简答题：

1、 请简要介绍下 MIPS32 下 TLB 的工作原理，以及 page fault 异常与 TLB 缺失异常的区别。

2、有如下的一个函数名为 SwapContext(ucontext_t *o, const ucontext_t *i)，其作用是将当前函数（即调用这个函数的父函数）的运行状态保存在指针 o 指向的连续内存空间中，同时恢复存储在指针 i 指向的内存中的运行状态，并继续运行。

```
SwapContext:
…
movq   %rbx, oRBX(%rdi)
movq   %rbp, oRBP(%rdi)
movq   %r12, oR12(%rdi)
movq   %r13, oR13(%rdi)
movq   %r14, oR14(%rdi)
movq   %r15, oR15(%rdi)

movq   %rdi, oRDI(%rdi)
movq   %rsi, oRSI(%rdi)
movq   %rdx, oRDX(%rdi)
movq   %rcx, oRCX(%rdi)
movq   %r8, oR8(%rdi)
movq   %r9, oR9(%rdi)

movq   (%rsp), %rcx      ①
movq   %rcx, oRIP(%rdi)
leaq   8(%rsp), %rcx     ②
movq   %rcx, oRSP(%rdi)
```

```
movq   oRSP(%rsi), %rsp
movq   oRBX(%rsi), %rbx
movq   oRBP(%rsi), %rbp
movq   oR12(%rsi), %r12
movq   oR13(%rsi), %r13
movq   oR14(%rsi), %r14
movq   oR15(%rsi), %r15

movq   oRIP(%rsi), %rcx ③
pushq  %rcx

movq   oRDI(%rsi), %rdi
movq   oRDX(%rsi), %rdx
movq   oRCX(%rsi), %rcx
movq   oR8(%rsi), %r8
movq   oR9(%rsi), %r9

movq   oRSI(%rsi), %rsi
xorl   %eax, %eax

ret
```

请回答如下问题：

（1） 上文中的汇编语句①及其之后的那一条语句的作用是？
（2） 上文中的汇编语句②及其之后的那一条语句的作用是？
（3） 上文中的汇编语句③及其之后的那一条语句的作用是？
（4） 结构 ucontext_t 的 size 是至少多少字节？
（5） 为何不保存与恢复 r10/r11/rax?

3、 有如下的 C 代码及其对应的 X86-64 汇编代码，请问
（1）局部变量 result 如何存储？
（2）i 如何存储？
（3）EXPR1、EXPR2、EXPR3、EXPR4、EXPR5 分别是？请用常数或者 C 程序中的变量表示。

```
long int puzzle(int a, int b)
{
int i;
long int result = EXPR1;
for (i = EXPR2; i > EXPR3; i -= EXPR4)
result *= EXPR5;
return result;
}
```

```
puzzle:
 movslq %esi,%rdx
 jmp .L60
.L61:
 movslq %edi,%rax
 subl %esi, %edi
 imulq %rax, %rdx
.L60:
 testl %edi, %edi
 jg .L61
 movq %rdx, %rax
 ret
```

4、假设存在如下的完成计数任务的 mips32 汇编代码（左侧框图内），被两个同时运行的任务调用，且这两个任务代码中的地址 65540($4)指向同一个物理内存地址，为确保代码能够正确的实现程序语义，需要替换原始代码中的两条指令，如何替换？此外，汇编器将现有的左侧代码转换为了右侧框图内的等价指令，请填空。

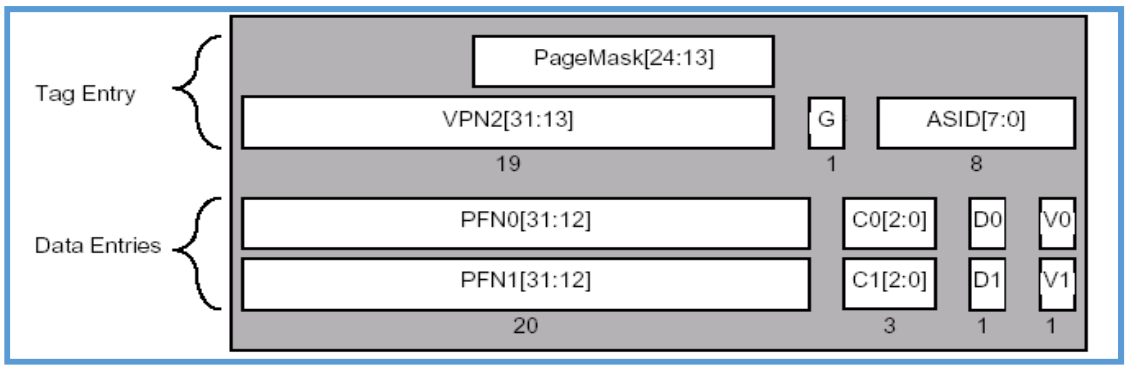
```
atomic_inc:
   lw     $2, 65540($4)
   addiu $2, $2, 1
   sw      $2, 65540($4)
   beq    $2, 0, atomic_inc
   nop
   jr $31
   nop
```

```
atomic_inc:
   lui $1,  _______
   addu $1,  _____,  _____
   lw $2, _____($1)
   addiu $2, $2, 1
   ……
```

5、IEEE 754-2008 标准中支持半精度浮点，exp 位数是 5，frac 位数是 10，符号位数为 1。其所能表示的最大规格化数的 exp 是（　　），frac 是（　　），数值是（　　）。如果存在一个如左图所示的 C 语言 union（联合）数据结构（在 x86-32 机器上，且假设 fp16 表示半精度浮点数类型），那么当 f 被赋予上述这个半精度数值后，s 的值为（　　）。

```
union {
  fp16 f;
  short s;
}
```

6、课上所讲的 MIPS 4KC 处理器的 JTLB 项结构如下所示：



PageMask 固定为全 0，即 page size 为 4KB。
其中 VPN2 项（19 位）是该表项结构的 Tag 部分主体，表示 Virtual Page Num，G（1 位）表示该页是否为全局页面（1 为全局页面），ASID（8 位）表示该页所属的任务号。PFN0/PFN1（均为 20 位）表示对应的物理页面号，即 physical page num，D0/1（1 位）表示该对应物理页是否可写（1 为可写），V0/1（1 位）表示该对应物理页是否有效（1 为有效）。C0/1 本例中不做考虑。
已知 JTLB 中以下表项：

	VPN2	G	ASID	PFN0	PFN1	D0/D1	V0/V1
1	0x47157（二进制为 100　0111 0001 0101 0111)	0	0x12	0x12345	0x12340	0/0	1/1
2	0x47157（二进制为 100　0111 0001 0101 0111)	0	0x13	0x22346	0x22340	1/1	1/1
3	0x4715E（二进制为 100　0111 0001 0101 1110）	0	0x14	0x32345	0x32341	1/1	1/1
4	0x4715E（二进制为 100　0111 0001 0101 1110)	0	0x12	0x22345	0x32340	0/0	1/1

请按照 TLB 的虚拟地址----->物理地址的转换流程，给出下列虚拟地址（包括其对应的 ASID）对应的物理地址，如果不命中（或者是任何引发 TLB 异常的），结果直接写 miss

	虚拟地址	ASID	类型	物理地址(16 进制)
1	0x 8E2AE 320	0x12	写	
2	0x8E2AF 324	0x13	写	
3	0x 8E2BC 330	0x14	写	
4	0x 8E2BD 340	0x19	写	