作业二：打印ASCII表

1）

**思路：**

loop1负责打印一行字母

loop2主要负责打印完一行后输出回车

考虑每行输出13个字母，所以cx寄存器设置13，为内圈循环次数。把外圈循环次数2放入bx低八位暂存

**代码：（文档中的ASCII\_1.asm）**

code segment

assume cs:code

start:

mov bl,2 ;外圈循环次数

mov cx,13 ;内圈循环次数

mov ah,02h

mov dl,'a' ;字符a

loop2: ;外圈循环开始

mov cx,13 ;第二次要重新初始化次数

loop1: ;内圈循环开始

int 21h ;打印字符

add dl,1 ;ASCII+1

loop loop1 ;内圈循环结束

mov bh,dl ;暂存当前字符

mov dl,0ah ;存入回车

int 21h ;打印回车

mov dl,bh ;恢复当前字符

mov cl,bl ;计数器更新外圈循环次数

sub bl,1 ;外圈循环次数-1

loop loop2 ;外圈循环结束

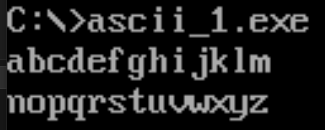
mov ah,4ch

int 21h

code ends

end start

**运行结果：**

****

2）

**思路：**

条件跳转当判断当前输出数量是13个字母时，打印回车，否则跳转继续输出字母。所以开始时，打印数量直接设置为26

**代码：（文档中的ASCII\_2.asm）:**

code segment

assume cs:code

start:

