考试中心填写：

___年___月___日
考试用

湖南大学课程考试试卷

课程名称：计算机系统(2018 春)；试卷编号：A；考试时间：120 分钟

姓名：学号：专业班级：

题号	一	二	三	四	五	六	七	八	九	十	总分
应得分	10	8	32	10	20	20					100
实得分											
评卷人											评分：

注意：请在答题纸上作答，做在试卷上无效。

一、选择题（每题 2 分，共 10 分）

1. 0x12345678 存放在采用小端存储的机器上，地址为 0x100 到 0x103，则 0x101 处存放值为 (　　)。

A. 0x12　　B. 0x34　　C. 0x56　　D. 0x78

2. 定点数运算产生溢出的原因是(　　)

A. 运算过程中最高位产生的进位或错

B. 参加运算的操作数超出了机器的表示范围

C. 运算结果超出了机器的表示范围

D. 寄存器的位数太少，不得不舍弃最低有效位

3. 3×4096+14×256+5×16+17 计算结果的二进制表示包含多少个 1？(　　)

A. 8　　B. 9　　C. 10　　D. 12

4. a=01101001，b=01010101，则 a&b = (　　)

A. 01000001　　B. 10110101　　C. 01011111　　D. 11011010

5. 对于指令 leal 7(%ebx,%edx,5),%eax，假设 %ebx 的值为 y，%edx 的值为 x，则在执行指令后%eax 的值为(　　)。

A. 5*y+x+7　　B. 5*x+y+7　　C. 7*y+x+5　　D. 7*x+y+5

二、简答题（8 分）

某公司需要将 4 个有符号字节封装成一个 32 位 unsigned，一个字中的字节从 0（最低有效字节）编号到 3（最高有效字节）。要求为一个使用补码运算和算术右移的机器编写一个具有如下原型的函数：

```
typedef    unsigned packed_t;
int xbyte(packed_t word, int bytenum);
//抽取出指定的字节，再符号扩展为一个 32 位的 int
```

小明由于完成《深入理解计算机系统》作业时不认真，编写了如下错误代码：

```
int xbyte(packed_t word,    int bytenum)
{
    return (word>>(bytenum<<3)) & 0xFF;
}
```

请问：

1．小明写的代码错在哪里？（4 分）

2．给出函数的正确实现，只能使用左右移位和一个减法。（4 分）

三、程序分析题（32 分）

1. 下面的 C 程序是对两个整数数组的元素进行某些操作，请参照其对应的 32 位环境下汇编语言代码将 C 程序补充完整。(每空 2 分，共 10 分)

```
#include "stdio.h"
void main()
{
    int array1[10]={1,5,3,4,7,6,8,10,9,2}, array2[10];
    int i=1, j=2;
    for (_____;_____;_____)
    {
        _____;
    _____;
    }
}
```

其所对应汇编代码如下：

```
main:
    pushl    %ebp
    movl     %esp, %ebp
    andl $-16, %esp
    subl $112, %esp
    …… //数组赋值语句
```

```
        movl    $1, 108(%esp)
        movl    $2, 104(%esp)
        movl    $0, 108(%esp)
        movl    $5, 104(%esp)
        jmp  .L2
.L4:
        movl    -4(%ebp), %eax
        movl    -8(%ebp), %edx
        movl    -48(%ebp,%edx,4), %edx
        movl    %edx, -88(%ebp,%eax,4)
        movl    -8(%ebp), %eax
        movl    -4(%ebp), %edx
        movl    -48(%ebp,%edx,4), %edx
        movl    %edx, -88(%ebp,%eax,4)
        addl    $1, -4(%ebp)
        addl    $1, -8(%ebp)
.L2:
        cmpl    $4, -4(%ebp)
        jg      .L5
        cmpl    $9, -8(%ebp)
        jle     .L4
.L5:
        leave
        ret
```

