

47. 某 C 函数(函数体只有一条 C 语句)的 64 位与 32 位的反汇编结果分别如下:

4005d6: push <u>%rbp</u>	804849b: push <u>%ebp</u>
4005d7: mov <u>%rsp, %rbp</u>	804849c: mov <u>%esp, %ebp</u>
4005da: mov <u>%rdi, -0x8(%rbp)</u>	804849e: mov <u>0x8(%ebp), %eax</u>
4005de: mov <u>-0x8(%rbp), %rax</u>	80484a1: mov <u>(%eax), %eax</u>
4005e2: mov <u>(%rax), %rax</u>	80484a3: lea <u>0x4(%eax), %ecx</u>
4005e5: lea <u>0x4(%rax), %rcx</u>	80484a6: mov <u>0x8(%ebp), %edx</u>
4005e9: mov <u>-0x8(%rbp), %rdx</u>	80484a9: mov <u>%ecx, (%edx)</u>
4005ed: mov <u>%rcx, (%rdx)</u>	80484ab: mov <u>(%eax), %eax</u>
4005f0: mov <u>(%rax), %eax</u>	80484ad: pop <u>%ebp</u>
4005f2: pop <u>%rbp</u>	80484ae: ret
4005f3: <u>retq</u>	

请写出 函数 f 的返回值类型 _____, 参数 p 的类型 _____

函数体的唯一一条 C 语句 _____

49. 设一个 C 语言源程序 p.c, 编译连接后生成执行程序 p, 反汇编如下:

C 程序	反汇编程序的 main 部分(还有系统代码)如下:
#include <stdio.h>	main 的地址为 0x80482C0: (short 占 2 字节).
unsigned short b[2500];	1 movw \$0x3ff, 0x80497d0
unsigned short k;	2 movw 0x804a324, %cx ;k=>cx
void main()	3 mov \$0x801, %eax
{	4 xorw %dx, %dx
b[1000]=1023;	5 div %ecx ;2049/k
b[2000]=2049%k;	6 movw %dx, 0x804a324
b[10000]=20000;	7 movw \$0x4e20, 0x804de20
}	8 ret

现代 Intel 桌面系统, 采用虚拟页式存储管理, 每页 4K, p 首次运行时系统中无其他进程。请阅读如上 C 与汇编程序, 结合进程与虚拟存储管理的知识, 分析:

