HW2- P3

Yanhan Lin

Q1:

This program is to calculate the value of 3 * 5 – 3/2. Result is 14.

C code is shown below

```
#include<stdio.h>
#include<stdlib.h>

int main(void) {
    int x = 3, y = 5, z = 0;
    z = x*y - x/2;
    printf("%d", z);
    return 0;
}
```

The output of Q1 has been attached below

```
F:\EECS 444\hw2q1.exe
14
--------------------------------
Process exited after 4.195 seconds with return value 0
请按任意键继续. . .
```

To confirm the result, assembly code is compared with original one.

IDA analysis for Q1

IDA View-A

```
; Attributes: bp-based frame fuzzy-sp

; int __cdecl main(int argc, const char **argv, const char **envp)
public _main
_main proc near

argc= dword ptr  8
argv= dword ptr  0Ch
envp= dword ptr  10h

push    ebp
mov     ebp, esp
and     esp, 0FFFFFFF0h
sub     esp, 20h
call    ___main
mov     dword ptr [esp+1Ch], 3
mov     dword ptr [esp+18h], 5
mov     dword ptr [esp+14h], 0
mov     eax, [esp+1Ch]
imul    eax, [esp+18h]
mov     edx, eax
mov     eax, [esp+1Ch]
mov     ecx, eax
shr     ecx, 1Fh
add     eax, ecx
sar     eax, 1
sub     edx, eax
mov     eax, edx
mov     [esp+14h], eax
mov     eax, [esp+14h]
mov     [esp+4], eax
mov     dword ptr [esp], offset aD ; "%d"
call    _printf
mov     eax, 0
leave
retn
_main endp
```

100.00% (-64,-8) (582,149) 00000900 0000000000401500: _ma (Synchronized with He:

Q2:

This program is to find the greatest one in an integer array with length of 8, and then print the greatest one. Therefore, the greatest one is 432 in this case.

C code

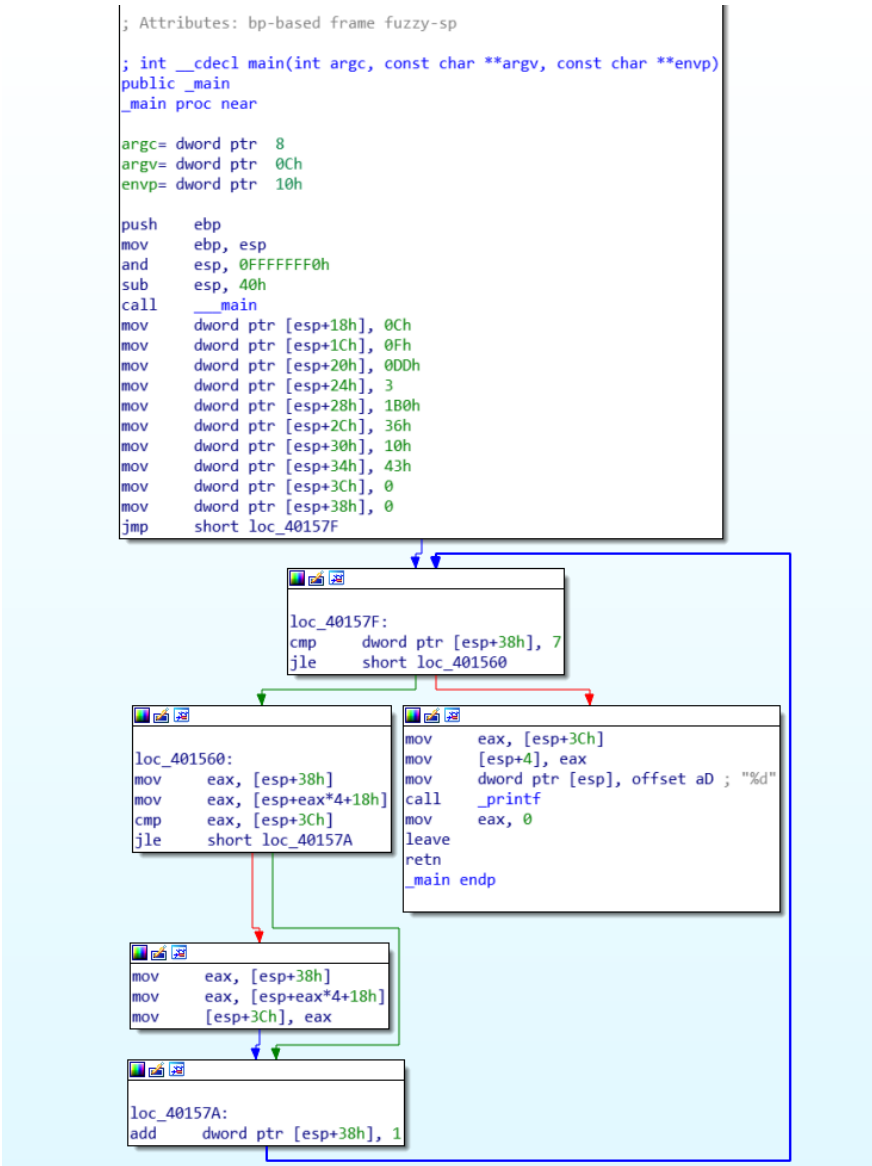
```
#include<stdio.h>
#include<stdlib.h>
```

```
int main(void) {
    int arr[8] = {12, 15, 221, 3, 432, 54, 16, 67};
    int max = 0, i = 0;

    while (i <= 7) {
        if (arr[i] > max) {
            max = arr[i];
        }
        i++;
    }
    printf("%d", max);

    return 0;
}
```

