Khoa Công nghệ Thông tin

Bộ môn **HỆ THỐNG NHÚNG**

Môn học: KỸ THUẬT SỐ & ƯD

Viết chương trình xử lý hệ đếm gồm các công việc sau:

*Hàm ***input()** để nhập số thập lục phân(hex) int **it**; hàm cho phép xóa dấu âm và chữ số hex(0⇔9, A⇔F, a⇔f) khi gõ nhầm bằng phím Backspace, hàm không cho hiển thị ký tự khác chữ số(hex) lên màn hình, đồng thời hàm cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị: -0x8000(-32768)⇔0X7Fff(32767). Đổi số int **it** ra chuỗi nhị phân dài 16 ký tự bit(char *sb). Hàm **bu1_2()** thực hiện bù 1 và bù 2 chuỗi char*sb thành chuỗi char*bu2. Đổi chuỗi char*bu2 ra số int **gt; (ta thấy gt là số đối của it**). Đổi chuỗi char*bu2 ra chuỗi số hệ cơ số 16(char*sh).Đổi chuỗi char*sh ra chuỗi số hệ cơ số 8(char*oct). Đổi chuỗi char*oct ra int gt. Đổi gt ra chuỗi số hệ cơ số 14(char*s14) và ngược lại, rồi thực hiện bù 2 số int gt để có lại số nhập ban đầu int **it.***

*//---------- **Khai báo các tệp tiêu đề của C: có hàm,kiểu, hằng,…chuẩn mà ta sẽ dùng trong chương trình** ------*

#include< conio.h> *// các hàm,hằng,… liên quan đến hiển thị: clrcsr(), getch(),..*

#include< stdio.h> *// các hàm, hằng,.. nhập xuất: putchar(),printf(),..,các hàm xử lý tệp tin: *fopen(), fscanf(),...*

#include< string.h> */* các hàm, hằng,.. xử lý chuỗi và mảng: *strcat(), *strcpy(), strlen(),..Ta sẽ viết hàm dai(), để tính độ dài thật của chuỗi ký tự thay hàm chuẩn strlen(), nên có thể không cần khai báo tệp nầy.*/*

#include< stdlib.h>*// các hàm liên quan đến cấp phát động: *malloc(), *calloc(),.. các hàm chuyển đổi: atoi(), isdigit()...*

#include< dos.h> *// các hàm, kiểu,…, liên quan đến xử lý ngắt: int86(), intdos(), union REGS…*

//---------- *Khai báo các hàm mà ta tự định nghĩa* ----------

int*input(); *// hàm trả về địa chỉ biến kiểu int*

int dai(char*s); *// hàm trả về giá trị biến kiểu int*

void in_bit(int n); *// hàm không trả về giá trị*

char *bu1_2(int it, int*gt); *// hàm trả về địa chỉ đầu của vùng nhớ lưu trữ chuỗi*

char* bin_hex(char *sb, int*gt); char *dao(char* s);

void hex_oct(char*sh, int*gt); void dec_he14(int gt, int*it);

//----------*hàm chính main()*----------

main(){ int gt, i, it; char *sh, *s14,,*bu2;

int a=-100, b=-0x64, c=-0144, d=0xff9C, e=0177634; char f=0x9C,g=0234; long h=0xffffff9C;

clrscr(); printf(" a=%d, b=%d , c=%d, d=%d, e=%d, f=%d, g=%d, h=%d\n",a,b,c,d,e,f,g,h);

/ Ta đã gán cùng 1 giá trị -100 cho các biến a=b=c=d=e=f=g=h=-100 . Số âm có 2 cách biểu diễn : theo quy cách số bù 2 hoặc theo quy cách đặt dấu – phía trước.Hãy xem tiếp cách lập trình sau đây */.*

bu2= (char *)malloc(17*1); s14= (char *)calloc(6, 1); sh=(char*)malloc(8*1);

