

PROJECT 1

PART I - HLL

PART II - AL

**CSCI 150
ASSEMBLY LANGUAGE**

MAI PHAM

**DEVELOPMENT ENVIRONMENT
VISUAL STUDIO 2012**

TABLE OF CONTENTS



Project Note



Output



Source Code (Binary.cpp & AddSub.asm)

PROJECT NOTE

OBJECTIVE:

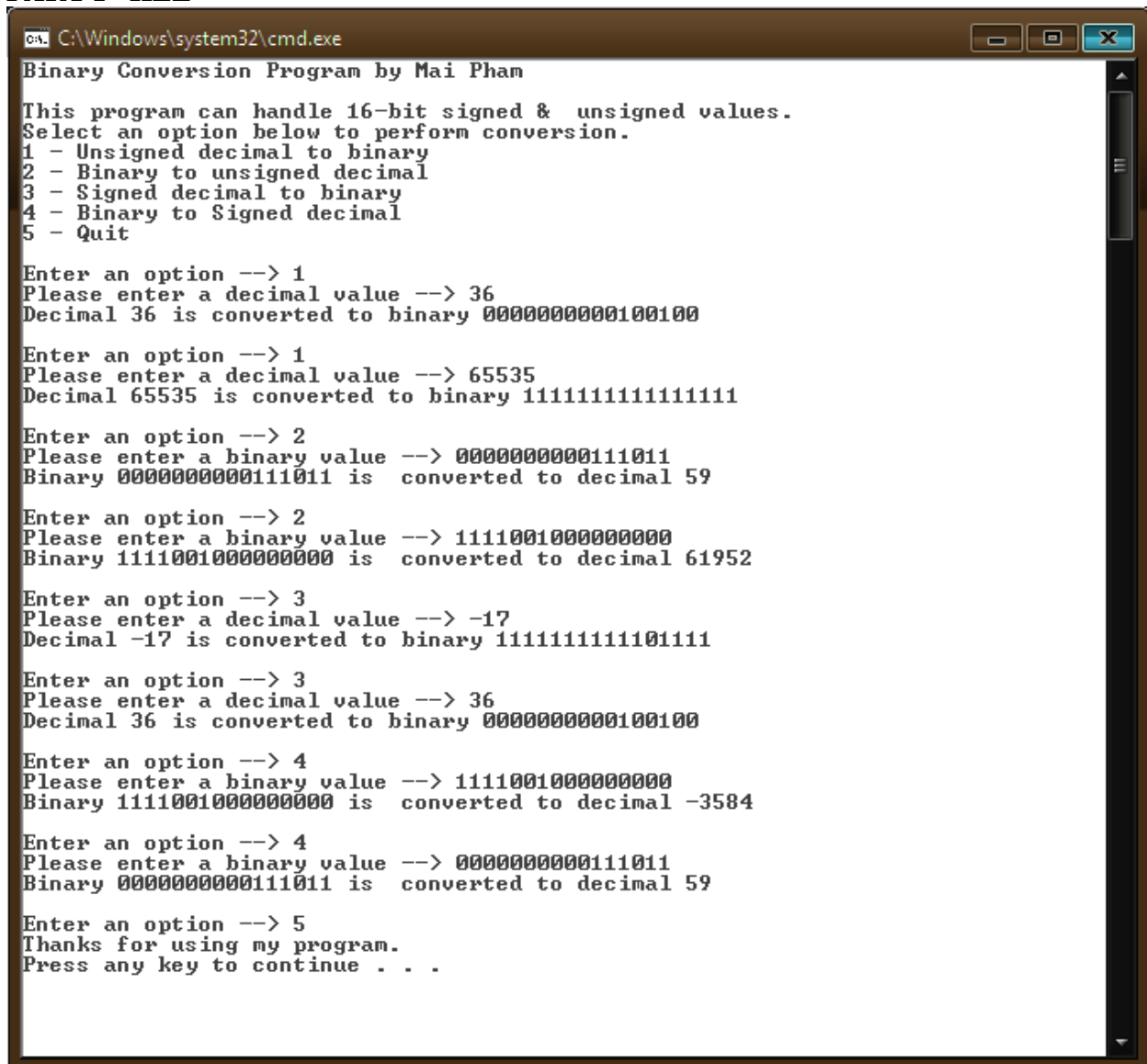
- ✚ Part A: Write a conversion program for 16 bit signed and unsigned decimal to binary and vice versa.
- ✚ Part B: Write an assembly program that implements two arithmetic expressions

SUMMARY/EXTRA CREDIT:

- ✚ I didn't encounter much problems with Part A and Part B since the Lab 1 to 3 does help with most of the coding, including for the extra credit 1 and 2. Overall, my project is completed and successfully run both parts of the project and the extra credits.

