作业 2

1．假设某个 C 语言函数 func 的原型声明如下：

```
void func(int *xptr, int *yptr, int *zptr);
```

函数 func 的过程体对应的机器级代码用 AT&T 汇编形式表示如下：

```
1    movl    8(%ebp), %eax
2    movl    12(%ebp), %ebx
3    movl    16(%ebp), %ecx
4    movl    (%ebx), %edx
5    movl    (%ecx), %esi
6    movl    (%eax), %edi
7    movl    %edi, (%ebx)
8    movl    %edx, (%ecx)
9    movl    %esi, (%eax)
```

请回答下列问题或完成下列任务：

（1）过程体开始时三个入口参数对应实参所存放的存储单元地址是什么？（提示：当前栈帧底部由帧指针寄存器 EBP 指示）

（2）根据上述机器级代码写出函数 func 的 C 语言代码。

2．已知 IA-32 是小端方式处理器，根据给出的 IA-32 机器代码的反汇编结果（部分信息用 x 表示）回答问题：

（1）已知 je 指令的操作码为 01110100，je 指令的转移目标地址是什么？call 指令中的转移目标地址 0x80483b1 是如何反汇编出来的？

```
804838c:    74 08              je xxxxxxx
804838e:    e8 le 00 00 00     call 80483b1<test>
```

（2）已知 jb 指令的操作码为 01110010，jb 指令的转移目标地址是什么？movl 指令中的目的地址如何反汇编出来的？

```
8048390:    72 f6              jb xxxxxxx
8048392:    c6 05 00 a8 04 08 01    movl $0x1, 0x804a800
8048399:    00 00 00
```

（3）已知 jle 指令的操作码为 01111110, mov 指令的地址是什么？

```
xxxxxxx:    7e 16              jle 80492e0
xxxxxxx:    89 d0              mov %edx, %eax
```

（4）已知 jmp 指令的转移目标地址采用相对寻址方式，jmp 指令操作码为 11101001，其转移目标地址是什么？

```
8048296:    e9 00 ff ff ff     jmp xxxxxxx
804829b:    29 c2              sub %eax, %edx
```

3．已知函数 f1 的 C 语言代码框架及其过程体对应的汇编代码如下所示，根据对应的汇编代码填写 C 代码中缺失的部分，并说明函数 f1 的功能。

C 语言代码：

```
1    int f1(unsigned  x)
2    {
3      int y = 0;
```

```
4      while( _____________ ) {
5        _____________ ;
6      }
7      return _____________ ;
8    }
```

汇编代码：

```
1    movl    8(%ebp), %edx
2    movl    $0, %eax
3    testl   %edx, %edx
4    je      .L1
5    .L2:
6    xorl    %edx, %eax
7    shrl    $1, %edx
8    jne     .L2
9    .L1:
10   andl    $1, %eax
```

4．已知函数 funct 的 C 语言代码如下：

```
1    #include <stdio.h>
2    int funct(viod) {
3    int x, y;
4    scanf("%d %d", &x, &y);
5    return x-y;
6    }
```

函数 funct 对应的汇编代码如下：

```
1    funct:
2    pushl   %ebp
3    movl    %esp, %ebp
4    subl    $40, %esp
5    leal    -8(%ebp), %eax
6    movl    %eax, 8(%esp)
7    leal    -4(%ebp), %eax
8    movl    %eax, 4(%esp)
9    movl    $.LC0, (%esp)      ; 将指向字符串"%d %d"的指针入栈
10   call    scanf              ; 假定 scanf 执行后 x = 15，y = 20
11   movl    -4(%ebp), %eax
12   subl    -8(%ebp), %eax
13   leave
14   ret
```

假设函数 funct 开始执行时，R[esp]=0xbc000020，R[ebp]=0xbc000030，指向字符串"%d %d"的指针为 0x804c000。回答下列问题或完成下列任务：

（1）执行第 3、10 和 13 行的指令后，寄存器 EBP 中的内容分别是什么？

（2）执行第 3、10 和 13 行的指令后，寄存器 ESP 中的内容分别是什么？

（3）局部变量 x 和 y 所在存储单元的地址分别是什么？

（4）画出执行第 10 行指令后 funct 的栈帧，指出栈帧中的内容及其地址。

5．假设函数 sumij 的 C 代码如下，其中 M 和 N 是用#define 声明的常数。

```
1    int    a[M][N], b[N][M];
2
3    int sumij(int i, int j) {
4      return a[i][j]+b[j][i];
5    }
```

已知函数 sumij 的过程体对应的汇编代码如下：

```
1    movl      8(%ebp), %ecx
2    movl      12(%ebp), %edx
3    leal      (, %ecx, 8), %eax
4    subl      %ecx, %eax
5    addl      %edx, %eax
6    leal      (%edx, %edx, 4), %edx
7    addl      %ecx, %edx
8    movl      a(, %eax, 4), %eax
9    addl      b(, %edx, 4), %eax
```