*/*Chuỗi nhị phân được đổi ra từ số int sẽ dài 16 ký tự bit + 1 ký tự kết thúc chuỗi '\0'(NULL), theo đó tính ra chiều dài cho các chuỗi số hệ thập lục phân thêm"0X" đứng trước (char* sh), chuỗi số hệ thập tứ phân(char s14), chuỗi số hệ bát phân thêm "0" đứng trước (char*oct)….*/*

printf("So int = "); it=*input(); */* khi nhập số hex, có thể nhập 0X trước hoặc không vd:nhập 143, ta gõ 0x8F(Enter) hoặc gõ 8f(Enter), hoặc nhập -100, ta gõ -0x64(Enter) hoặc -64(Enter)*/*

printf("So nhap= %d = ",it); in_bit(it); printf("= 0%o = 0x%x\n",it,it);

/ In số int it vừa nhập dưới dạng số thập phân, nhị phân, bát phân, thập lục phân, để so sánh với kết quả lập trình. Riêng in số nhị phân và số hệ thập tứ phân C không hỗ trợ nên ta phải lập trình (hàm in_bit() và hàm dec_he14()).*/*

bu2=bu1_2(it,&gt);

printf(" Bu2 = %s \n ",bu2); printf("gt= %d ⇔ (dao)it=%d\n ",gt,it);

sh= bin_hex(bu2,&gt); printf("= %s \n ", sh);

hex_oct(sh,&gt); dec_he14(gt,&it); printf("it (so nhap)= %d\n ",it); getch();

}

//----------***định nghĩa các hàm***----------

int *input(void) *// Tên hàm ***input()** trả về hàm main() địa chỉ biến số int nhập từ bàn phím*

{ typedef unsigned char byte;

union REGS v, r; byte a, ok,kx=0; long n=0; int k=0;

/ union REGS trong <dos.h> có định nghĩa struct x với các thanh ghi 16 bit : ax, bx, cx, dx, si, di, cflag, flags(thanh ghi cờ),struct h với các thanh ghi 8 bit: ah, al, bh, bl, ch, cl, dh, dl. Để gọi ngắt, ta đặt chức năng của ngắt vào thanh ghi ah,các tham số vào các thanh ghi khác,nếu có(cần tra cứu bảng sử dụng ngắt ở môn học lập trình hệ thống), rồi gọi hàm xử lý ngắt: **int int86(int shngat, union REGS*v, union REGS*r);** trong đó shngat là số hiệu ngắt phải là số hệ cơ số 16(CPU Intel hoặc tương thích có 256 ngắt được đánh số từ **0x0 ⇔ 0xFF**) tuy vậy chủ yếu sử dụng ngắt 0x10 và 0x21, chúng có rất nhiều chức năng, riêng ngắt 0x21 có thêm 1 hàm gọi riêng: **int intdos(union REGS*v, union REGS*r);** union REGS*v dùng với các thanh ghi chứa chức năng và các tham số đầu vào, union REGS*r dùng với các thanh ghi chứa kết quả trả về sau khi gọi ngắt. vd: ta dùng chức năng 1 và gọi ngắt 0x21, chương trình dừng để chờ ta gõ 1 ký tự từ bàn phím, ngắt trả về mã ASCII của ký tự mà ta gõ trong al: union REGS v,r; v.h.ah=1; intdos(&v,&r); gõ 5 sẽ trả về r.h.al=0x35.*/*

ok=0;// *nhập số dương*

lu: do{ v.h.ah=1; intdos(&v,&r); a= r.h.al;

if(a==13)goto het;// *nếu ấn Enter(mã ASCII là 13) thì kết thúc*

if((a=='-')&&(n==0)){ ok=1; goto lu;}// *nhập số âm,*

if(((a=='x')||(a=='X'))&&(n==0)) {kx=2;goto lu; }

if((a=='-')&&(n!=0)) {v.h.ah=2; v.h.dl=8; intdos(&v,&r); v.h.dl=32; intdos(&v,&r);
v.h.dl=8; intdos(&v,&r); goto lu; }

// Khi đã nhập số liệu thì không cho '-','x','X' hiển thị nữa.

if(((a=='x')||(a=='X'))&&(n!=0)) {v.h.ah=2; v.h.dl=8; intdos(&v,&r); v.h.dl=32; intdos(&v,&r);
v.h.dl=8; intdos(&v,&r); goto lu; }

if(a==8){ if(n){ n/=16; k=n;} else if(kx)kx--;else ok=0;

// đoạn chương trình xóa ký tự thập lục phân,"0x" hoặc dấu âm khi gõ nhầm, mã ASCII của Backspace là 8.

