3.54

int decode2(int x,int y,int z)

{

z-=y;

int t=z;

t<<=15; //t=t<<15,算术左移

t>>=15; //t=t>>15,算术右移

z=z^x; //异或操作

return z\*t;

}

3.56

A：x ->%esi n->%ebx result->%edi mask->%edx

B:初始值 result=1431655765 mask=-2147483648

C：mask测试条件 mask!=0;

D:修改mask： ((unsigned)mask)>>(n%256))

E：修改result：result ^= (mask & x);

F:代码如下

int loop(int x, int n)

{

int result = 1431655765;

   int mask;

for(mask = 1<<31; mask != 0; mask = ((unsigned)mask)>>(n%256)))

{

result ^= (mask & x);

}

return result;

}

解析：其中需要特别注意的语句是 shrl %cl,%edx, %cl表示的是寄存器ecx的后八位，这里即n的后八位，因此这里右移的是n%256.因为mask的初始值为有符号的最小数，为负数，这里又是逻辑右移所以应将mask强制转换为unsigned类型。

3.58

typedef enum {MODE\_A, MODE\_B, MODE\_C, MODE\_D, MODE\_E} mode\_t;

Int switch3(int \*p1,int \* p2,mode\_t action)

{

Int result =0;

switch(action)

{

case MODE\_A: result=\*p1; \*p1=\*p2; break;

case MODE\_B: result=\*p1+\*p2; \*p2=result; break;

case MODE\_C: \*p2=15; result=\*p1; break;

case MODE\_D: \*p2=\*p1; result=17; break;

case MODE\_E: result=17; break;

default: result=-1; break;

}

return result;

}

3.62

A: M=76/4 根据指令addl $76,%ebx;又因为每个数有4个字节。

B：i->%edi j->%ecx

C:代码如下

 int transpose(int M, int A[M][M])

{

    int i,j;

for(i=0; i<M; ++i)

  {

       int \*a = &A[i][0];

int \*b=&A[0][j];

  for(j=0; j<i; ++j)

{

int t = \*a;

\*a = \*b;

\*b = t;

++a;

b += M;

   }

  }

}

3.68

根据分析可知 mov 0xc8(%ecx),%eax,是在计算bp->right，由mov 0xc(%ebp),%ecx知%ecx保存了bp->left，size表示a\_struct a[CNT]的大小。则CNT\*size=0xc8-0x4=0xc4，即196

由mov %eax,0x8(%ecx,%edx,4) 可知ap->x[ap->idx]的地址为0x8+%ecx+%edx\*4，%ecx存储的是bp的地址。

由\*ap=&bp->a[i];知ap的地址是bp+4+i\*size。ap->x[0]的地址是bp+4+i\*size+pos(idx),pos(idx)为a\_struct中idx的所占字节

则那么ap->x[ap->idx] 的地址是 bp + 4 + i\*size + pos(idx) + 4\*(ap->idx)

则 %edx\*4= 4 + i\*size + 4\*(ap->idx)

注意到在代码第10行add 0x4(%ecx,%ebx,1),%edx可知%edx=7\*i+ bp[4 + 28\*i]

猜测bp[4 + 28\*i]是bp中的a[i]的首地址,则则size=28, 且bp[4+28\*i]的值为ap->idx

用%edx\*4= 4 + i\*size + 4\*(ap->idx)代入检测可知正确

则size=28

A:CNT=196/28=7,x[]的length=（28-4）/4=6

B:typedef struct { int idx;

Int x[6]; }