

## Unit II : embedded C programming Sample Program

**//addition**

**// write a program in embedded C to add two 8 bit numbers**

```
#include<reg51.h> // initialize preprocessor directive
```

```
void main ()          // main function
```

```
{                      //start of main function
```

```
//Declare three variables//
```

```
unsigned char x,y,z;
```

```
//Assign values to integers
```

```
x=0x34;
```

```
y=0x54;
```

```
//Now declare port0 as output port
```

```
P0=0x00;
```

```
//Now perform addition
```

```
z=x+y;
```

```
//Now give values of z to ports
```

```
P0=z;
```

```
}
```

**//addition**

```
#include <reg51.h>
```

```
unsigned char result1;
```

```
sfr m = 0xE0;
```

```
sfr n = 0xF0;
```

```
sfr port1= 0X90;
```

```
void main (void)
```

```
{
```

```
    unsigned char result;
```

```
    m = 0x21;          // First number 0X indicate hex data
```

```
    n = 0x71;          // Second number 0X indicate hex data
```

```
    result = m + n;     // Add two numbers
```

```
    port1 = result;     // Display result on P1
```

```
}
```

### Write a program to multiply two 8 bit Numbers

```
// Multiplication :
#include <reg51.h>
    sfr a = 0xE0;           // Define a as an accumulator
    sfr b = 0xF0;           // Define b as B Register
    sfr port0 = 0x80;       // Define P0 as RESULT
    void main (void)
    {
        unsigned char x;
        a = 0x05;           // First number
        b = 0x09;           // Second number
        x = a*b;             // Multiply first and second numbers
        port0 = x;           // Display result
    }
```



### // Division

```
#include <reg51.h>
sfr a = 0xE0;           // Define i as an accumulator
sfr b = 0xF0;           // Define j as B Register
sfr Quotient = 0x80;    // Define P1 as Quotient
sfr remainder = 0x90;   // Define P2 as Remainder

void main (void)
{
    //unsigned char x, y;
    a = 27;             // First number
    b = 5;              // Second number
    Quotient = a/b;      // Divide a by b and store quotient
    remainder = a%b;     // Store remainder
    remainder = remainder+1;
}
```

### Logical operations

**// Bitwise AND write a program to bitwise AND two 8 bit numbers**

```
#include <reg51.h>
sfr a = 0xE0;
sfr b = 0xF0;
sfr port0 = 0x80;
sfr port1 = 0x90;

void main()
{
    unsigned char result;
    a=0x40;
    b=0x45;
    result=a&b;
    port0=result;
}
```

### **// Bitwise OR**

```
#include <reg51.h>
sfr a = 0xE0;
sfr b = 0xF0;
sfr port0 = 0x80;
sfr port1 = 0x90;

void main()
{
    unsigned char result;
    a=0x40;
    b=0x45;
    result=a|b;
    port0=result;
}
```

### **// Logical shift right and shift left**

```
#include <reg51.h>
sfr a = 0xE0;
sfr b = 0xF0;
sfr port0 = 0x80;
sfr port1 = 0x90;

void main()
{
    unsigned char result,result1;
    a=0x40; //0100 0000
    b=0x08; //0100 0101
    result=a>>2;
    result1 =b<<2;

    port0=result;
    port1=result1;
}
```