Output is shown below,

```
F:\EECS 444\hw2q2.exe
432
--------------------------------
Process exited after 0.01729 seconds with return value 0
请按任意键继续. . .
```

IDA analysis for Q2



Q3

The functionality of Q3 is to find the value x from 100 to 999 which satisfies sum of cube of three variables involving this value, which is x equals a^3 + b ^3 + c^3, and then print results, which are 153, 370, 371, and 407.

Detailed code in c is shown below

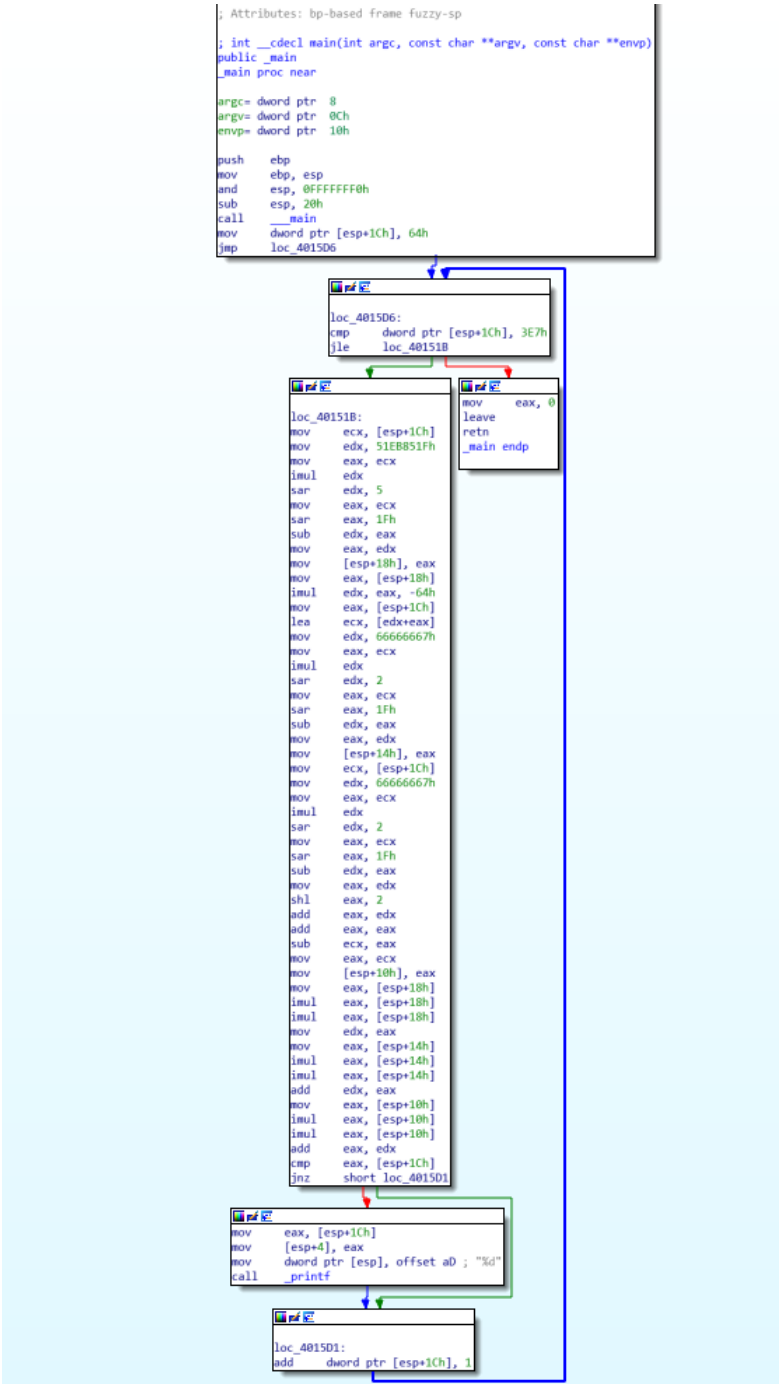
```
#include<stdio.h>
#include<stdlib.h>
```

```
int main(void) {
    int x = 100;
    int a, b, c;
    while (x <= 999) {
        a = x/100;
        b = (-100*a + x)/10;
        c = x - x/10 * 10;
        if (x == a*a*a + b*b*b + c*c*c) {
            printf("%d", x);
        }
        x++;
    }

    return 0;
}
```