2. 现在的 C 程序中的矩阵计算函数 f 对两个矩阵的值进行了操作，请根据此 C 程序的 32 位环境下汇编代码将函数 f 补充完整。（8 分）

```
#include "stdio.h"
#define H 4
#define J 17
int A[H][J];
int B[J][H];
int C[H][H];

void f(int x, int y, int z)
{
        int i=0;
        for(i=0; i<z; i++)
                _____;
}
int main()
{
        return 0;
}
```

f 函数对应汇编代码如下：（已为 A,B,C 数组分配好内存空间，代码中的 A，B，C 分别表示数组的首地址）

```
f:
        pushl   %ebp
        movl    %esp, %ebp
        pushl   %edi
        pushl   %esi
        pushl   %ebx
        subl    $16, %esp
        movl    $0, -16(%ebp)
        movl    $0, -16(%ebp)
        jmp  .L2
.L3:
        movl    8(%ebp), %ebx
        movl    12(%ebp), %ecx
        movl    8(%ebp), %edx
        movl    12(%ebp), %eax
        sall    $2, %edx
        leal    (%edx,%eax), %eax
        movl    C(,%eax,4), %esi
        movl    8(%ebp), %edx
        movl    -16(%ebp), %edi
        movl    %edx, %eax
        sall    $4, %eax
        addl    %edx, %eax
        addl    %edi, %eax
        movl    A(,%eax,4), %edx
        movl    -16(%ebp), %edi
        movl    12(%ebp), %eax
        sall    $2, %edi
        leal    (%edi,%eax), %eax
        movl    B(,%eax,4), %eax
        imull   %eax, %edx
        movl    -16(%ebp), %edi
        movl    12(%ebp), %eax
        sall    $2, %edi
        leal    (%edi,%eax), %eax
        movl    B(,%eax,4), %eax
        imull   %edx, %eax
        leal    (%esi,%eax), %edx
        leal    0(,%ebx,4), %eax
        addl    %ecx, %eax
        movl    %edx, C(,%eax,4)
```

```
        addl    $1, -16(%ebp)
.L2:
        movl    -16(%ebp), %eax
        cmpl    16(%ebp), %eax
        jl      .L3
        addl    $16, %esp
        popl    %ebx
        popl    %esi
        popl    %edi
        popl    %ebp
        ret
```

3. 请仔细阅读如下 C 语言代码及对应的 32 位环境下汇编代码。

```
int bar (int a, int b) {
return a + b;
}
int foo(int n, int m, int c) {
c += bar(m, n);
return c;
}
```

```
08048374 <bar>:
8048374:    55              push %ebp
8048375:    89 e5           mov %esp,%ebp
8048377:    8b 45 0c        mov 0xc(%ebp),%eax
804837a:    03 45 08        add 0x8(%ebp),%eax
804837d:    5d              pop %ebp
804837e:    c3              ret
0804837f <foo>:
804837f:    55              push %ebp
8048380:    89 e5           mov %esp,%ebp
8048382:    83 ec 08        sub $0x8,%esp
8048385:    8b 45 08        mov 0x8(%ebp),%eax
8048388:    89 44 24 04     mov %eax,0x4(%esp)
804838c:    8b 45 0c        mov 0xc(%ebp),%eax
804838f:    89 04 24        mov %eax,(%esp)
8048392:    e8 dd ff ffff   call 8048374 <bar>
8048397:    03 45 10        add 0x10(%ebp),%eax
804839a:    c9              leave
804839b:    c3              ret
```

