

```

int main() {
00007FF6642A1880 push    rbp      ; 保存基址指针
00007FF6642A1882 push    rdi      ; 保存目的索引寄存器
00007FF6642A1883 sub     rsp,108h  ; 在栈上分配局部变量空间(264字节)
00007FF6642A188A lea     rbp,[rsp+20h] ; 设置新的基址指针
00007FF6642A188F lea     rcx,[944E5FE1_hw4_01_C@c (07FF6642B1008h)]
00007FF6642A1896 call    CheckForDebuggerJustMyCode (07FF6642A1370h) ; 调试检查
    printf("The 9mu19 table:\n");
00007FF6642A189B lea     rcx,[string "The 9mu19 table:\n" (07FF6642A9C40h)]
00007FF6642A18A2 call    printf (07FF6642A119Ah) ; 调用printf输出标题
    for (int i = 9; i >= 1; i--) {
00007FF6642A18A7 mov     dword ptr [rbp+4],9 ; 初始化循环变量i=9 (存储在[rbp+4])
00007FF6642A18AE jmp     main+38h (07FF6642A18B8h) ; 跳转到循环条件检查
00007FF6642A18B0 mov     eax,dword ptr [rbp+4] ; 循环体开始: 加载i到eax
00007FF6642A18B3 dec     eax      ; i-- (递减)
00007FF6642A18B5 mov     dword ptr [rbp+4],eax ; 将递减后的值存回i
00007FF6642A18B8 cmp     dword ptr [rbp+4],1 ; 比较i和1
00007FF6642A18BC jl      main+48h (07FF6642A18C8h) ; 如果i<1, 跳出循环
        print_row(i);
00007FF6642A18BE mov     ecx,dword ptr [rbp+4] ; 将i作为参数传入ecx寄存器
00007FF6642A18C1 call    print_row (07FF6642A10DCh) ; 过程调用: 调用print_row函数
    }
00007FF6642A18C6 jmp     main+30h (07FF6642A18B0h) ; 跳回循环开始, 继续下一次迭代
    return 0;
00007FF6642A18C8 xor     eax,eax ; 返回值设为0 (eax=0)
}
00007FF6642A18CA lea     rsp,[rbp+0E8h] ; 恢复栈指针
00007FF6642A18D1 pop     rdi      ; 恢复rdi寄存器
00007FF6642A18D2 pop     rbp      ; 恢复基址指针
00007FF6642A18D3 ret
; 函数返回

```

```

int main() {
00007FF7B7C91980 push    rbp      ; 保存基址指针
00007FF7B7C91982 push    rdi      ; 保存目的索引寄存器
00007FF7B7C91983 sub     rsp,0E8h  ; 在栈上分配局部变量空间(232字节)
00007FF7B7C9198A lea     rbp,[rsp+20h] ; 设置新的基址指针
00007FF7B7C9198F lea     rcx,[E580E9C9_hw4_02_C@c (07FF7B7CA2008h)]
00007FF7B7C91996 call    CheckForDebuggerJustMyCode (07FF7B7C91370h) ; 调试检查

```

```
    check_table();
00007FF7B7C9199B call    check_table (07FF7B7C91127h) ; 过程调用: 调用检查函数
    return 0;
00007FF7B7C919A0 xor     eax,eax    ; 返回值设为0 (eax=0)
}
00007FF7B7C919A2 lea     rsp,[rbp+0C8h] ; 恢复栈指针
00007FF7B7C919A9 pop     rdi      ; 恢复rdi寄存器
00007FF7B7C919AA pop     rbp      ; 恢复基址指针
00007FF7B7C919AB ret     ; 函数返回
```