Output of Q3

```
F:\EECS 444\hw2q3.exe
153370371407
--------------------------------
Process exited after 0.01797 seconds with return value 0
请按任意键继续. . .
```

IDA analysis for Q3



Q4

Q4 is tricky. The functionality of Q4 is to find the last non-zero number in an array from 1 to 100. Take out this value when it counts 7 and reset it to 0. In detail, this array surround to a circle, take out each non-zero value when it counts 7 in loop 100.

In this case, the last non-zero value to take is 50.

C code is shown below

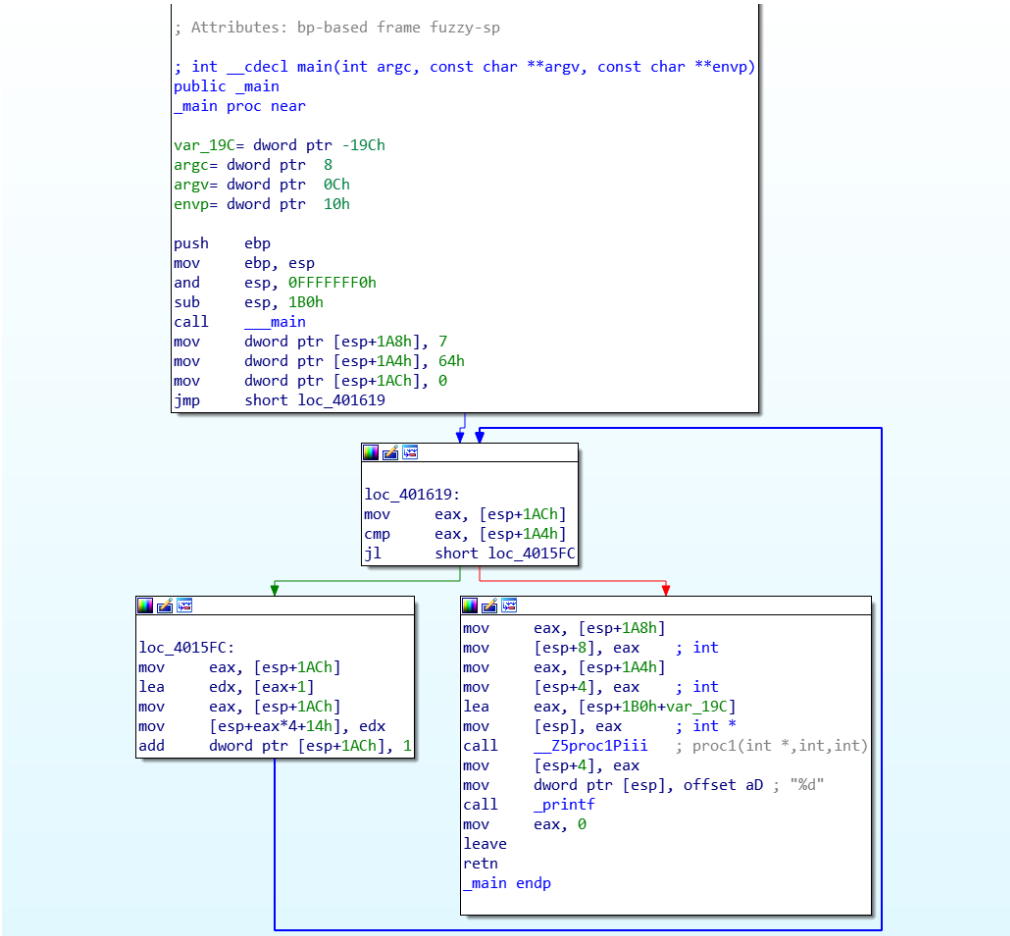
```
#include<stdio.h>

int _Z5proc1Piii(int* arr, int length, int num) {
    // length = 100, num = 7
    int iters, count, index, lastTaken;
    index = 0;
    lastTaken = 0;
    iters = 0;
    while (iters < length) {
        // find the num-th 'non-zero' index
        count = 1;
        // find (num-1)-th 'non-zero' index, and set index to the next
        while (count < num) {
            // find the first index 'index' where arr[index] != 0
            while (arr[index] == 0) {
                index = (index + 1) % length;
            }

            count++;
            // index is the next of the first 'non-zero' index
            index = (index + 1) % length;
        }
        // find the first 'non-zero' index after (num-1)-th 'non-zero' index
        while (arr[index] == 0) {
            index = (index + 1) % length;
        }
```

```
        lastTaken = arr[index];
        arr[index] = 0;
        iters++;
    }

    return lastTaken;
}

int main(void) {
    int arr[100];
    int num = 7, length = 100, i = 0;
    // arr: 1 ~ 100
    while (i < length) {
        arr[i] = i + 1;
        i++;
    }

    printf("%d", _Z5proc1Piii(arr, length, num));
    return 0;
}
```

