有如下的 C 语言代码，以及编译生成的对应汇编代码，其中注释掉 if (likely (a == 2))这行生成汇编代码段-1，注释掉 if (unlikely (a == 2)) 这行生成汇编代码段-2。

问题：请简要分析编译指示（directives）
“#define likely(x)   __builtin_expect(!!(x), 1)
#define unlikely(x)   __builtin_expect(!!(x), 0)”
的作用——为何生成的指令序列的顺序不同，与处理器流水线的运行过程与优化有何关系？

```
#include<stdlib.h>
#define likely(x)    __builtin_expect(!!(x), 1)
#define unlikely(x)  __builtin_expect(!!(x), 0)
int main(char *argv[], int argc)
{
   int a,b;
   /* Get the value from somewhere GCC can't optimize */
   a = atoi (argv[1]);
   b = a*a;
   if (unlikely (a == 2))
   // if (likely (a == 2))
   {
      a++;  b++;
   }
   else
   {
      a--;  b--;
   }
   return a+b;
}
```

代码段-1

```
main:
        subq     $8, %rsp
        movq      8(%rdi), %rdi
        xorl     %esi, %esi
        movl      $10, %edx
        call     strtol                # atoi 调用，返回值在 eax 中
        movl      %eax, %esi
        movl      $3, %ecx
        imull    %eax, %esi
        cmpl      $2, %eax
        leal     1(%rsi), %edx
        je        .L3
        leal     -1(%rax), %ecx
        leal     -1(%rsi), %edx
.L3:
        leal     (%rcx,%rdx), %eax
        addq      $8, %rsp
        ret
```

代码段-2

```
main:
        subq    $8, %rsp
        movq    8(%rdi), %rdi
        xorl    %esi, %esi
        movl    $10, %edx
        call    strtol
        movl    %eax, %ecx
        imull   %eax, %ecx
        cmpl    $2, %eax
        jne     .L2
        leal    1(%rcx), %eax
        movl    $3, %edx
.L3:
        addl    %edx, %eax
        addq    $8, %rsp
        ret
.L2:
        leal    -1(%rax), %edx
        leal    -1(%rcx), %eax
        jmp     .L3
```