#include <stdio.h>  
#define N 1000  
struct instruct{  
 char operate[5]; //操作  
 int pageid1; //页号  
 int address; //页内地址  
}  
 instruct[N]={{"+",0,70},{"+",1,50},{"\*",2,15},{"cun",3,21},{"qu",0,56},{"-",6,40},{"yi",4,53}, {"+",5,23}, {"cun",1,37}, {"qu",2,78}, {"+",4,1}, {"cun",6,84}};  
struct page{  
 int pageid; //页号  
 int flag; //标志  
 int yejiaid; //页架号  
 int chageflag; //修改标志  
 int diskplace; //在磁盘上位置  
}page[N];  
int p[4]={0,1,2,3};  
void init(){  
 int i;  
 for(i=0;i<7;i++)  
 {  
 page[i].pageid=i;  
 page[i].flag=1;  
 page[i].chageflag=0;  
 }  
 for(i=4;i<7;i++)  
 {  
 page[i].flag=0;  
 }  
 page[0].yejiaid=5;  
 page[1].yejiaid=8;  
 page[2].yejiaid=9;  
 page[3].yejiaid=1;  
 page[4].yejiaid=-1;  
 page[5].yejiaid=-1;  
 page[6].yejiaid=-1;  
 page[0].diskplace=11;  
 page[1].diskplace=12;  
 page[2].diskplace=13;  
 page[3].diskplace=21;  
 page[4].diskplace=22;  
 page[5].diskplace=23;  
 page[6].diskplace=121;  
}  
void print1(){  
 printf("\*\*\*\*\*\*\*\*\*\*\*\*作业的页表\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
 int i;  
 for(i=0;i<7;i++)  
 {  
 printf(" 页号: %d ",page[i].pageid);  
 printf(" 页内标志: %d ",page[i].flag);  
 printf(" 页架号: %d ",page[i].yejiaid);  
 printf(" 修改标志: %d ",page[i].chageflag);  
 printf(" 磁盘位置: %d\n",page[i].diskplace);  
 }  
}  
void print2(){  
 printf("\*\*\*\*\*\*\*\*\*\*\*\*作业的执行指令\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
 int i;  
 for(i=0;i<12;i++)  
 {  
 printf("操作为:%s ",instruct[i].operate);  
 printf(" 页号:%d ",instruct[i].pageid1);  
 printf(" 页内标志:%d\n",instruct[i].address);  
 }  
}  
void transform(){  
 printf("\*\*\*\*\*\*\*\*\*\*\*\*作业的执行\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
 int i;  
 int flag1,excpageid,pageid3,changeflag1;//页标志，页面唯一标志，页面id3，修改标志1  
 int k=0;  
 int m=4;  
 long absaddress; //绝对地址  
 for(i=0;i<12;i++){  
 pageid3=instruct[i].pageid1;//页面id3=各功能所对应的页号的id  
 flag1=page[pageid3].flag;//页标志=页内信息的flag  
 printf("操作为: %s ",instruct[i].operate);  
 printf("页号为: %d ",instruct[i].pageid1);  
 printf("页内标志为: %d \n",instruct[i].address);  
 if(flag1==1){  
 absaddress=page[pageid3].yejiaid\*1024+instruct[i].address;//绝对地址=页内的页架号信息\*1024+ 页内地址  
 printf("页号%d",pageid3);  
 printf("中执行\n");  
 if(instruct[i].operate=="cun"){  
 page[pageid3].chageflag=1;//修改标志置为1  
 printf("绝对地址是: %d\n\n",absaddress);  
 }  
 else{  
 printf("绝对地址是: %d\n\n",absaddress);//不是存就直接输出绝对地址  
 }  
 }  
 //模拟FIFO页面调度  
 if(flag1==0){ //产生缺页中断  
 printf(" \*页号(发生缺页中断）\n");  
 excpageid=p[k]; //当出现缺页中断后要装入的新页页号  
  
 changeflag1=page[excpageid].chageflag;//新页页号的修改标志是否为1  
 if(changeflag1==1){  
 printf("修改标志为1\n");  
 printf("调出的页号为: %d ",excpageid);//输出 OUT J  
 printf("要装入的新页页号为:%d\n\n",pageid3);//输出 IN L  
 }  
 else{  
 printf("修改标志为0\n");  
 printf("要装入的新页页号: %d\n\n",pageid3);//输出 IN L  
 }  
 //修改页表  
 page[pageid3].flag=page[excpageid].flag;//各功能所对应的页号id=新页页号的flag  
 page[pageid3].yejiaid=page[excpageid].yejiaid;//各功能所对应的页架号id=新页页号的页架号  
 page[excpageid].yejiaid=-1;  
 page[excpageid].flag=0;  
 p[k]=pageid3;//要装入的新页页号  
 k=(k+1)%m; //当产生缺页中断后，操作系统总是选择P[K]所指出的页面调出  
 page[excpageid].chageflag=0;//新页的修改标志  
 page[pageid3].chageflag=1; //各功能所对应的修改标志  
 //修改结束之后回到查页表，判断页标志  
 }  
 }  
}  
void main(){  
 init();  
 print1();  
 print2();  
 transform();  
}



