

作业题目 2：按要求打印 ASCII 表

基本要求：输出 ASCII 表中的小写字母部分，要求每行 13 个字符。

1. 用 loop 指令实现

```
C:\>ascii1
a b c d e f g h i j k l m
n o p q r s t u v w x y z
```

2. 用条件跳转指令实现

```
C:\>ascii2
a b c d e f g h i j k l m
n o p q r s t u v w x y z
```

3. C 语言实现后查看反汇编代码并加注释

反汇编代码加注释后如下：

0000000140001380 <main>:

```
140001380: 55                push    %rbp
140001381: 48 89 e5          mov     %rsp,%rbp
140001384: 48 83 ec 30       sub     $0x30,%rsp
140001388: e8 ef 00 00 00    call   14000147c <__main>
```

; 初始化局部变量

```
14000138d: c6 45 ff 61      movb    $0x61,-0x1(%rbp) ; char c = 'a' (0x61)
140001391: c7 45 f8 00 00 00 movl    $0x0,-0x8(%rbp) ; int i = 0
```

; 跳转到循环条件检查

```
140001398: eb 5f            jmp     1400013f9 <main+0x79>
```

; 循环体开始

```
14000139a: 0f be 45 ff      movsbl -0x1(%rbp),%eax ; 将字符 c 符号扩展为 32
```

位

```
14000139e: 89 c1            mov     %eax,%ecx ; 参数准备
1400013a0: e8 ab 0d 00 00    call   140002150 <putchar> ; putchar(c)
```

; 字符递增

```
1400013a5: 0f b6 45 ff      movzbl -0x1(%rbp),%eax ; 读取字符 c
1400013a9: 83 c0 01          add     $0x1,%eax ; c++
1400013ac: 88 45 ff          mov     %al,-0x1(%rbp) ; 存回变量
```

; 计数器递增

```
1400013af: 83 45 f8 01      addl    $0x1,-0x8(%rbp) ; i++
```

; 复杂的模运算 (i % 5 == 0 ?)

; 编译器优化：用乘法代替除法

```
1400013b3: 8b 4d f8          mov     -0x8(%rbp),%ecx ; 加载 i 到 ecx
1400013b6: 48 63 c1          movslq %ecx,%rax ; 符号扩展 i 到 64 位
1400013b9: 48 69 c0 4f ec c4 4e imul    $0x4ec4ec4f,%rax,%rax ; 乘以魔数
1400013c0: 48 c1 e8 20       shr     $0x20,%rax ; 右移 32 位
```

1400013c4:	89 c2	mov	%eax,%edx	
1400013c6:	c1 fa 02	sar	\$0x2,%edx	; 算术右移 2 位
1400013c9:	89 c8	mov	%ecx,%eax	
1400013cb:	c1 f8 1f	sar	\$0x1f,%eax	; 获取符号位
1400013ce:	29 c2	sub	%eax,%edx	
1400013d0:	89 d0	mov	%edx,%eax	
1400013d2:	01 c0	add	%eax,%eax	; eax * 2
1400013d4:	01 d0	add	%edx,%eax	; eax * 3
1400013d6:	c1 e0 02	shl	\$0x2,%eax	; eax * 4
1400013d9:	01 d0	add	%edx,%eax	; eax * 5
1400013db:	29 c1	sub	%eax,%ecx	; i - (i/5)*5 = i % 5
1400013dd:	89 ca	mov	%ecx,%edx	
; 条件判断: if (i % 5 == 0)				
1400013df:	85 d2	test	%edx,%edx	; 测试余数是否为 0
1400013e1:	75 0c	jne	1400013ef <main+0x6f>	; 不为 0 则跳转
; 如果 i % 5 == 0, 输出换行符				
1400013e3:	b9 0a 00 00 00	mov	\$0xa,%ecx	; '\n' = 0x0a
1400013e8:	e8 63 0d 00 00	call	140002150 <putchar>	; putchar('\n')
1400013ed:	eb 0a	jmp	1400013f9 <main+0x79>	; 跳转到循环条件
; 否则输出空格				
1400013ef:	b9 20 00 00 00	mov	\$0x20,%ecx	; ' ' = 0x20
1400013f4:	e8 57 0d 00 00	call	140002150 <putchar>	; putchar(' ')
; 循环条件检查: while (c <= 'z')				
1400013f9:	80 7d ff 7b	cmpb	\$0x7b,-0x1(%rbp)	; 比较 c 与 0x7b
('z'+1)				
1400013fd:	75 9b	jne	14000139a <main+0x1a>	; 如果 c != 'z'+1, 继续循环
; 函数返回				
1400013ff:	b8 00 00 00 00	mov	\$0x0,%eax	; return 0
140001404:	48 83 c4 30	add	\$0x30,%rsp	; 恢复栈指针
140001408:	5d	pop	%rbp	; 恢复基址指针
140001409:	c3	ret		; 返回