假设我们调用了函数将 foo(3, 4, 5)，画出从调用 foo 函数开始，到刚刚执行完 bar 函数 ret 指令时的栈帧图。

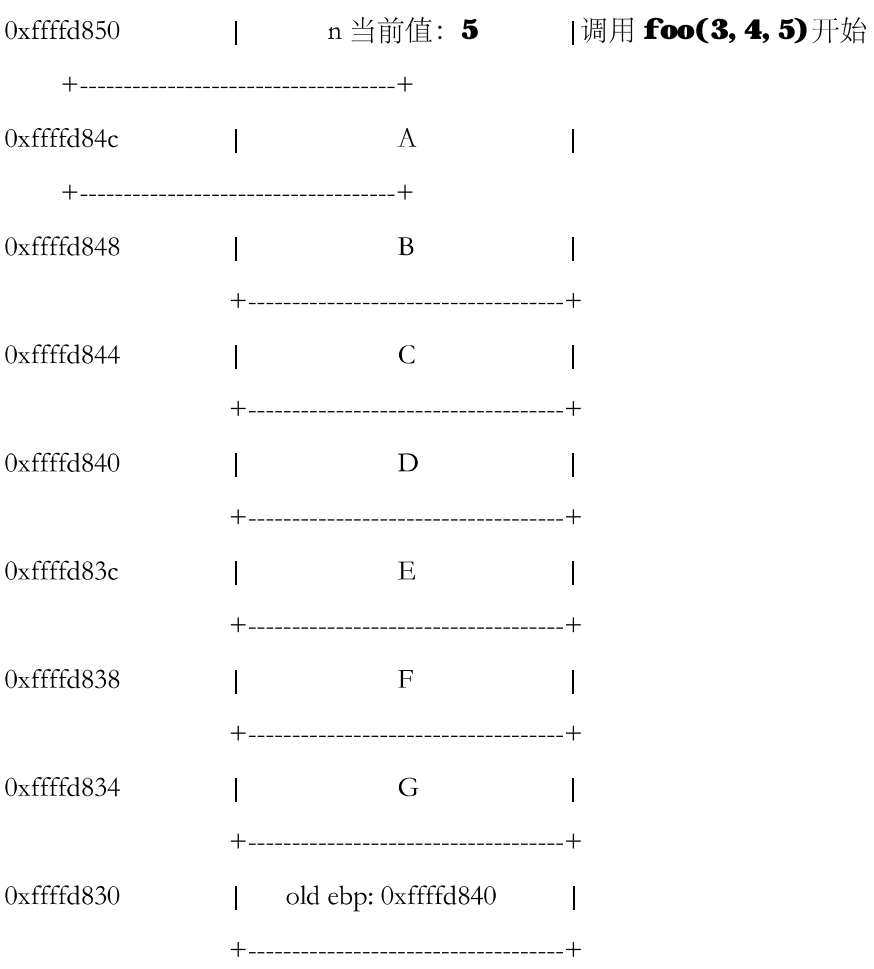
在调用 foo 函数时，%ebp 值为 0xffffd858，返回地址为：0x080483c9

要求：

1） 如果为变量，则写出具体的值

2） 如果是%ebp 值，则需要标明，例如：%ebp0xffffd858

3） 如果是返回地址，则需要标明，例如: 返回地址 0x080483c9



（每空 2 分，共 14 分）

A. _____

B. _____

C. _____

D. _____

E. _____

F. _____

G. _____

四、简述题（10 分）

请简述 Amdahl 定律的基本思想。假设我们可以将某个计算系统中 50%的部分速度提高到其计算时间可以忽略不计的话，那么系统的加速比为多少？从这个例子中可以说明什么实际意义？

五、（20 分）假设主存按字节编制，直接映射 cache，块大家为 512 字节。cache 容量为 8K 字节，主存大小为 1M 字节。问：

主存地址如何划分？请用图形的方式画出主存块和 cache 之间的映射关系；若当前 cache 为空，对地址 0230CH 的访问过程。

六、（20分）现有一个包含一个TLB 和L1 d-cache（16行，每块大小为4Byte，直接映射）的小系统，其存储器按字节寻址，存储器访问针对1个Byte（8bit）的字，其虚拟地址长度为n=14bit，物理地址长度为m=12bit，页面大小为P=64Byte，若有虚拟地址**0X0040**和虚拟地址**0X03A9**，分别回答如下问题：

1. 虚拟地址格式是什么？（可直接画出并填空）VPN 和VPO 是什么意思？该虚拟地址中哪些位表示VPN？哪些位表示VPO？

2. 是否使用TLB对系统有什么影响？若TLB 采用4 路组相联，其TLB 索引和标记格式多少？并请判断TLB 是否命中？是否缺页？