v.h.ah=10; v.x.cx=1; v.h.al=0x20; int86(0x10,&v,&r);

v.h.ah=2; v.h.dl=32; int86(0x21,&v,&r);v.h.dl=8; intdos(&v,&r); goto lu; }

// đoạn chương trình không cho ký tự khác ký tự thập lục phân hiển thị

if(!((a>='0'&&a<='9')||(a>='A'&&a<='F')||(a>='a'&&a<='f'))){ v.h.ah=2; v.h.dl=8; intdos(&v,&r);
v.h.dl=32; intdos(&v,&r); v.h.dl=8; intdos(&v,&r);goto lu; }// *Đặt dấu trống và giữ lại vị trí con trỏ*

n=n<<4|(a<='9'?a& 0x0f: a<='F'?a-'A'+10:a-'a'+10); k=n;

if(n==0x8000 && ok) goto lu;

if(n!=k){ *//giá trị nhập nằm ngoài -0x8000 ⇔0x7fFF, thì xóa về 0, xuống dòng nhập lại*

n=k=0; if(ok) ok=0;

v.h.ah=2; v.h.dl=10; intdos(&v,&r);

v.h.dl=13; intdos(&v,&r);printf("=> nhap lai: "); }

het:}while(a!=13);

if(ok){ n=~n+1; k=n;}// *nếu nhập số âm, thì phải bù 2*

return (&k);

} //……..…….....

void in_bit(int n) *// in n ra bit nhưng không đổi n ra số nhị phân*

{ int size=sizeof(int)*8; // *Hàm sizeof() lấy kích thước số byte(1 byte 8 bit) của kiểu*

int mask= 1<< (size-1), k;

for(k=1; k<=size; ++k) {putchar((n&mask&&1)|060); n<<=1; }}

*/*Để xác định từng bit của 1 số int(từ trái sang phải) ta chọn mặt nạ (mask)có bit MSB (bit ở tận cùng bên trái có trọng số lớn nhất) bằng 1, muốn thế ta dịch bit 1 sang trái (size-1) vị trí: mask=1<<(sise-1), với thanh ghi 16 bit mask= 1000000000000000 Thực hiện phép & giữa mask với số int(thực ra chỉ & bit 1 của mask với bit MSB của số int) đổi ra ký tự bit để in ra. Để xác định bit kế tiếp ta dịch số int sang trái 1 vị trí để bit kế tiếp trở thành bit MSB */*

//----------

char *bin_hex(char *sb, int*gt)

/ Tham số vào là địa chỉ chuỗi số nhị phân (sb), địa chỉ biến gt truyền cho con trỏ *gt lưu giữ số int được đổi ra từ chuỗi số thập lục phân (sh).Tên hàm *bin_hex() trả về địa chỉ chuỗi số hex(sh)*/*

{ int i,j,w=0,t,ok=0,h; char* sh,*s16,*nf; unsigned int k;sh=s16= (char*) calloc(8,1); nf= (char*) calloc(17,1);

// Viết theo quy cách có dấu '-' đặt trước ***:***

i=0;if(*gt<0){k=-*gt;while(k){h=k%16;*(s16+i++)=h<=9?h|0x30:ok==1?h+'A'-10:h+'a'-10;k/=16;}
***(s16+i++)='X';*(s16+i++)='0';*(s16+i++)='-';*(s16+i)=NULL;s16=dao(s16); printf("hex = %s => ",s16);**

***gt=0;i=3;//** *để bỏ qua "-0X"*

while(*(s16+i)) *gt=*gt *16|(*(s16+i)<='9'?*(s16+i++)&0x0f:*(s16+i)<='F'?*(s16+i++)-'A'+10:*(s16+i++)-'a'+10); nf=bu1_2(*gt,&h); *//gọi hàm bu1_2() để đổi số dương *gt ra số đối h (bù 2 một số thì được số đối)*

printf("Bu2= %s= %d= ",nf,h);

}

//Viết theo quy cách số bù 2 (sb nhận chuỗi bu_2) :

