**Internal working of C program**

**#include<stdio.h>**

int addition(int,int);

int main()

{

int val1=0; //val1 is a local variable so its memory location on stack

int val2=0; //val2 is a local variable so its memory location on stack

int res=0; //res is a local variable so its memory location on stack

printf("Enter first no."); //8

scanf("%d",&val1); //value assign to val1 variable

printf("\nenter 2nd no."); //10

scanf("%d",&val2); //value assign to val2 variable

sum=addition(val1,val2); //we call addition function with passing variable and assign to sum

printf("sum is: %d", res);

return 0;

}

int addition(int no1,int no2) //separate addition function.

{

int ans=0; //local variable

ans=x+y; // logic of addition

return ans; //return ans variable

}

**To understand internal working of program we have to draw stack layout because all function get executed in specific order. And each function separate stack frame is created.**

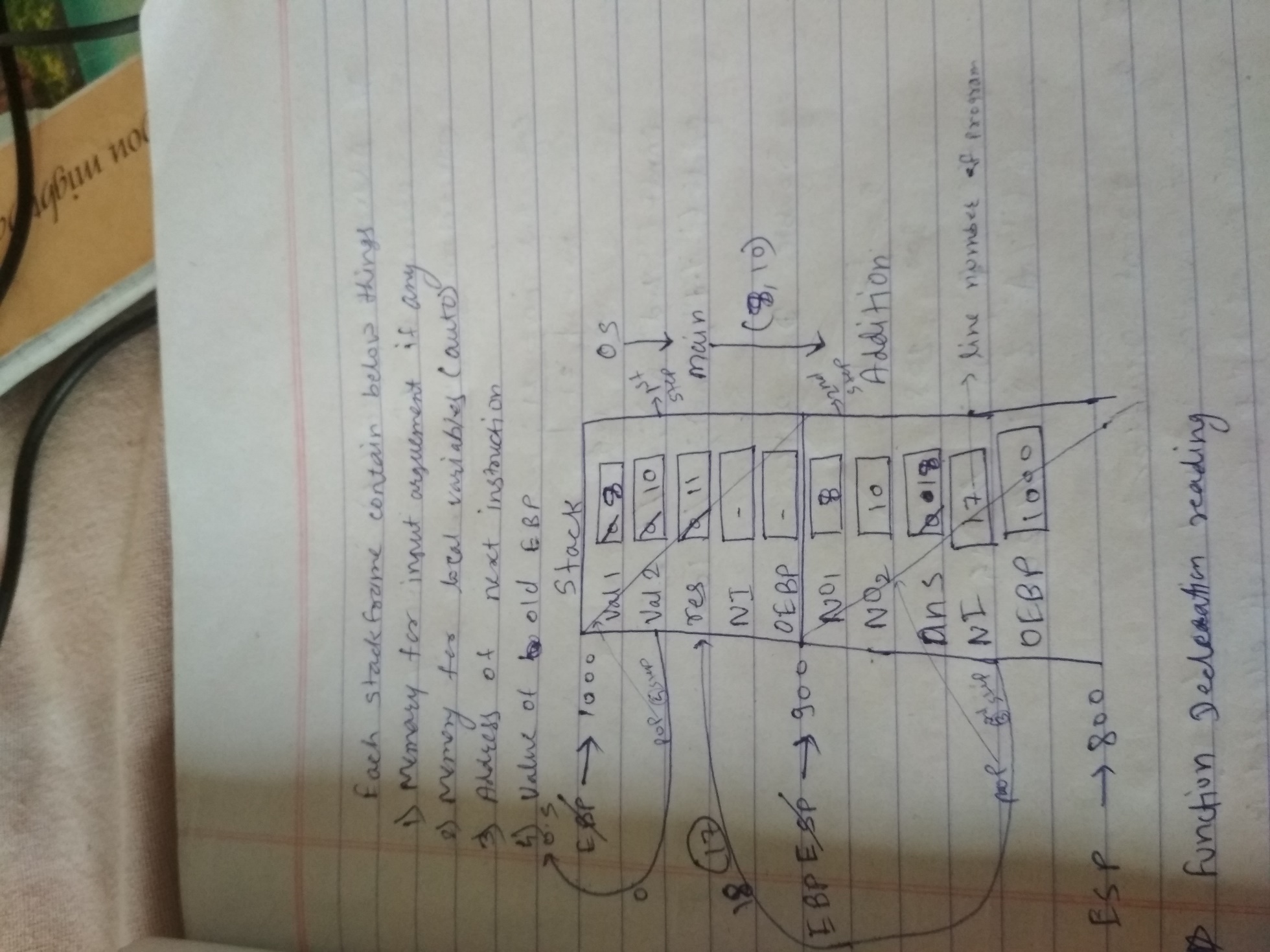
**Each stack frame contains 4 arguments as follows:**

1. Memory for input argument
2. Memory for local variables
3. Address of next instruction
4. Value of old EBP

EBP- It is pointer register which contains base address of stack frame

ESP- It is pointer register which contains end point address of stack frame

**How stack frame looks like**

****