Considering a limited instruction set machine as the following:

Registers:

ID	Register	Initial Value	Description
000	A	0x00	General Purpose
001	B	0x00	General Purpose
010	C	0x00	General Purpose
011	D	0x00	General Purpose
100	SP	0xC0	Stack Pointer
101	PC	0xC0	Program Counter

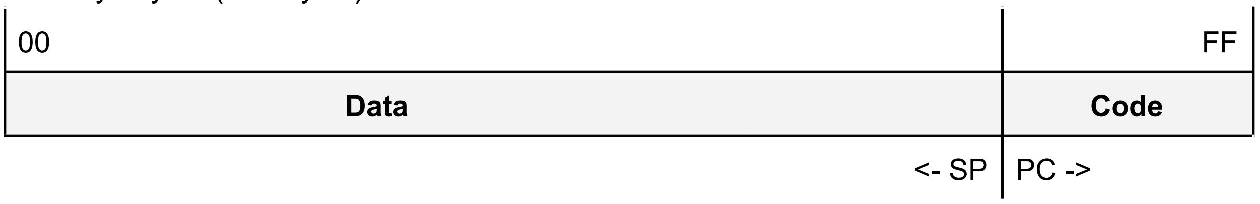
Instructions Set:

Opcode	Mnemonic	Operand 1	Operand 2	Description
00	mov	r	imm8	Move immediate byte to register
01	mov	r	[m]	Move memory address value to register
02	mov	[m]	r	Move register value to memory address
03	push	imm8		Push immediate byte to stack
04	pop	r		Pop byte from stack to register
05	add	r	imm8	Add immediate byte value to register
06	dec	r	imm8	Decrease register by immediate byte
07	prn	[m]		Print string at memory address

Instruction Format:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Op code								**Operand 1**								**Operand 2**							

Memory Layout (256 Bytes):



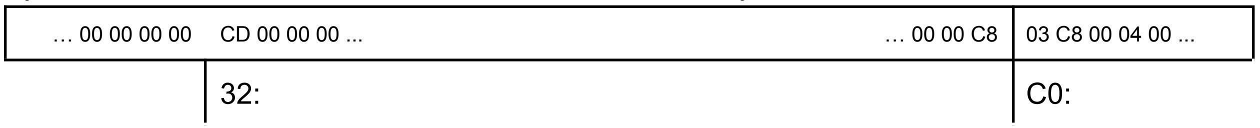
So for example, to push 200 to the stack, pop it into A register, increase it by 5 then write it in address 50, we code the following instructions:

```
C0: 03 C8 00 push 200
C3: 04 00 00 pop A
C6: 05 00 05 add A, 5
C9: 02 00 32 mov [50], A
```

Raw bytes of the code will be:

```
03C800040000050005020032
```

By the end of the execution of the last command, memory should look like:



And the Registers status will be:

Register	Value
A	0xCD
B	0x00
C	0x00
D	0x00
SP	0xC0
PC	0xCC

Answer the following -in a report that includes the source code with-:

1. Write a C or Python code that implements the machine, reads code from a file, loads the memory with code and starts executing, stopping the execution after the first prn instruction.
2. Write an assembly code that writes a null-terminated ascii string to memory using stack operations, then print the string to the console.
3. What's wrong with the following code:

```
push 200
push A
add SP, 20
push 259
push 243
dec SP, 20
push 80
```

4. Disassemble the following code:

```
00006403C80004010003FA000102BF040300050301
```

5. What are the registers values after the execution of the previous code?

All the best!