

```
--- E:\Desktop\大三上课程文件\汇编语言\Homework01\hw2_C语言代码\hw2_C语言代码\hw2_C语言代码.c -----
#include <stdio.h>
```

```
int main() {
00007FF7BC611880 push    rbp      ; 保存基址指针
00007FF7BC611882 push    rdi      ; 保存rdi寄存器
00007FF7BC611883 sub     rsp,148h   ; 在栈上分配148h字节空间给局部变量
00007FF7BC61188A lea     rbp,[rsp+20h] ; 设置新的基址指针
00007FF7BC61188F lea     rcx,[DE5EAC98_hw2_C语言代码@c (07FF7BC621008h)] ; 加载调试信息地址
00007FF7BC611896 call   CheckForDebuggerJustMyCode (07FF7BC61136Bh) ; 调用调试检查函数
```

```
char ch = 'a';
```

```
00007FF7BC61189B mov     byte ptr [ch],61h   ; ch = 'a' (ASCII码61h)
```

```
int i, j;
```

```
// 外层循环开始 (i从0到1)
for (i = 0; i < 2; i++) {
```

```
00007FF7BC61189F mov     dword ptr [i],0    ; i = 0
00007FF7BC6118A6 jmp     main+30h (07FF7BC6118B0h) ; 跳转到循环条件检查
```

```
; 循环体结束后的跳转点
```

```
00007FF7BC6118A8 mov     eax,dword ptr [i]    ; 将i加载到eax
00007FF7BC6118AB inc     eax      ; i++
00007FF7BC6118AD mov     dword ptr [i],eax   ; 将更新后的i存回内存
```

```
; 外层循环条件检查
```

```
00007FF7BC6118B0 cmp     dword ptr [i],2    ; 比较i与2
00007FF7BC6118B4 jge     main+78h (07FF7BC6118F8h) ; 如果i >= 2, 跳出循环到程序结束
```

```
// 内层循环开始 (j从0到12)
for (j = 0; j < 13; j++) {
```

```
00007FF7BC6118B6 mov     dword ptr [j],0    ; j = 0
00007FF7BC6118BD jmp     main+47h (07FF7BC6118C7h) ; 跳转到内层循环条件检查
```

```
; 内层循环体结束后的跳转点
```

```
00007FF7BC6118BF mov     eax,dword ptr [j]    ; 将j加载到eax  
00007FF7BC6118C2 inc     eax          ; j++  
00007FF7BC6118C4 mov     dword ptr [j],eax   ; 将更新后的j存回内存
```

; 内层循环条件检查

```
00007FF7BC6118C7 cmp     dword ptr [j],0Dh    ; 比较j与13(0Dh)  
00007FF7BC6118CB jge     main+6Ah (07FF7BC6118EAh) ; 如果j >= 13, 跳出内层循环到换行处
```

```
// 循环体: 打印字符和空格  
printf("%c ", ch);
```

```
00007FF7BC6118CD movsx   eax,byte ptr [ch]    ; 将ch符号扩展到eax (准备参数)  
00007FF7BC6118D1 mov     edx, eax        ; 将字符值移动到edx (可能是第二个参数)  
00007FF7BC6118D3 lea     rcx,[string "%c " (07FF7BC619C24h)] ; 加载格式字符串地址到rcx (第一个参数)  
00007FF7BC6118DA call    printf (07FF7BC611195h) ; 调用printf函数
```

ch++; // 移动到下一个字符

```
00007FF7BC6118DF movzx   eax,byte ptr [ch]    ; 将ch零扩展到eax  
00007FF7BC6118E3 inc     al          ; al寄存器加1 (ch++)  
00007FF7BC6118E5 mov     byte ptr [ch],al   ; 将更新后的字符存回ch
```

} // 内层循环结束

```
00007FF7BC6118E8 jmp     main+3Fh (07FF7BC6118BFh) ; 跳回内层循环继续执行
```

```
printf("\n"); // 打印换行
```

```
00007FF7BC6118EA lea     rcx,[string "\n" (07FF7BC619C28h)] ; 加载换行符字符串地址到rcx  
00007FF7BC6118F1 call    printf (07FF7BC611195h) ; 调用printf函数打印换行
```

} // 外层循环结束

```
00007FF7BC6118F6 jmp     main+28h (07FF7BC6118A8h) ; 跳回外层循环继续执行
```

```
return 0;
```

```
00007FF7BC6118F8 xor    eax,eax      ; 将eax清零（返回值为0）
} // 函数结束
00007FF7BC6118FA lea    rsp,[rbp+128h] ; 恢复栈指针
00007FF7BC611901 pop   rdi          ; 恢复rdi寄存器
00007FF7BC611902 pop   rbp          ; 恢复基址指针
00007FF7BC611903 ret
```