OUTPUT

PART I - HLL



```
C:\Windows\system32\cmd.exe
Binary Conversion Program by Mai Pham

This program can handle 16-bit signed & unsigned values.
Select an option below to perform conversion.
1 - Unsigned decimal to binary
2 - Binary to unsigned decimal
3 - Signed decimal to binary
4 - Binary to Signed decimal
5 - Quit

Enter an option --> 1
Please enter a decimal value --> 36
Decimal 36 is converted to binary 0000000000100100

Enter an option --> 1
Please enter a decimal value --> 65535
Decimal 65535 is converted to binary 111111111111111

Enter an option --> 2
Please enter a binary value --> 000000000011011
Binary 000000000011011 is converted to decimal 59

Enter an option --> 2
Please enter a binary value --> 1111001000000000
Binary 1111001000000000 is converted to decimal 61952

Enter an option --> 3
Please enter a decimal value --> -17
Decimal -17 is converted to binary 11111111110111

Enter an option --> 3
Please enter a decimal value --> 36
Decimal 36 is converted to binary 0000000000100100

Enter an option --> 4
Please enter a binary value --> 1111001000000000
Binary 1111001000000000 is converted to decimal -3584

Enter an option --> 4
Please enter a binary value --> 000000000011011
Binary 000000000011011 is converted to decimal 59

Enter an option --> 5
Thanks for using my program.
Press any key to continue . . .
```

PART II - AL

```

C:\Windows\system32\cmd.exe
Author: Mai Pham
Part 1: r1 = -(v2 - (v3 + v1)) + 1

EAX=00000014 EBX=7FFD6000 ECX=00000000 EDX=00405027
ESI=00000000 EDI=00000000 EBP=0012FF94 ESP=0012FF8C
EIP=0040102E EFL=00000202 CF=0 SF=0 ZF=0 OF=0 AF=0 PF=0

EAX=00000024 EBX=7FFD6000 ECX=00000000 EDX=00405027
ESI=00000000 EDI=00000000 EBP=0012FF94 ESP=0012FF8C
EIP=00401039 EFL=00000206 CF=0 SF=0 ZF=0 OF=0 AF=0 PF=1

EAX=FFFFFFF5 EBX=7FFD6000 ECX=00000000 EDX=00405027
ESI=00000000 EDI=00000000 EBP=0012FF94 ESP=0012FF8C
EIP=00401056 EFL=00000286 CF=0 SF=1 ZF=0 OF=0 AF=0 PF=1

r1 = -11
Part 2: r2 = v4 + 10h

EAX=00000000 EBX=7FFD6000 ECX=00000000 EDX=0040504C
ESI=00000000 EDI=00000000 EBP=0012FF94 ESP=0012FF8C
EIP=00401088 EFL=00000202 CF=0 SF=0 ZF=0 OF=0 AF=0 PF=0

EAX=0000000F EBX=7FFD6000 ECX=00000000 EDX=0040504C
ESI=00000000 EDI=00000000 EBP=0012FF94 ESP=0012FF8C
EIP=0040109B EFL=00000207 CF=1 SF=0 ZF=0 OF=0 AF=0 PF=1

r2 = 15
Press any key to continue . . .

```

SOURCE CODE

PART I – HLL

```

#include <iostream>
#include <string>
using namespace std;

string decToBinary (int dec);
int binary2Decimal(string b);
void addOne(string &b);
void flipBinary(string &b);

int main()
{
    /* TESTING
    string a = "0";
    a[0]++;
    cout << a << endl;           // output: 1
    string b = "1";
    b[0]++;
    cout << b << endl;           // output: 2
    int dec = -45;
    cout << dec*-1 << endl;       // output: 45
    */

    int option;
    string binary;

