#include<stdio.h>

int main(){

00A01840 push ebp

00A01841 mov ebp,esp

00A01843 sub esp,0D8h

00A01849 push ebx

00A0184A push esi

00A0184B push edi

00A0184C lea edi,[ebp-18h]

00A0184F mov ecx,6

00A01854 mov eax,0CCCCCCCCh

00A01859 rep stos dword ptr es:[edi]

00A0185B mov ecx,offset \_65B1FB4C\_hw1@c (0A0C003h)

00A01860 call @\_\_CheckForDebuggerJustMyCode@4 (0A01316h)

int sum = 0;

00A01865 mov dword ptr [sum],0

#将0移入变量sum的寄存器中，dword ptr指双字地址

for (int i = 0; i < 10; i++) {

00A0186C mov dword ptr [ebp-14h],0

#将0移入地址为ebp-14h的寄存器中，此处存储的为变量i的值

00A01873 jmp \_\_$EncStackInitStart+32h (0A0187Eh)

#跳转至下文(1)处，对i的值进行判断（此条语句仅在进入for循环时执行一次）

00A01875 mov eax,dword ptr [ebp-14h]

(3)#将变量i的值移至%eax中

00A01878 add eax,1

#对%eax中的值进行加1操作

00A0187B mov dword ptr [ebp-14h],eax

#将%eax中更新过的值移回变量i的存储地址中

00A0187E cmp dword ptr [ebp-14h],0Ah

(1)#比较i的值，若>=10则执行下一条指令

00A01882 jge \_\_$EncStackInitStart+67h (0A018B3h)

#跳转至下文(4)处结束for循环，进行输出语句

if (i <= 5) {

00A01884 cmp dword ptr [ebp-14h],5

#比较i的值，若>5则执行下一条指令

00A01888 jg \_\_$EncStackInitStart+51h (0A0189Dh)

#跳转至下文(2)处，进入else分支

sum = sum + 2;

00A0188A mov eax,dword ptr [sum]

#将变量sum的值移至%eax中

00A0188D add eax,2

#对%eax中的值进行加2操作

00A01890 mov dword ptr [sum],eax

#将%eax中更新过的值移回变量sum的存储地址中

sum = sum \* 2;

00A01893 mov eax,dword ptr [sum]

#将变量sum的值移至%eax中

00A01896 shl eax,1

#对%eax中的值进行乘2操作（左移一位

00A01898 mov dword ptr [sum],eax

#将%eax中更新过的值移回变量sum的存储地址中

}

00A0189B jmp \_\_$EncStackInitStart+65h (0A018B1h)

#i<=5的分支执行完毕，跳转至（2）处

else {

sum = sum - 2;

00A0189D mov eax,dword ptr [sum]

#将变量sum的值移至%eax中

00A018A0 sub eax,2

#对%eax中的值进行减2操作

00A018A3 mov dword ptr [sum],eax

#将%eax中更新过的值移回变量sum的存储地址中

sum = sum / 2;

00A018A6 mov eax,dword ptr [sum]

#将变量sum的值移至%eax中

00A018A9 cdq

#该指令先把edx的每一位置成eax的最高位,（若eax>=0x80000000, edx=0xFFFFFFFF；若eax<0x80000000，edx=0x00000000。） 再把edx扩展为eax的高位，即变为64位

00A018AA sub eax,edx

#寄存器%eax 和%edx的值相减

00A018AC sar eax,1

#对%eax中的值进行除2操作（右移一位

00A018AE mov dword ptr [sum],eax

#将%eax中更新过的值移回变量sum的存储地址中

}

}

00A018B1 jmp \_\_$EncStackInitStart+29h (0A01875h)

(2)#if语句执行完成,跳至(3)处进行i++操作

printf("%d", sum);

00A218B3 mov eax,dword ptr [sum]

(4) #将变量sum的值移至%eax中，下面进行打印输出

00A218B6 push eax

00A218B7 push offset string "%d" (0A27B30h)

00A218BC call \_printf (0A210CDh)

00A218C1 add esp,8

}

00A218C4 xor eax,eax

00A218C6 pop edi

00A218C7 pop esi

00A218C8 pop ebx

00A218C9 add esp,0D8h

00A218CF cmp ebp,esp

00A218D1 call \_\_RTC\_CheckEsp (0A2123Fh)

00A218D6 mov esp,ebp

00A218D8 pop ebp

00A218D9 ret