/ để đổi 1 chuỗi số nhị phân (sb) ra chuỗi số thập lục phân(sh) ta nhóm từng 4 bit một(i duyệt từ 0 => 15(16 bit), nếu (i+1)%4==0 thì được 1 nhóm 4 bit của chuỗi sb(lần lượt từ trái sang phải) rồi đổi mỗi nhóm ra trị thập lục phân t(0=>15) dùng lệnh thao tác bit(t=0; t=t<<1|(*(sb+j)&1); j duyệt từ i-3 =>i) đổi tiếp ra ký tự thập lục phân ('0','1',...,'9','A',...,'F') hoặc ('0','1',...,'9','a',...,'f') để đưa vào chuỗi sh */*

ok=1; *// Đổi các giá trị thập lục phân 10=>15, với ok=1 đổi ra chữ cái in hoa, nếu không đổi ra chữ cái in thường.*

i=w=0; *(sh+w++)='0'; *(sh+w++)='X'; for(; i<dai(sb); i++) if((i+1)%4==0){ t=0; for(j=i-3; j<=i; j++) t=t<<1|(*(sb+j)&1);*(sh+w++)=t<=9?t|0x30: ok==1?t+'A'-10:t+'a'-10;} *(sh+w)='\0';

/ để đổi 1 dãy số hệ đếm bất kỳ về trị thập phân (int d=0 ;) dùng phương pháp Horner*

d=(...(((a+0)*cs+*(a+1))*cs+*(a+2))*cs+...+*(a+k-1))*cs+*(a+k) với cs: cơ số của hệ đếm, *(a+i) là phần tử thứ i*

hoặc viết theo thao tác bit, dùng cho hệ nhị phân, bát phân hoặc thập lục phân. vd: int a; mảng số hex, đổi ra trị thập phân tương ứng: int d=0; while(i<n)) d=(d<<4) |*(a+i++); chú ý: nếu int *a; là dảy số biểu diễn số âm, thì phải bù 2 d, và đặt dấu âm vào. Đổi 1 chuỗi số hệ đếm bất kỳ về trị thập phân(tương tự như trên), nhưng chú ý đổi ký tự ra giá trị khi tính toán*/*

***gt=0;i=2;//** *để bỏ qua "0X"*

while(*(sh+i))*gt=*gt<<4|(*(sh+i)<='9'?*(sh+i++)&017:*(sh+i)<='F'?*(sh+i++)-'A'+10:*(sh+i++)-'a'+10);
return(sh);

} //----------

void hex_oct(char*sh, int *gt)

/ Hàm đổi từ chuỗi số thập lục phân (sh) ra chuỗi nhị phân(bin), đổi chuỗi bin ra chuỗi hệ cơ số 8(oct). Tham số vào là địa chỉ chuỗi số thập lục phân (sh), địa chỉ biến gt truyền cho con trỏ *gt lưu giữ số int được đổi ra từ chuỗi số hệ cơ số 8 (oct).Tên hàm hex_oct() không trả về giá trị.*/*

{ int i,j,t,w,mask,ok; char *bin, *oct,*s8,*s16;unsigned int k;

bin=(char*) malloc(17*1); oct=(char*)calloc(7,1); s8=s16=(char*) malloc(8*1);

*/*Đổi từ chuỗi số hệ cơ số 16(sh) ra chuỗi số hệ cơ số 8(oct) và ngược lại, hiệu quả nhất là đổi qua chuỗi số trung gian nhị phân(bin). Để đổi chuỗi sh ra chuỗi bin, ta lấy từng ký tự của sh, đổi ra chữ số hệ cơ số 16 có giá trị (t) từ(0=>15 tương ứng với 4 bit thấp nhất của t), mỗi chữ số t ta đổi thành 4 ký tự bit để đưa vào chuỗi bin, muốn thế ta chọn mặt nạ (mask=1<<3), tạo 4 vòng lặp : thực hiện phép & giữa mask với t(thực ra chỉ & bit 1 của mask với bit thứ 3 của t),lấy kết quả && tiếp với 1 để có trị logic 0 hoặc1 , đổi ra ký tự bit đưa vào chuỗi bin(hoặc đơn giản: t& mask?"1":"0"), dịch phải mask 1 bit(mask>>=1) lặp lại.(4&bit thứ 2 của t, 2& bit thứ 1 của t, 1& bit thứ 0 của t).In chuỗi bin trong thân hàm */*

t=j=0;i=2;// *i=2,để bỏ qua"0x"*

while(*(sh+i)){ if(*(sh+i)<='9') t=*(sh+i++)&017; else if(*(sh+i)<='F') t=*(sh+i++)-'A'+10; else

t=*(sh+i++)-'a'+10; mask=1<<3; for(w=0;w<=3;w++){*(bin+j++)=(t&mask?1:0)|'0';

mask>>=1;} } *(bin+j)='\0'; printf("hex=> %s => ",bin);

