operation, usefull for breakpoints

        end
```

Implementare C – varianta 2:



```c
#include <htc.h>

void main(void)
{
	unsigned char x, y;

	x = 200;
	y = 150;

	asm("NOP"); // use asm instruction

	x += y;
	y = x - y;
	x -= y;

	asm("NOP"); // use asm instruction
}
```

Implementare ASM – variant 2:



```
#include p16f84.inc

; Enunt: Sa se scrie un program ASM care sa interschimbe valorile a doua
; variabile. Restricite: nu se poate folosi o variabila intermediara.
; ScorpionIPX

;assign memory for variables
x equ 0x20
y equ 0x21

main:
        ; assign values to variables
        MOVLW D'13'; W <- 13 - acumulator get value 13
        MOVWF x;x <- W - x gets acumulator value

        MOVLW D'10'; W <- 10 - acumulator get value 10
        MOVWF y;y <- W - y gets acumulator value

        NOP;no operation, usefull for breakpoints

        ; swap variables' values
        ADDWF x, 1;x <- x + y
        SUBWF x, 0;W <- W - x
        MOVWF y;y = w
        SUBWF x,1;x <- x - w

        NOP;no operation, usefull for breakpoints

        end
```