```

```

int decimal;

cout << "Binary Conversion Program by Mai Pham\n\n";
cout << "This program can handle 16-bit signed & unsigned values.\n";
cout << "Select an option below to perform conversion.\n";
cout << "1 - Unsigned decimal to binary\n";
cout << "2 - Binary to unsigned decimal\n";
cout << "3 - Signed decimal to binary\n";
cout << "4 - Binary to Signed decimal\n";
cout << "5 - Quit\n\n";

cout << "Enter an option --> ";
cin >> option;
while (option > 0 && option < 5)
{
    if (option == 1)    {
        cout << "Please enter a decimal value --> ";
        cin >> decimal;
        cout << "Decimal " << decimal << " is converted to binary " <<
decToBinary (decimal) << endl;
    }
    if (option == 2)    {
        cout << "Please enter a binary value --> ";
        cin >> binary;
        cout << "Binary " << binary << " is " << " converted to decimal "
<<binary2Decimal(binary) << endl;
    }
    if (option == 3)    {
        cout << "Please enter a decimal value --> ";
        cin >> decimal;
        if (decimal >= 0)
            binary = decToBinary (decimal);
        if (decimal < 0)    {
            binary = decToBinary ((decimal*-1));
            flipBinary(binary);
            addOne(binary);
        }
        cout << "Decimal " << decimal << " is converted to binary " <<
binary << endl;
    }
    if (option == 4)    {
        cout << "Please enter a binary value --> ";
        cin >> binary;
        if (binary[0] == '0')
            cout << "Binary " << binary << " is " << " converted to
decimal " <<binary2Decimal(binary) << endl;
        else if (binary[0] == '1') {
            string tempBinary;
            tempBinary = binary;
            flipBinary(tempBinary);
            addOne(tempBinary);
            cout << "Binary " << binary << " is " << " converted to
decimal " << (0-binary2Decimal(tempBinary)) << endl;
        }
    }

    cout << "\nEnter an option --> ";
    cin >> option;
}

```

```

        cout << "Thanks for using my program.\n";
        return 0;
    }

    string decToBinary (int dec)    {
        string binary = " ";
        char c;
        while (dec != 0)    {
            c = '0' + dec % 2;
            binary = c + binary;
            dec/=2;
        }
        while (binary.length() <= 16)    {
            binary = "0" + binary;
        }

        return binary;
    }

    void flipBinary(string &b) {
        for (int i = 0; i < 16; i++)
            if (b[i] == '0')
                b[i] = '1';
            else
                b[i] = '0';
    }

    void addOne(string &b)    {
        int remainder = 0;
        int i = 15;

        b[i]++;
        while (b[i] == '2') {
            remainder = 1;
            b[i] = 0;
            i--;
            b[i] += remainder;
        }
    }

    int binary2Decimal(string b)    {
        int decimal = 0;
        int j = 0;
        for (int i = 15; i >= 0; i--)    {
            if (b[i] == '1')
                decimal += pow(2,j);
            j++;
        }
        return decimal;
    }
}

```

PART II - AL

TITLE ASM Template
INCLUDE Irvine32.inc

(template.asm)

```

.data
v1    SDWORD 10h
v2    SDWORD 30h
v3    SDWORD 20

```

```

r1    SDWORD ?
v4    WORD    0FFFFh
r2    WORD    ?
author BYTE    "Author: Mai Pham", 0Dh, 0Ah, 0
p1    BYTE    "Part 1: r1 = -(v2 - (v3 + v1)) + 1", 0Dh, 0Ah, 0
p2    BYTE    "Part 2: r2 = v4 + 10h", 0Dh, 0Ah, 0
ans1  BYTE    "r1 = ", 0
ans2  BYTE    "r2 = ", 0

```

.code

```

main PROC
    mov     edx, OFFSET author    ; display name
    call WriteString              ; display a null-terminated string.

    mov     edx, OFFSET p1       ; display part 1
    call WriteString              ; display a null-terminated string.

    ;r1 = -(v2 - (v3 + v1)) + 1
    mov     eax, v3              ; eax = 14h (20d)
    call DumpRegs                ; display the registers

    add     eax, v1              ; 14h add 10h (20d + 16d), eax = 24h (36d)
    call DumpRegs                ; display the registers

    sub     v2, eax              ; 30h - 24h (48d - 36d), v2 = Ch (12d)
    neg     v2                   ; v2 = FFF4h (-12d)
    add     v2, 1                ; v2 = FFF5h (-11d)
    mov     eax, v2              ; eax = FFF5h
    call DumpRegs                ; display the registers
    mov     r1, eax

    mov     edx, OFFSET ans1 ; display answer r1
    call WriteString
    call WriteInt                ; 32 bit signed number
    call crlf                    ; enter
    call crlf                    ; enter

    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

    mov     edx, OFFSET p2       ; display name
    call WriteString              ; display a null-terminated string.

    ;r2 = v4 + 10h
    mov     eax, 0                ; clear eax
    call DumpRegs

    add     v4, 10h              ; 0FFFFh add 10h (65535d + 16d), v4 = 1000Fh (65551d)
    mov     ax, v4                ; ax = 000Fh (15d)
    call DumpRegs
    mov     r2, ax                ; r2 = 000Fh (15d)

    mov     edx, OFFSET ans2 ; display answer r2
    call WriteString
    call WriteDec                 ; 32 bit unsigned number
    call crlf

    exit
main ENDP
END main

```