/ Để đổi 1 chuỗi số nhị phân (bin) ra chuỗi số hệ cơ số 8(oct) ta nhóm từng 3 bit một để tạo 1 ký tự bát phân, tuy vậy 16 không phải là bội số của 3. Vì thế ta đưa bit dấu(bit MSB) của chuỗi bin vào làm bit dấu của chuỗi oct, còn lại 15 bit(i duyệt từ 1 =>15, nếu i%3==0 thì được 1 nhóm 3 bit)rồi đổi mỗi nhóm ra trị bát phân t(0=>7) dùng lệnh thao tác bit:(t=0; t=t<<1|(*(bin+j)&1); j duyệt từ i-2 =>i) đổi tiếp ra ký tự bát phân('0','1',...,'7') để đưa vào chuỗi oct. */*

w=i=0; *(oct+w++)='0';

***(oct+w++)=*(bin+i++);** *//Đưa bit dấu của chuỗi số bin vào làm bit dấu chuỗi số oct(char*oct viết theo quy cách số bù 2)*

for(; i<dai(bin); i++) if((i%3==0){ t=0; for(j=i-2; j<=i; j++) t=t<<1|(*(bin+j)&1); *(oct+w++)= t|060; } *(oct+w)='\0'; printf("Oct= %s ",oct);

/ Để đổi 1 chuỗi số hệ cơ số 8 ra chuỗi số nhị phân, ta đưa bit dấu của chuỗi oct vào làm bit dấu của chuỗi bin, còn lại 5 ký tự , ta lấy từng ký tự đổi ra chữ số hệ cơ số 8 có giá trị t: (từ 0=>7 tương ứng với 3 bit thấp nhất của t), mỗi chữ số t ta đổi thành 3 ký tự bit để đưa vào chuỗi bin, muốn thế ta chọn mặt nạ (mask=1<<2), tạo 3 vòng lặp : thực hiện phép & giữa mask với t(thực ra chỉ & bit 1 của mask với bit thứ 2 của t) đổi ra ký tự bit đưa vào chuỗi bin(t& mask?"1":'0"), dịch phải mask 1 bit(mask>>=1) lặp lại.(2&bit thứ 1 của t, 1& bit thứ 0 của t).In chuỗi bin trong thân hàm */*

i=0;j=1; *// j=1 để bỏ qua '0'.*

***(bin+i++)=*(oct+j++);** *//Đưa bit dấu của chuỗi số oct vào làm bit dấu chuỗi số bin*

for(;j<dai(oct); j++){ t=*(oct+j)&7; mask=1<<2;

for(w=0;w<=2; w++){*(bin+i++)=(t&mask&&1)|060; mask>>=1;}}

***(bin+i)='\0'; printf("=> %s ",bin);**

***gt=0;i=1; while(*(oct+i))*gt=*gt<<3| (*(oct+i++)&7);** */* dùng lệnh thao tác bit để đổi chuỗi oct ra int*gt. chú ý: tên biến chuỗi (oct), tên biến mảng là con trỏ, trong khi tên biến kiểu vô hướng không phải là con trỏ */.*

i=0;if(*gt<0){k=-*gt;while(k){t=k%8;*(s8+i++)=t|060;k/=8;}*(s8+i++)='0';*(s8+i++)='-';

***(s8+i)='\0'; s8=dao(s8); printf("= %s = ",s8);**

***gt=0;i=2; while(*(s8+i))*gt=*gt*8| (*(s8+i++)-060);*gt=-*gt; printf(" %d\n ",*gt);**

} *// char *s8 viết theo quy cách có dấu '-' đặt trước*

} **//----------**

char* bu1_2(int it, int*gt)

/ Tham số vào là giá trị int it, địa chỉ biến số int truyền cho con trỏ *gt lưu giữ số int được đổi ra từ chuỗi số nhị phân hình thành từ số int it .Tên hàm *bu1_2() trả về địa chỉ chuỗi nhị phân.*/.*