根据上述汇编代码，确定 M 和 N 的值。

6．假设函数 trans_matrix 的 C 代码如下，其中 M 是用#define 声明的常数。

```
1    void trans_matrix(int a[M][M]) {
2      int i, j, t;
3      for (i = 0; i<M; i++)
4        for(j = 0; j<M; j++){
5          t = a[i][j];
6          a[i][j] = a[j][i];
7          a[j][i] = t;
8        }
9    }
```

已知采用优化编译(选项-O2)后函数 trans_matrix 的内循环对应的汇编代码如下：

```
1    .L2:
2    movl      (%ebx), %eax
3    movl      (%esi, %ecx,4), %edx
4    movl      %eax, (%esi, %ecx, 4)
5    addl      $1, %ecx
6    movl      %edx, (%ebx)
7    addl      $76, %ebx
8    cmpl      %edi, %ecx
9    jl        .L2
```

根据上述汇编代码，回答下列问题或完成下列任务：

（1）M 的值是多少？常数 M 和变量 j 分别存放在哪个寄存器中？

（2）写出上述优化汇编代码对应的函数 trans_matrix 的 C 代码。

7．假设联合类型 utype 的定义如下：

```
typedef union {
 struct {
   int x;
   short y;
   short z;
 } s1
 struct {
   short a[2];
   int b;
   char *p;
 } s2
} utype;
```

若存在具有如下形式的一组函数：

```
void getvalue(utype *uptr, TYPE *dst) {
    *dst = EXPR;
}
```

该组函数用于计算不同表达式 EXPR 的值，返回值的数据类型根据表达式的类型确定。假设函数 getvalue 的入口参数 uptr 和 dst 分别被装入寄存器 EAX 和 EDX 中，仿照例子填写下表，说明在不同的表达式下的 TYPE 类型以及表达式对应的汇编指令序列（要求尽量只用 EAX 和 EDX，不够用时再使用 ECX）

题 13 表

表达式 EXPR	TYPE 类型	汇编指令序列
uptr->s1.x	int	movl (%eax), %eax movl %eax, (%edx)
uptr->sl.y		
&uptr->s1.z		
uptr->s2.a		
uptr->s2.a[uptr->s2.b]		
*uptr->s2.p		

8．函数 lproc 的过程体对应的汇编代码如下：

```
1     movl     8(%ebp), %edx
2     movl     12(%ebp), %ecx
3     movl     $255, %esi
4     movl     $-2147483648, %edi
5     .L3:
6     movl     %edi, %eax
7     andl     %edx, %eax
8     xor1     %eax, %esi
9     movl     %ecx, %ebx
10    shrl     %bl, %edi
11    testl    %edi, %edi
12    jne      .L3
13   movl      %esi, %eax
```

上述代码根据以下 lproc 函数的 C 代码编译生成：

```
1    int
2    lproc(int x, int k) {
3      int val = _____ ;
4      int i;
5      for (i = _____ ; i _____ ; i = _____ ) {
6        val ^= _____ ;
7      }
8      return val;
9    }
```

回答下列问题或完成下列任务：

（1）给每条汇编指令添加注释。

（2）参数 x 和 k 分别存放在哪个寄存器中？局部变量 val 和 i 分别存放在哪个寄存器中？

（3）局部变量 val 和 i 的初始值分别是什么？

（4）循环终止条件是什么？循环控制变量 i 是如何被修改的？

（5）填写 C 代码中缺失的部分。

9．假设嵌套的联合数据类型 node 声明如下：

```
1    union node {
2      struct {
3        int *ptr;
4        int datal;
5      } n1;
6      struct {
7        int data2;
8        union node *next;
9      } n2;
10   };
```

有一个进行链表处理的过程 chain_proc 的部分 C 代码如下：

```
1    void chain_proc(union node *uptr) {
2      uptr-> _____ = *(uptr-> _____ ) - uptr-> _____ ;
3    }
```

过程 chain_proc 的过程体对应的汇编代码如下：

```
1    movl    8(%ebp), %edx
2    movl    4(%edx), %ecx
3    movl    (%ecx), %eax
4    movl    (%eax), %eax
5    subl    (%edx), %eax
6    movl    %eax, 4(%ecx)
```

回答下列问题或完成下列任务：

（1）node 类型中结构成员 n1.ptr、n1.data1、n2.data2、n2.next 的偏移量分别是多少？

（2）node 类型总大小占多少字节？

（3）根据汇编代码写出 chain_proc 的 C 代码中缺失的表达式。

提交作业
作业（word 或 pdf 格式）拷贝到一个文件夹中，命名为：
作业 n
其中，n=1…3 为第 n 次作业
课程结束时，将 3 次作业与 6 次实验拷贝到同一个文件夹中，命名为如下格式：
班号-学号-姓名
以班为单位一起提交。