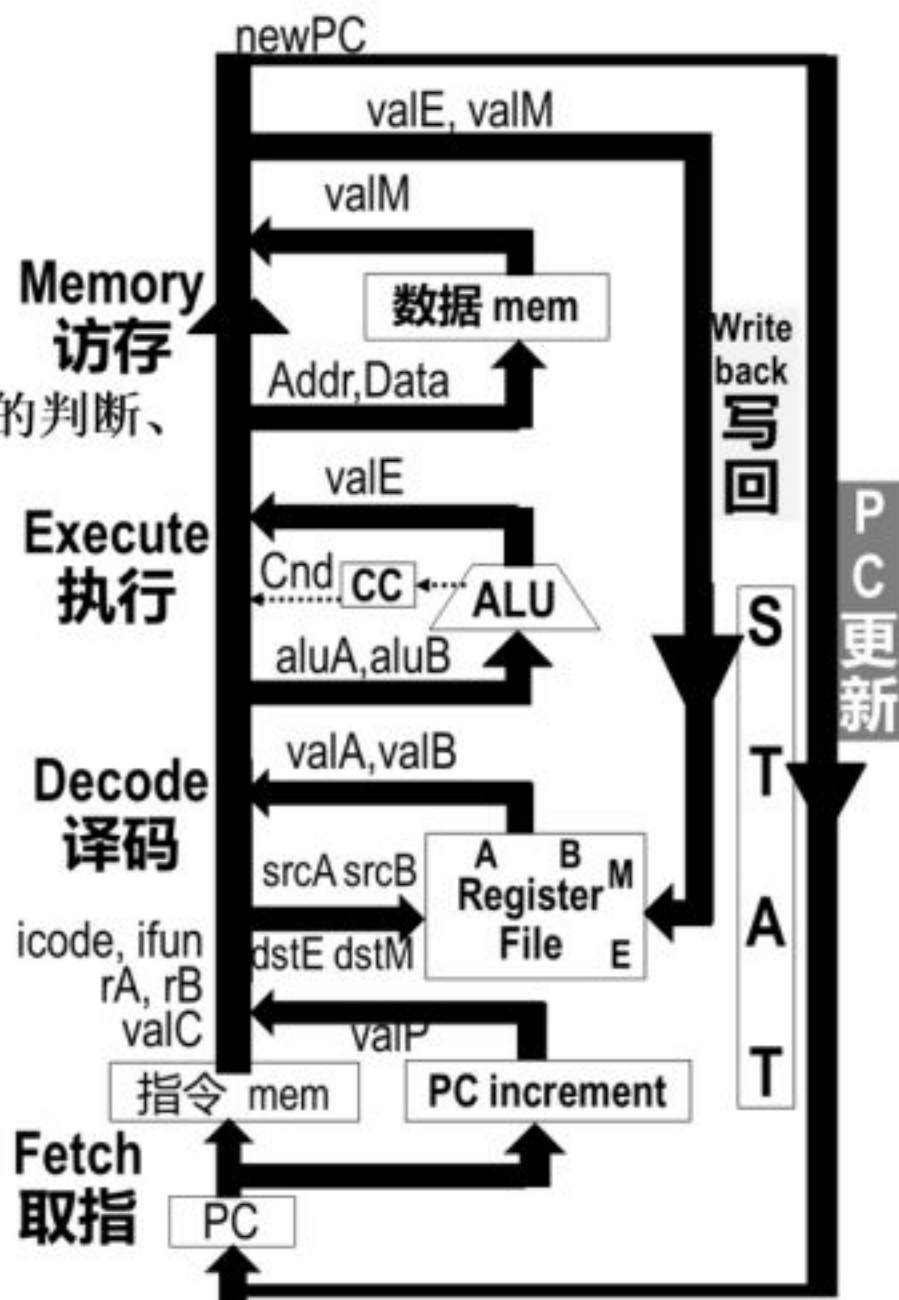
- (1) 上述程序的执行过程中, 在取指令时发生的缺页异常次数为_____.
- (2) 写出已恢复的故障指令序号与故障类型_____.
- (3) 写出没有恢复的故障指令序号与故障类型_____.

50. 计算机的 FPU 是采用堆栈架构实现的（其运算在栈顶附近的数据进行），中间层语言如 MSIL、JavaByteCode 也采用类堆栈 CPU。请按照 Y86-64 的顺序结构实验原理，设计一个 S86-64 的 Stack CPU，完成堆栈的压栈与出栈等基本操作。CPU 要求的指令系统如下：

```
halt:      00
nop:       10
push imm:  20 64 位立即数
push rA:   3|rA
pop  rA:   4|rA
```

注：先期不用考虑堆栈的初始化、空、满的判断、运算的支持等等，以后可逐步扩展指令与标志位等等。且 S86-64 的寄存器与 Y86-64 一样，硬件结构与指令执行的阶段可根据需要进行优化。

- (1) 请写出 POP rA 指令在各阶段的微操作。
- (2) 画出访存阶段的硬件结构图。
- (3) 用 HCL 语言写出存储器地址与数据的控制逻辑。



51. 程序优化： 矩阵 $c[n, n] = a[n, n] * b[n, n]$ ， 采用 48 题 I7 CPU。块 64B。

```
for(int i=0; i<n; i++)  
    for(int j=0; j<n; j++)  
    {  
        c[i, j]=0;  
        for(int k=0; k<n; k++)  
            c[i, j]+=a[i, k]*b[k][j];  
    }
```

请给出基于一般有用的优化、编译器友好、CPU 友好、存储器友好的至少三种优化方法，说明依据，并编写程序。