Lab Practice Week 11
To be Submitted as Assembler Exercise by Week 12

Internal Students: You need to show a working version of your solutions to program 24, 25 and 26. Your tutor will expect to see by the time of your lab class in week 12 as part of your Assembler assessment.

External Students: Please email your program 23, 24 and 25 to your tutor. Your tutor will expect to receive them by the end of week 12.

**Program 24**

Compile and run the program in the given example. Modify the program by shifting the actual subtraction calculation into two separate functions with function declaration:

Void subtractInt(int input1, int input2, int *result)
Void subtractShort(short input1, short input2, short *result)

**Program 25**

Complete the following program using inline assembler:

```
#include <stdio.h>

int main() {
    const int SIZE = 5;
    int inputs[SIZE];
    int i = 0;
    int result;
    for (i = 0; i < SIZE; i ++)
    {
        printf("Please enter an integer number --> ");
        scanf("%d", &inputs[i]);
        fflush(stdin);
    }
    // Your job is to write a LOOP using
    // inline assembler to calculate the
    // sums of the input array
    _asm{
        ; ...
    }
    printf("Sum of the numbers is --> %d\n", result);
    return 0;
}
```
Program 26

Write a program in C using in-line assembly code that does the following:

(i) input two positive integer numbers from the keyboard,
(ii) find the product of the two numbers,
(iii) divide the first number by the second number to get the quotient and the remainder
(iv) output the results of multiplication and division.

The program is to consist of three functions - main, Multiply and Divide. The main function can be written entirely in C. It will input two positive integer numbers using the library function `scanf`, call the functions Multiply and Divide and then output the results (using `printf` statements). The two numbers are to be passed as arguments to the function calls for Multiply and Divide.

The function Multiply will take two integer numbers as parameters, multiply them and return the product. The calculation of product is to be done using in-line assembly statements.

The function Divide will take four integer parameters, divide the first integer by the second integer, and pass out the quotient and the remainder to the calling function via the third and fourth parameters. The division is to be performed using in-line assembly statements.
Example on using inline assembler in Visual C

```c
#include <stdio.h>
/**
 * A simple input program for demonstrating inline assembler
 * in Visual C++.
 * NOTE: This program will ONLY compile in Visual C++.
 **/

int main() {
    int intVar1, intVar2, intResult;
    short shortVar1, shortVar2, shortResult;

    printf("Please enter an integer number --> ");
    scanf("%d", &intVar1);

    printf("Please enter another integer number --> ");
    scanf("%d", &intVar2);

    printf("Please enter an integer number --> ");
    scanf("%d", &shortVar1);

    printf("Please enter another integer number --> ");
    scanf("%d", &shortVar2);

    _asm{
        mov ax, shortVar2 ; Assign shortVar2 to ax register
        mov dx, shortVar1
        sub dx, ax
        mov shortResult, dx
    }

    printf("The result for the short number subtraction is --> %d\n", shortResult);

    // Extended register has to be used for storing 32 bit integer.
    _asm{
        mov eax, intVar2
        mov edx, intVar1
        sub edx, eax
        mov intResult, edx
    }

    printf("The result for the integer number subtraction is --> %d\n", intResult);
    return 0;
}
```