

NO	INSTRUCTION	CS offsets
1.	MOV BX, 09H	0003H
2.	CALL BX	0005H
3	MOV AX, 0CH	0007H
4.	MOV DX, 0003H	0009H
5.	PUSH DX	000BH
6.	RET	000CH

Assume DS = 5555h, SS = 3200h, CS = 2000h, DI = 3467h, SP = 2766h.

Now, answer all the questions below according to the table.

- Determine the **addressing mode** of instruction No. 6 and after executing instruction No. 6, deduce the **starting physical address** of the next instruction that will be pointed for execution with necessary calculations. [Hint: In usual case **RET** returns the IP to the location where '**CALL**'ed]

Solution:

- After the No.2 call instruction stack was pushed with 0005h for return, then the program jumped to the $CS*10h+09h$ location, $20000h+09h = 20009h$ location [Logical address 2000:0009h],
- As, in the table all the offsets are of CS; therefore, it jumped to the instruction no.4.
- After executing line No.4 and 5. The stack was again pushed with 0003h.
- As, the SP indicates the value 0003h. Finally, while executing the line No.6 for **RET** instruction it was suppose to pop the returning offset from the stack which was initially 0005h but because of the second push it will pop 0003h and will return to $CS*10h+ 0003h = 20003h$ location [Logical address 2000:0005h], Instruction No.1.