mov cx,26

mov ah,02h

mov dl,'a' ;字符a

row1: ;打印第一行

sub cx,1 ;次数-1

int 21h ;打印字符

add dl,1 ;ASCII+1

cmp cx,13 ;比较

jne row1 ;不等跳转

mov bl,dl ;暂存当前字符

mov dl,0ah ;存入回车

int 21h ;打印回车

mov dl,bl ;恢复当前字符

row2: ;打印第二行

sub cx,1 ;次数-1

int 21h ;打印字符

add dl,1 ;ASCII+1

cmp cx,0 ;比较

jne row2 ;不等跳转

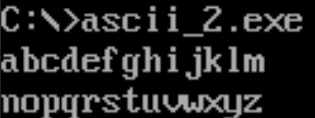
mov ah,4ch

int 21h

code ends

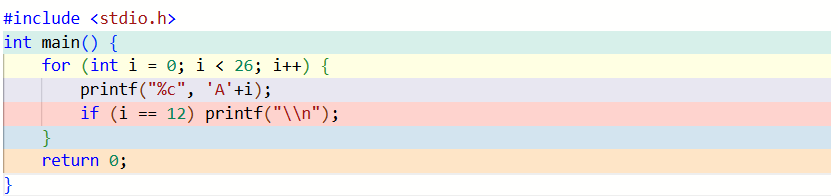
end start

**运行结果：**

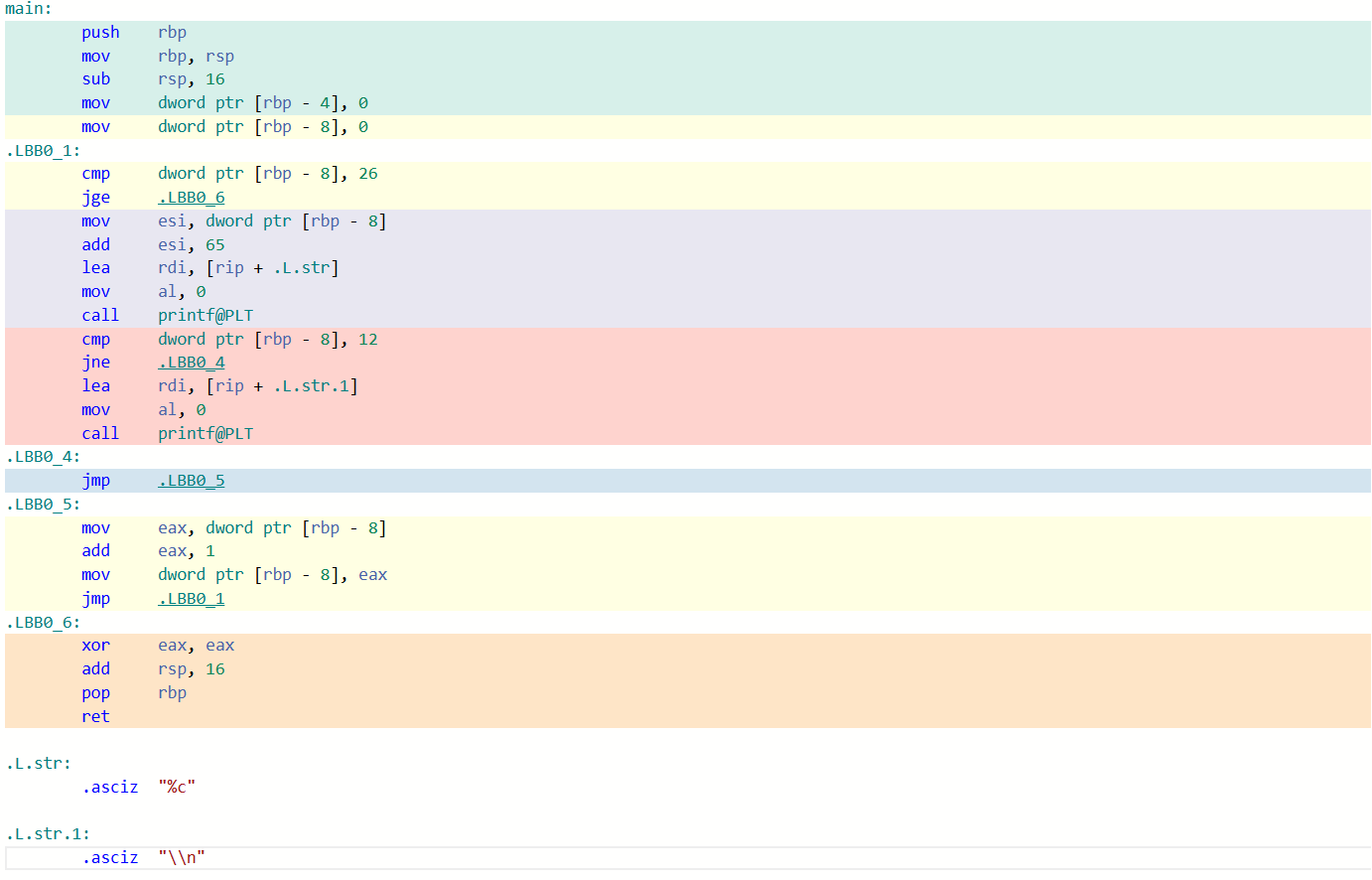
****

3)

**C语言实现：**

****

**反汇编代码：**

****

**给代码加注释：**

main:

push rbp ; 保存旧的基指针

mov rbp, rsp ; 设置新的基指针

sub rsp, 16 ; 为局部变量分配空间

mov dword ptr [rbp - 4], 0 ; 初始化变量，可能用于计数或其他目的

mov dword ptr [rbp - 8], 0 ; 初始化循环计数器为0

.LBB0\_1: ; 循环标签

cmp dword ptr [rbp - 8], 26 ; 比较计数器和26

jge .LBB0\_6 ; 如果计数器大于等于26，跳转到结束标签

mov esi, dword ptr [rbp - 8] ; 将计数器的值移动到esi

add esi, 65 ; 将计数器的值转换为ASCII码（'A'的ASCII码是65）

lea rdi, [rip + .L.str] ; 加载格式化字符串的地址到rdi

mov al, 0 ; 清空al寄存器

call printf@PLT ; 调用printf函数打印字符

cmp dword ptr [rbp - 8], 12 ; 比较计数器和12

jne .LBB0\_4 ; 如果计数器不等于12，跳过接下来的换行打印

lea rdi, [rip + .L.str.1] ; 加载换行字符串的地址到rdi

mov al, 0 ; 清空al寄存器

call printf@PLT ; 调用printf函数打印换行符

.LBB0\_4: ; 跳转标签

; (这里没有操作，直接跳转到.LBB0\_5)

.LBB0\_5: ; 循环递增标签

mov eax, dword ptr [rbp - 8] ; 将计数器的值移动到eax

add eax, 1 ; 计数器加1

mov dword ptr [rbp - 8], eax ; 更新计数器的值

jmp .LBB0\_1 ; 无条件跳转回循环开始

.LBB0\_6: ; 结束标签

xor eax, eax ; 清空eax寄存器

add rsp, 16 ; 释放之前分配的空间

pop rbp ; 恢复旧的基指针

ret ; 返回到调用者

.L.str: ; 字符串标签

.asciz "%c" ; 格式化字符串，用于打印单个字符

.L.str.1: ; 字符串标签

.asciz "\\n" ; 换行字符串