

Practical exercise 9

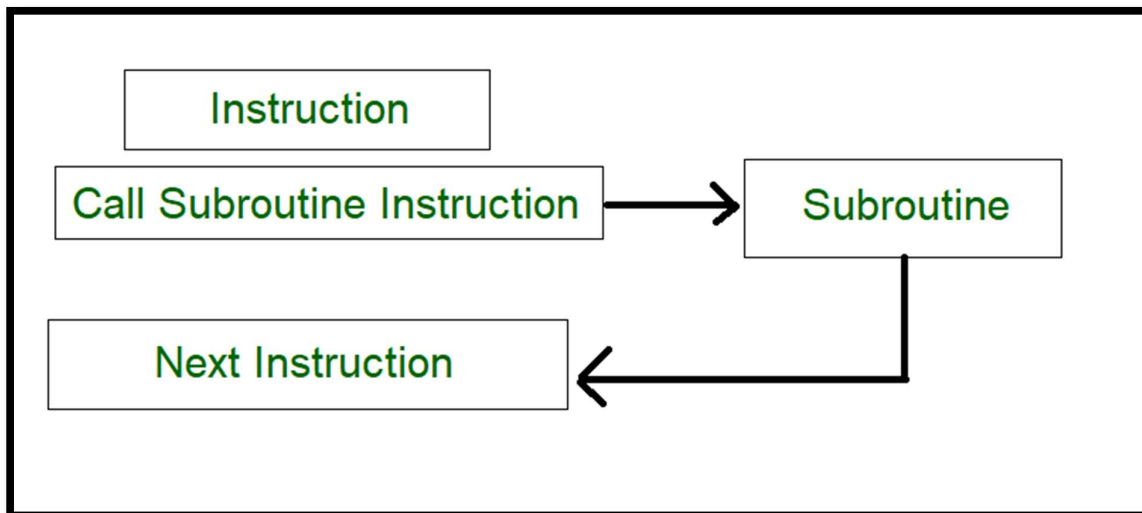
Aim: Factorial of 8-bit number using subroutine.

Theory:

Subroutine

A **set of instructions** that are used repeatedly in a program can be referred to as a Subroutine. Only one copy of this Instruction is stored in the memory. When a Subroutine is required, it can be called many times during the Execution of a particular program. A call Subroutine Instruction calls the Subroutine. Care Should be taken while returning a Subroutine as a Subroutine can be called from a different place from the memory.

The content of the PC must be Saved by the call Subroutine Instruction to make a correct return to the calling program.



Advantages of Subroutines

- **Code reuse:** Subroutines can be reused in multiple parts of a program, which can save time and reduce the amount of code that needs to be written.

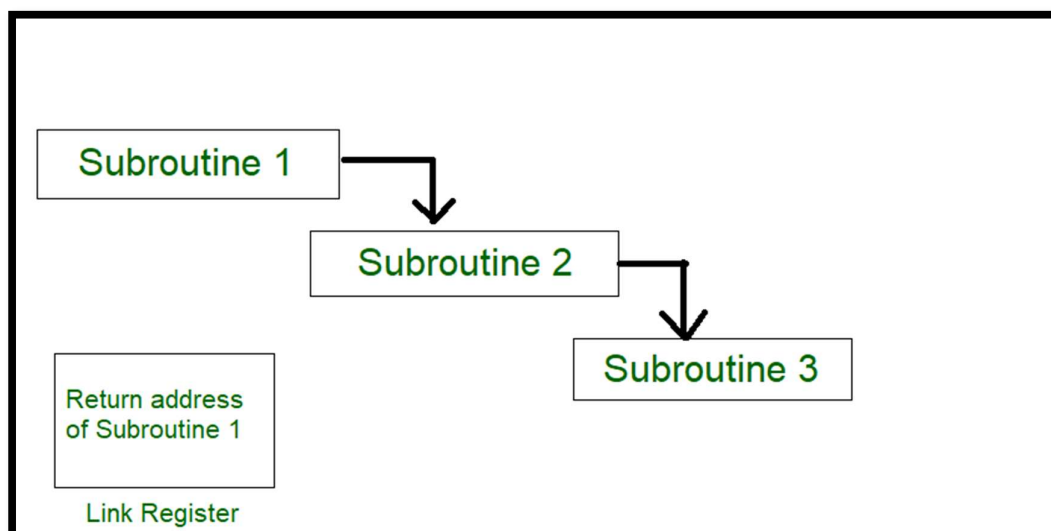
- **Modularity:** Subroutines help to break complex programs into smaller, more manageable parts, making them easier to understand, maintain, and modify.
- **Encapsulation:** Subroutines provide a way to encapsulate functionality, hiding the implementation details from other parts of the program.