{ int i=0, slen,bu2len, t=0, cary; unsigned int k; char*s, *bu2, *bu1; s= (char*) calloc(17,1);

if(it<0){k=-it; k=~k+1;} else k=it; */* chú ý:số âm bé nhất của kiểu biến có dấu và số bù 2 của nó, đều có chuỗi nhị phân bằng nhau vd: số bù 2 của -32768 là +32768 cùng là 1000000000000000, tuy vậy cần phân biệt với kiểu biến không dấu unsigned int là +32768 */*

for(;i<16;i++){ *(s+i)=k%2|0x30; k/=2;} *(s+i)='\0'; s=dao(s); slen=dai(s); bu2len=slen;

bu2=bu1=(char*)malloc(bu2len*sizeof(char));

/ Quá trình tính và hiển thị là ngược nhau vd: đổi 4 ra chuỗi nhị phân 8 bit, thu được chuỗi "00100000" nếu đổi ra số thập phân thì được 32, chứ không phải 4, vì thế phải đảo chuỗi lại"00000100" bằng lệnh gọi hàm dao(s);*/*

*/*Để bù 1 ký tự bit, ta chỉ cần đảo bit LSB(bit tận cùng bên phải, có trọng số nhỏ nhất)của ký tự, bằng cách ^(XOR) LSB với 1. vd:'1'= 00110001, bit LSB là 1 => 1^1=0 , ta được 00110000 là ký tự '0' và ngược lại. '0'=00110000, bit LSB là 0 =>0^1=1 ta được 00110001 là ký tự '1'*/*

i=0;while(*(s+i)){*(bu1+i)=*(s+i)^1; i++;} *(bu1+i)='\0';

printf("Bu1= %s ",bu1);

//Để thực hiện bù 2 chuỗi số bu1, ta cọng chuỗi bu1 với 1, thực ra là cọng bit LSB của chuỗi bu1 với 1

cary=0; *(bu2+bu2len)='\0'; t=(*(bu1+(slen-1))&1)+ 1; cary=t/2; *(bu2+(bu2len-1))= t%2|0x30; slen--; bu2len--;

*/*Sau khi cọng các vị trí tiếp theo bit LSB có thể có bit nhớ cary=1, ta tiếp tục cọng cary với chuỗi số bu1, tạo thành chuỗi bu2 vd: s="01000000"(+64)=>bu1="10111111" =>bu2=10111111+1="11000000"(-64) */*

while(slen){t=(*(bu1+(slen-1))&1)+ cary; cary=t/2; *(bu2+(bu2len-1))= t%2|060; slen--; bu2len--; }

***gt=i=0; while(*(bu2+i))*gt=*gt*2+(*(bu2+i++) -'0'); return(bu2);**

} **//----------**

void *dec_he14(int gt, int*it)

/ Hàm đổi từ số int gt ra chuỗi số hệ cơ số 14. Tham số vào là int gt. Địa chỉ biến &it truyền cho con trỏ *it lưu giữ số đối của int gt.(gt được đổi ra từ chuỗi số hệ 14).*/*

{ int i=0,h,t,ok=1; unsigned int k; char*s14,*s2,*he14;

s14=he14= (char*) calloc(7,1); s2=(char*)malloc(17*1);

