1.数字进行类型转换，输出转化后的结果

D

2.补码的计算，整数正负绝对值相差1,

A

3.对于整数分别进行除以和右移的操作，有符号的右移是算术右移，所以要注意区别

c语言的除法，向零取整，而右移是向下取整

D

4.判断数字的大小

A.-1在无符号中是最大的，错

B.对的

C.两个整数相乘，如果溢出可能负

D.int转成float可能会有信息丢失，因为float位数少

5.栈帧的组成

A.反过来了

C.栈是向下增长的

6.

就是小端法

7.求一个数的负数

取反+1即可，除了符号位都是对应的

8.不考

9.fork：调用一次返回两次，子进程的返回

10.缓存的计算

16，4次方，h是四位ci，32，2的五次方得不到，五个co

剩下的是ct

浮点数计算

k阶码，n个frac位数

非规格化：e全为0，e=1-bias，大小直接frac的大小然后多少次幂

最大非规格化数：frac全1，则frac为1-2（-。。）

最小非规格化数：frac最后一位是1

规格化：e不全为0，不全为1，e等于exp-bias，然后1+frac再次幂

最小规格化数：后面全0

最大非规格化数：后面全1

e全1：m为0：正负无穷

计算时，先利用二进制计算出整数（进行约化），然后判断是规格化数还是非规格化

再进行下一步的计算

如15/16,0.1111：先判断规格化数，然后计算得出是2的-1次幂乘1.1111

便可得出e和frac

在乘2的多少次幂是，可以用左右移小数点的方法，先直接变二进制，然后移动小数点到m允许的范围，然后便可得出表示

超出范围直接无穷，编码也就是无穷的写法

函数的运行，栈帧的变化

画出栈帧即可

Leave和ret

先leave再ret

Leave：mov %ebp,%esp

Pop %ebp

也就是ebp的大小

Ret: push %esp

由于有无用区域，所以虽然溢出但是没有造成损坏

Mn，汇编代码计算即可

Ecx：i；ebx：j，先的数字小

关键路径的计算

CPE的计算

首先根据计算的方法画出关键路径的图（最后赋值到的值是关键的）



直线三个乘法：3\*3，除以三个元素=》3



关键路径两个乘法：2\*3/3=2



关键路径仅有一个乘法（其他的都是并行的）

（计算乘上等待时间即可）

4,5即可

问题六

命中不命中

判断参数

Int字节数

储存从0开始，第一行

一个block8个字节：两个int

Set16，也就是两块

也就是两行，一行两个

对应到数组中元素时，按顺序移动即可

接着判断访问方式

m不命中h命中

不命中更新，添加数据，命中继续进行访问

缓存大小加倍：由于块的大小没变，所以多了两行

还是同样步骤

问题七

链接

本地变量和全局变量

首先a是本地变量

b是2，初始化的强变量

c是未初始化的弱变量

Foo中

Abc均为未初始的弱变量

a局部不影响a，局部不会变

Foo中的bc若会被main中的b强变量覆盖，c也是，不过都弱，所以不会影响

注意foo中运行时由于b已经被main中的覆盖，所以实质是main中的b变了

指向short指针的指针p

Short指针加200：内存中\*p一次指两块，加1后移两块

移动200：加400

a是long\*，所以a

第八题

Fork的运行

多次判断输出字符串

判断字符串有无可能输出

首先从程序入口进入

直接第一个判断

由于fork所以一个a，一个b

b直接输出2终止

a再进行判断：一个c，一个d

c直接输出3，然后跳出来打印0，终止

d的话是一个wait，等待另一个子进程执行完才会执行下一步4，然后再跳出输出0即可

一定3040

第九题

地址翻译{{{{{再看再看}}}}}

0x7bcd8001

先转化为二进制

后面3个xpo，前面五个是vpn

Vpn的前十个是一块：dir idx

剩下的是pte iex

二级页表的起始地址：