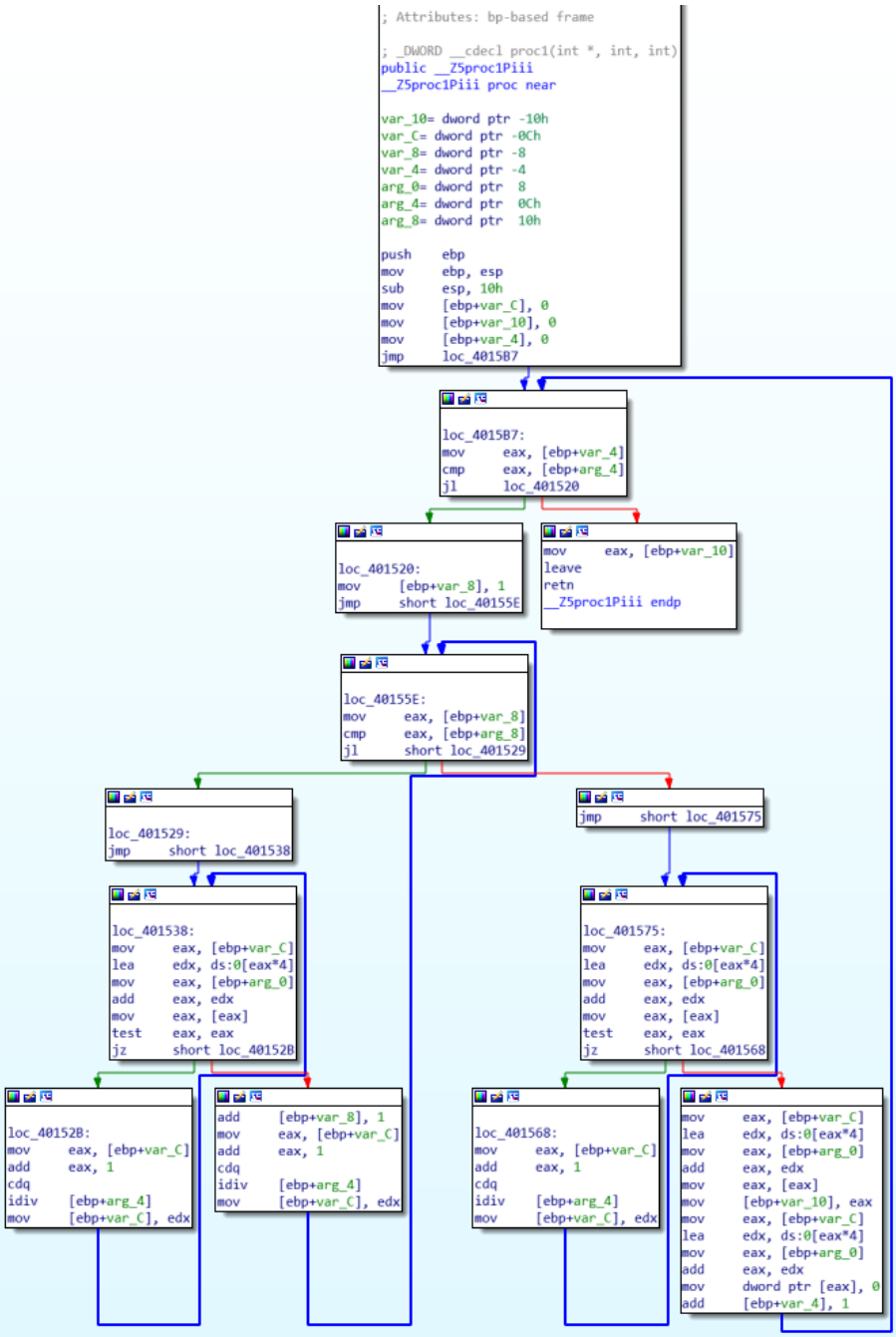
output of Q4 is

```
F:\EECS 444\hw2q4.exe
50
--------------------------------
Process exited after 0.01377 seconds with return value 0
请按任意键继续. . .
```