//Viết theo quy cách số bù 2 :

```
if(gt<0){k=-gt; k=~k+1;} else k=gt;
while(k){h=k%14; *(s14+i++)=h<=9?h +'0':ok==1? h+'A'-10:h+'a'-10; k/=14;} *(s14+i)='\0';
s14=dao(s14); // Quá trình tính và hiển thị là ngược nhau nên phải đảo chuỗi
 printf("\ngt(he14) = %s ",s14); gt=i=0;
while(*(s14+i))gt=gt*14+(*(s14+i)<='9'?*(s14+i++)-'0':*(s14+i)<='D'?*(s14+i++)-'A'+10:*(s14+i++)-a'+10);
/* dùng phương pháp Horner (không dùng lệnh  thao tác bit được)để đổi chuỗi s14 ra số int t ( gt), rồi bù 2 t để có lại giá trị nhập ở đầu chương trình là số int it.*/
//Viết theo quy cách có dấu '-' đặt trước :
ok=i=0;
if(gt<0){k=-gt;while(k){h=k%14;*(he14+i++)=h<=9?h+060:ok==1?h+'A'-10:h+'a'-10;k/=14;}
*(he14+i++)='-'; *(he14+i)='\0';he14=dao(he14);
 gt=0; i=1;//bỏ qua  '-'
while(*(he14+i))gt=gt*14+(*(he14+i)<='9'?*(he14+i++)-48:*(he14+i)<='D'?*(he14+i++)-A'+10:*(he14+i++)-
a'+10); gt=~gt+1; printf(" OR= %s= %d(he10)=> ",he14,gt);
        }
s2=bu1_2(gt,&*it); //hoặc it
printf("Bu2 = %s ", s2);
}  //----------------------------------------------------------------------------------------------------------------------
char *dao(char *s)
// Tham số vào là địa chỉ chuỗi (s) Tên hàm *dao() trả về địa chỉ chuỗi là chuỗi đảo từ chuỗi s.
 int i=0; char ch;
for(;i<dai(s)/2; i++){ ch=*(s+i); *(s+i)=*(s+(dai(s)-i-1)); *(s+(dai(s)-i-1))=ch; }
// Phải có -1 để không đảo NULL ở cuối chuỗi lên đầu chuỗi.
return s;
} //----------------------------------------------------------------------------------------------------------------------
int dai(char *s)
// Tham số vào là địa chỉ chuỗi (s). Tên hàm dai() trả về độ dài thật của chuỗi s.
{ char*st=s;
 while(*++st){} /* cho con trỏ *st chạy từ đầu chuỗi đến cuối chuỗi( gặp NULL), rồi lấy địa chỉ của st trừ cho địa
                  chỉ đầu chuỗi, ta được độ dài thật của chuỗi */
 return(st-s);}     /* hoặc: int i=0;for(; *(s+ ++i); ); return i; cho i chạy từ đầu chuỗi đến cuối chuỗi, thì trả i về.
Lệnh ; hoặc {} là lệnh rỗng.*/
 //----------------------------------------------------------------------------------------------------------------------
```

Khoa Công nghệ Thông tin

Bộ môn **HỆ THỐNG NHÚNG**

IT FACULTY

Đề thi môn học: **KỸ THUẬT SỐ & ƯD**

Thời gian: **75 phút**

Khi làm bài thi giữa kỳ/ cuối kỳ môn kỹ thuật SỐ& ỨNG DỤNG sinh viên được sử dụng tài liệu của riêng mình(nghiêm cấm mượn tài liệu của người khác). Đề thi giữa kỳ tương tự câu 2) của đề thi cuối kỳ. Dưới đây là các đề thi tham khảo, trong học kỳ sinh viên cùng giáo viên phân tích, thảo luận, phản biện để giải các bài tập dưới dạng đề thi nầy.

****

Khoa Công nghệ Thông tin Môn thi: **KỸ THUẬT SỐ& ỨNG DỤNG**

Đề 1: Thời gian 75 **phút** (*Sinh viên được dùng tài liệu của riêng mình, ghi tên vào đề thi và nộp lại*)

2) Tìm biểu thức tối giản của nguyên hàm và đảo hàm(không dùng bảng Karnaugh)

f(a,b,c,d,e)=a/bcde+/ab/cd+ac/d+bc/d+/acd+/a/bd+/a/bc/d+ab.

(Khi dùng định lý logic, triển khai đầy đủ các bước, không làm tắt, không đưa về biểu thức chuẩn tắc tuyển)

1)Thiết kế mạch đếm thập phân, không đồng bộ, đếm thuận, dùng mạch flipflop T(CP tác động bằng sườn dương), mã đầu vào tùy chọn. **Không vẽ mạch**.

+ phân tích yêu cầu thiết kế,xác định đồ hình trạng thái ban đầu, lập bảng Karnaugh chung.

+ vẽ giản đồ thời gian, tìm phương trình định thời

+Tìm phương trình đầu ra, phương trình trạng thái

+ kiểm tra quá trình chuyển đổi trạng thái của mạch, kiểm tra tự khởi động của mạch.

+ tìm phương trình đầu vào kích.