Disadvantages of Subroutines

- **Overhead:** Calling a subroutine can incur some overhead, such as the time and memory required to push and pop data on the stack.
- **Complexity:** Subroutine nesting can make programs more complex and difficult to understand, particularly if the nesting is deep or the control flow is complicated.
- **Side Effects:** Subroutines can have unintended side effects, such as modifying global variables or changing the state of the program, which can make debugging and testing more difficult.

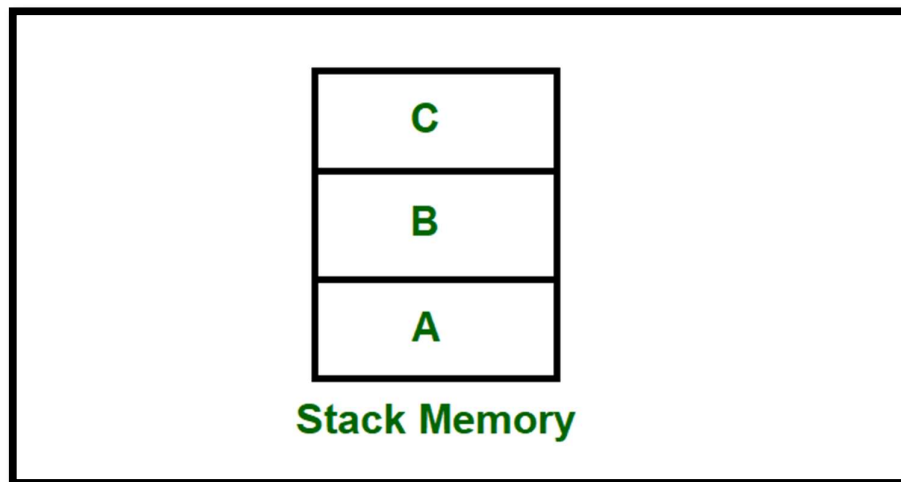
Subroutine Nesting

Subroutine nesting is a common Programming practice In which one Subroutine calls another Subroutine.



Stack Memory

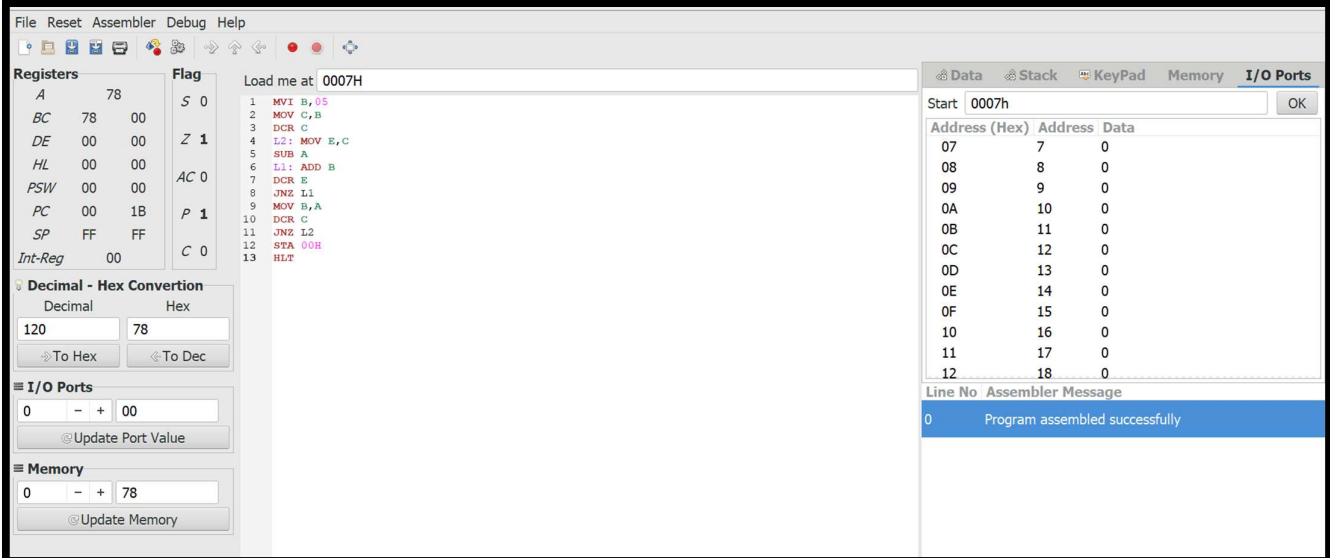
A Stack is a basic [data structure](#) that can be implemented anywhere in the memory. It can be used to store variables that may be required afterwards in the program Execution. In a stack, the first data put will be the last to get out of a stack. So, the last data added will be the first one to come out of the stack ([last in first out](#)).



- Program to find factorial of 8-bit number using subroutines.

```
MVI B,05  
MOV C, B  
DCR C  
L2: MOV E, C  
SUB A  
L1: ADD B  
DCR E  
JNZ L1  
MOV B, A
```

```
DCR C
JNZ L2
STA 00H
HLT
```



Conclusion: Thus, we have implemented a program to find the factorial of number using 8085 instructions and subroutines.