Code:

Lab4_main.cpp

```
#include <cstdio>
#include <iostream>
#define N 210
typedef unsigned char byte;

using namespace std;

extern "C" {

    void ShowN(byte* p1, long int p2);

    bool Bigne(byte* arr1, byte* arr2, long int length);

    void printNum(byte num)
    {
        printf("%.2x", num);
    }

    void printSpace()
    {
        printf(" ");
    }

    void printNewLine()
    {
        printf("\n");
    }
}

void main() {
    byte x[N], y[N], z[N];
    byte result;

    for (int i = 0; i < N; i++)
    {
        x[i] = i;
        y[i] = i;
        z[i] = i / 2;
    }
    y[N - 3] = 0x10;

    printf("X=\n");
    ShowN(&x[0], N);
    printf("\n\nY=\n");
    ShowN(&y[0], N);
    printf("\n\nZ=\n");
```

```
    ShowN(&z[0], N);

    printf("\n\n(X != Y) is %d\n", Bigne(&x[0], &y[0], N));
    printf("(X != Z) is %d\n", Bigne(&x[0], &z[0], N));
}

 != Z) is %d\n", Bigne(&x[0], &z[0], N));
}
```

Lab4.asm:

```
.686
.MODEL FLAT, C

.DATA
    len dd 0
    i dd 0
    j dd 0

.CODE

Bigne PROC @arr1:dword, @arr2:dword,  @length:dword

    mov esi, @arr1
    mov edi, @arr2
    mov eax, @length
    mov [len], eax
    mov edx, 0
    mov ebx, 4

    div ebx
    mov i, eax
    mov j, edx

    mov eax, 0
    @loopStart:
    mov edx, dword ptr [esi + 4*eax]
    mov ecx, dword ptr [edi + 4*eax]

    @loopContinue:
    cmp edx, ecx
    jne @DifferentBytes

    inc eax
    mov ebx, [i]
    cmp eax, ebx
    jne @loopStart

    mov ebx, 4
```

```
    mul ebx
    add esi, eax
    add edi, eax
    mov eax, 0
    @loopStart2:
    mov dl, byte ptr [esi + eax]
    mov dh, byte ptr [edi + eax]

    @loopContinue2:
    cmp dh, dl
    jne @DifferentBytes

    inc eax
    mov ebx, [j]
    cmp ebx, eax
    jne @loopStart2


    mov al, 0

    ret

    @DifferentBytes:
    mov al, 1

    ret
Bigne ENDP

END


BigShowN.asm:
.686
.model flat,C

.data
    index db 0

.const

.code
    printNum proto num:byte
    printSpace proto
    printNewLine proto

ShowN proc @mas:dword, @len:dword

    mov ebx, @mas
```

```
    mov ecx, @len

    xor eax, eax

    mov al, index

    @loop:
        invoke printNum, byte ptr [ebx + eax]
        add index, 1
        mov al, index
        push ebx
        mov ebx, 4
        xor edx, edx
        div ebx
        cmp edx, 0
        je @loopSpace
    @loopContinue1:
        mov al, index
        mov ebx, 32
        xor edx, edx
        div ebx
        cmp edx, 0
        je @loopNewLine
    @loopContinue2:
        pop ebx
        mov al, index
        cmp eax, @len
        jne @loop
        jmp @loopEnd

    @LoopSpace:
        invoke printSpace
        jmp @loopContinue1

    @loopNewLine:
        invoke printNewLine
        jmp @loopContinue2

    @loopEnd:
    mov index, 0
    ret
ShowN endp
end
```

How it works:

```
void ShowN(byte* p1, long int p2);

bool Bigne(byte* arr1, byte* arr2, long int length); //

void printNum(byte num)
{
    printf("%.2x", num);
}

void printSpace()
{
    printf(" ");
}

void printNewLine()
{
    printf("\n");
}
```

There are 5 functions; the first two are for importing from assembly code, and the last three are for importing into assembly code.

Debbuging:

```
X=
00010203 04050607 08090a0b 0c0d0e0f 10111213 14151617 18191a1b 1c1d1e1f
20212223 24252627 28292a2b 2c2d2e2f 30313233 34353637 38393a3b 3c3d3e3f
40414243 44454647 48494a4b 4c4d4e4f 50515253 54555657 58595a5b 5c5d5e5f
60616263 64656667 68696a6b 6c6d6e6f 70717273 74757677 78797a7b 7c7d7e7f
80818283 84858687 88898a8b 8c8d8e8f 90919293 94959697 98999a9b 9c9d9e9f
a0a1a2a3 a4a5a6a7 a8a9aaab acadaeaf b0b1b2b3 b4b5b6b7 b8b9babb bcbdbebf
c0c1c2c3 c4c5c6c7 c8c9cacb cccdcecf d0d1

Y=
00010203 04050607 08090a0b 0c0d0e0f 10111213 14151617 18191a1b 1c1d1e1f
20212223 24252627 28292a2b 2c2d2e2f 30313233 34353637 38393a3b 3c3d3e3f
40414243 44454647 48494a4b 4c4d4e4f 50515253 54555657 58595a5b 5c5d5e5f
60616263 64656667 68696a6b 6c6d6e6f 70717273 74757677 78797a7b 7c7d7e7f
80818283 84858687 88898a8b 8c8d8e8f 90919293 94959697 98999a9b 9c9d9e9f
a0a1a2a3 a4a5a6a7 a8a9aaab acadaeaf b0b1b2b3 b4b5b6b7 b8b9babb bcbdbebf
c0c1c2c3 c4c5c6c7 c8c9cacb cccdcecf d0d1

Z=
00000101 02020303 04040505 06060707 08080909 0a0a0b0b 0c0c0d0d 0e0e0f0f
10101111 12121313 14141515 16161717 18181919 1a1a1b1b 1c1c1d1d 1e1e1f1f
20202121 22222323 24242525 26262727 28282929 2a2a2b2b 2c2c2d2d 2e2e2f2f
30303131 32323333 34343535 36363737 38383939 3a3a3b3b 3c3c3d3d 3e3e3f3f
40404141 42424343 44444545 46464747 48484949 4a4a4b4b 4c4c4d4d 4e4e4f4f
50505151 52525353 54545555 56565757 58585959 5a5a5b5b 5c5c5d5d 5e5e5f5f
60606161 62626363 64646565 66666767 6868

(X != Y) is 0
(X != Z) is 1
```