IDA analysis of Q4



IDA analysis for sub-function


; Attributes: bp-based frame
; _DWORD __cdecl proc1(int *, int, int)
public __Z5proc1Piii
__Z5proc1Piii proc near
var_10= dword ptr -10h
var_C= dword ptr -0Ch
var_8= dword ptr -8
var_4= dword ptr -4
arg_0= dword ptr 8
arg_4= dword ptr 0Ch
arg_8= dword ptr 10h
push ebp
mov ebp, esp
sub esp, 10h
mov [ebp+var_C], 0
mov [ebp+var_10], 0
mov [ebp+var_4], 0
jmp loc_4015B7
loc_4015B7:
mov eax, [ebp+var_4]
cmp eax, [ebp+arg_4]
jl loc_401520
loc_401520:
mov [ebp+var_8], 1
jmp short loc_40155E
mov eax, [ebp+var_10]
leave
retn
__Z5proc1Piii endp
loc_40155E:
mov eax, [ebp+var_8]
cmp eax, [ebp+arg_8]
jl short loc_401529
loc_401529:
jmp short loc_401538
jmp short loc_401575
loc_401538:
mov eax, [ebp+var_C]
lea edx, ds:0[eax*4]
mov eax, [ebp+arg_0]
add eax, edx
mov eax, [eax]
test eax, eax
jz short loc_40152B
loc_401575:
mov eax, [ebp+var_C]
lea edx, ds:0[eax*4]
mov eax, [ebp+arg_0]
add eax, edx
mov eax, [eax]
test eax, eax
jz short loc_401568
loc_40152B:
mov eax, [ebp+var_C]
add eax, 1
cdq
idiv [ebp+arg_4]
mov [ebp+var_C], edx
add [ebp+var_8], 1
mov eax, [ebp+var_C]
add eax, 1
cdq
idiv [ebp+arg_4]
mov [ebp+var_C], edx
loc_401568:
mov eax, [ebp+var_C]
add eax, 1
cdq
idiv [ebp+arg_4]
mov [ebp+var_C], edx
mov eax, [ebp+var_C]
lea edx, ds:0[eax*4]
mov eax, [ebp+arg_0]
add eax, edx
mov eax, [eax]
mov [ebp+var_10], eax
mov eax, [ebp+var_C]
lea edx, ds:0[eax*4]
mov eax, [ebp+arg_0]
add eax, edx
mov dword ptr [eax], 0
add [ebp+var_4], 1