```
//1.Adding Two Hex Numbers
#include <xc.h>
void main(void) {
    static int num1=3, num2=4, sum;
    sum=num1+num2;
TRISB=0;
PORTB=sum;

return;
}

//2.add array of n Hex numbers.
#include <xc.h>
#define N 10
void main(void)
{
        int numbers[N] = {0x01, 0x23, 0x45, 0x67, 0x89, 0xAB, 0xCD, 0xEF, 0x11,0x22};
        int sum=0;
        int i;
        for(i=0;i<N;i++)
        {
                sum = sum + numbers[i];
        }
        TRISB=0;
        PORTB=sum;

return;
}

//3.transfer element from one location to another
//Exchange source and destination array elements
#include <xc.h>
void main(void)
{
        int source1[] = { 0x21,0x22, 0x23, 0x24, 0x25 };
        int dest[] = {0x99, 0x99,0x99,0x99,0x99};
        int i,temp;
        for(i=0;i<5;i++)
        {
                temp = source1[i];
                source1[i] = dest[i];
                dest[i] = temp;
        }
        TRISB=0;
        PORTB=dest;

        return;
}
//Move source elements to destination
#include <xc.h>
void main(void)
{
        int source1[] = { 0x21,0x22, 0x23, 0x24, 0x25 };
        int dest[] = {0x00, 0x00,0x00,0x00,0x00};
        int i;
        for(i=0;i<5;i++)
        {
                dest[i] = source1[i];
        }

        TRISB=0;
        PORTB=dest;

        return;
}

//4.Write an Embedded C menu driven program
//Multiply two hex 8 bit numbers:
#include <xc.h>
```

```
//Multiply two hex 8 bit numbers:
void main(void)
{
        unsigned char num1, num2;
        unsigned int result;

        // Get input from user
        num1 = 0x02; // Example input value
        num2 = 0x04; // Example input value

        // Perform multiplication
        result = num1 * num2;

        // Display result
        TRISD=0;
        PORTD = (unsigned char) result;
}

//multiplying two 8-bit Hex numbers using successive addition method::
#include <xc.h>

void main(void)
{
        unsigned char num1, num2;
        unsigned int result = 0;
        unsigned char i;

        // Get input from user
        num1 = 0x02; // Example input value
        num2 = 0x04; // Example input value

        // Perform multiplication using successive addition
        for(i = 0; i < num2; i++)
        {
                result += num1;
        }

        // Display result
        TRISC=0;
        TRISD=0;
        PORTD = (unsigned char) result;
        PORTC = (unsigned char) (result >> 8);
        return;
}

//Divide 8 bit number by 8 bit number
#include <xc.h>

void main(void)
{
        unsigned char num1, num2;
        unsigned char quotient;

        // Get input from user
        num1 = 0x06; // Example input value
        num2 = 0x03; // Example input value

        // Perform division
        quotient = num1 / num2;

        // Display quotient
        TRISD=0;
        PORTD = quotient;
        return;
}

//dividing two 8-bit hex numbers using successive subtraction method:
#include <xc.h>
```

```
void main(void)
{
        unsigned char num1, num2;
        unsigned char quotient;

        // Get input from user
        num1 = 0x06; // Example input value
        num2 = 0x02; // Example input value

        // Perform division using successive subtraction
        while(num1 >= num2)
        {
                num1 -= num2;
                quotient++;
        }
        // Display quotient
        TRISD=0;
        PORTD = quotient;
        return;
}

//5.Sorting the numbers in ascending and descending order
#include <xc.h>
// Function to swap two integers
void swap(int *a, int *b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}
// Function to perform bubble sort
void bubbleSortAscending(int arr[], int n)
{
    for (int i = 0; i < n-1; i++) {
        for (int j = 0; j < n-i-1; j++) {
             if (arr[j] > arr[j+1]) {
                swap(&arr[j], &arr[j+1]);
            }
        }
    }
}
int main(void)
{
    // Declare and initialize the array
    int numbers[] = {5, 2, 8, 1, 7};
    int size = sizeof(numbers) / sizeof(numbers[0]);
    // Sort in ascending order
    bubbleSortAscending(numbers,size);
    while (1){
    }
    TRISB=0;
    PORTB=numbers;
    return 0;
}
```