As we can see, the function works well.

Let's check if there are cases where the function does not work properly. For this, we will set y[N - 1] = 0x10; (The program has two parts: the first is processing 4-byte commands, and in the end, 1-byte ones).

```
40414243 44454647 48494a4b 4c4d4e4f 50515253 54555657 58595a5b 5c5d5e5f
60616263 64656667 68696a6b 6c6d6e6f 70717273 74757677 78797a7b 7c7d7e7f
80818283 84858687 88898a8b 8c8d8e8f 90919293 94959697 98999a9b 9c9d9e9f
a0a1a2a3 a4a5a6a7 a8a9aaab acadaeaf b0b1b2b3 b4b5b6b7 b8b9babb bcbdbebf
c0c1c2c3 c4c5c6c7 c8c9cacb cccdcecf d0d1

Y=
00010203 04050607 08090a0b 0c0d0e0f 10111213 14151617 18191a1b 1c1d1e1f
20212223 24252627 28292a2b 2c2d2e2f 30313233 34353637 38393a3b 3c3d3e3f
40414243 44454647 48494a4b 4c4d4e4f 50515253 54555657 58595a5b 5c5d5e5f
60616263 64656667 68696a6b 6c6d6e6f 70717273 74757677 78797a7b 7c7d7e7f
80818283 84858687 88898a8b 8c8d8e8f 90919293 94959697 98999a9b 9c9d9e9f
a0a1a2a3 a4a5a6a7 a8a9aaab acadaeaf b0b1b2b3 b4b5b6b7 b8b9babb bcbdbebf
c0c1c2c3 c4c5c6c7 c8c9cacb cccdcecf d010

Z=
00000101 02020303 04040505 06060707 08080909 0a0a0b0b 0c0c0d0d 0e0e0f0f
10101111 12121313 14141515 16161717 18181919 1a1a1b1b 1c1c1d1d 1e1e1f1f
20202121 22222323 24242525 26262727 28282929 2a2a2b2b 2c2c2d2d 2e2e2f2f
30303131 32323333 34343535 36363737 38383939 3a3a3b3b 3c3c3d3d 3e3e3f3f
40404141 42424343 44444545 46464747 48484949 4a4a4b4b 4c4c4d4d 4e4e4f4f
50505151 52525353 54545555 56565757 58585959 5a5a5b5b 5c5c5d5d 5e5e5f5f
60606161 62626363 64646565 66666767 6868

(X != Y) is 1
(X != Z) is 1
```

If y[N - 3] = 0x10;

```
40414243 44454647 48494a4b 4c4d4e4f 50515253 54555657 58595a5b 5c5d5e5f
60616263 64656667 68696a6b 6c6d6e6f 70717273 74757677 78797a7b 7c7d7e7f
80818283 84858687 88898a8b 8c8d8e8f 90919293 94959697 98999a9b 9c9d9e9f
a0a1a2a3 a4a5a6a7 a8a9aaab acadaeaf b0b1b2b3 b4b5b6b7 b8b9babb bcbdbebf
c0c1c2c3 c4c5c6c7 c8c9cacb cccdcecf d0d1

Y=
00010203 04050607 08090a0b 0c0d0e0f 10111213 14151617 18191a1b 1c1d1e1f
20212223 24252627 28292a2b 2c2d2e2f 30313233 34353637 38393a3b 3c3d3e3f
40414243 44454647 48494a4b 4c4d4e4f 50515253 54555657 58595a5b 5c5d5e5f
60616263 64656667 68696a6b 6c6d6e6f 70717273 74757677 78797a7b 7c7d7e7f
80818283 84858687 88898a8b 8c8d8e8f 90919293 94959697 98999a9b 9c9d9e9f
a0a1a2a3 a4a5a6a7 a8a9aaab acadaeaf b0b1b2b3 b4b5b6b7 b8b9babb bcbdbebf
c0c1c2c3 c4c5c6c7 c8c9cacb cccdce10 d0d1

Z=
00000101 02020303 04040505 06060707 08080909 0a0a0b0b 0c0c0d0d 0e0e0f0f
10101111 12121313 14141515 16161717 18181919 1a1a1b1b 1c1c1d1d 1e1e1f1f
20202121 22222323 24242525 26262727 28282929 2a2a2b2b 2c2c2d2d 2e2e2f2f
30303131 32323333 34343535 36363737 38383939 3a3a3b3b 3c3c3d3d 3e3e3f3f
40404141 42424343 44444545 46464747 48484949 4a4a4b4b 4c4c4d4d 4e4e4f4f
50505151 52525353 54545555 56565757 58585959 5a5a5b5b 5c5c5d5d 5e5e5f5f
60606161 62626363 64646565 66666767 6868

(X != Y) is 1
(X != Z) is 1
```

Everything works as it should.