3) Hệ CPU 8 bit (16 bit địa chỉ A15⇔A0, 8 bit dữ liệu D7⇔ D0, 2 bit điều khiển đọc/ ghi tích cực thấp /RD, /WR). Hãy dùng cổng logic và IC 74138, để thiết kế mạch giải mã địa chỉ, đặt 2KB ROM (1x 2716) vào địa chỉ nền 0x1800, kế tiếp về phía địa chỉ cao đặt 4KB RAM(1/2x 6264). Hãy giải thích lệnh đọc ngăn nhớ có địa chỉ 0x2222 (giả thiết nội dung của ngăn nhớ là 0130) bằng các bit 0 hoặc 1 trên bus địa chỉ, bus dữ liệu, và bus điều khiển của hệ.(3đ)

KHOA CÔNG NGHỆ THÔNG TIN **MÔN THI:** ***KỸ THUẬT SỐ& ỨNG DỤNG***

Đề 2: Thời gian: **75 phút** (*Sinh viên được dùng tài liệu của riêng mình. Ghi tên vào đề thi và nộp lại*)

*****

1)Thiết kế mạch giải mã LED 7 đoạn(cathode chung 1: LED sáng, 0: LED tắt) hiển thị số hệ thập tứ phân (0,1,…,9,A,b,C,d), mã đầu vào là mã GRAY. Thiết kế mạch dạng NAND_NAND . viết chương trình mô phỏng tín hiệu đầu vào/ đầu ra, vẽ mạch điện .

gợi ý: +lập bảng công tác của mạch

*+lập 7 bảng **Karnaugh** cho 7 đầu ra (mỗi đầu ra tương ứng với 1 segment)*

+đánh vòng các ô toàn 1, tìm biểu thức tối giản của hàm,

+phủ định 2 lần nữa, triển khai đến cấp số hạng,ta có biểu thức tối giản toàn NAND

2)Tìm biểu thức tối giản(BTTG) của hàm và đảo hàm(dùng bảng Karnaugh):

f(a,b,c,d,e)=∑(0,1,3,4,5,7,8,9,10,12,13,14,15,24,26,30) và ∑(11,18,25,29,31)=0**.**

gợi ý: +đánh vòng các ô toàn 1 và x để tìm BTTG của nguyên hàm(hàm).

+đánh vòng các ô toàn 0 và x để tìm BTTG của đảo hàm.

3) Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 ⇔A0), 8 bit dữ liệu(D0⇔D7) và 2 bit điều khiển đọc/ ghi tích cực thấp (/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổng logic) để đặt 6KBROM (1 x 2732+ 1 x 2716)vào địa chỉ cơ sở 0xA000. Hãy giải thích lệnh đọc 0200 tại ngăn nhớ 0xA01A bằng các bit 0(bit 1) trên các bus địa chỉ, dữ liệu và điều khiển .

KHOA CÔNG NGHỆ THÔNG TIN **MÔN THI:** ***KỸ THUẬT SỐ& ỨNG DỤNG***

Đề 3: Thời gian: **75 phút** (*Sinh viên được dùng tài liệu của riêng mình, nhớ ghi tên vào đề thi và nộp lại*)

*****

1) Thiết kế mạch chuyển mã BCD_2421=> mã JOHNSON dạng AND_OR(dùng cho hệ đếm thập tam phân: 0,1,..,9,A,b,C). Viết chương trình mô phỏng tín hiệu đầu vào/ đầu ra, vẽ mạch điện.

gợi ý: *+lập bảng công tác của mạch*

*+lập 5 bảng **Karnaugh** cho 5 đầu ra.*

+đánh vòng các ô toàn 0, tìm biểu thức tối giản của đảo hàm,

+phủ định 1 lần nữa, triển khai đến cấp biến,ta có biểu thức tối giản dạng AND_OR

2) Cho đảo hàm: /f(a,b,c,d,e)= ∑(0,1,3,4,5,6,7,10,14,15,25,26,28,30) **, hãy tìm biểu thức chuẩn tắc tuyển (BTCTT) và biểu thức chuẩn tắc hội(BTCTH) của hàm f(a,b,c,d,e).**

gợi ý: *+để tìm BTCTT ta dùng định luật" **triển khai"***

*+ để tìm BTCTH ta dùng định luật "**De_Morgan"***

3) Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điều khiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổng logic) để đặt 2KBROM(1x 2716) vào vùng địa chỉ cao nhất của hệ, kế tiếp về phía địa chỉ thấp đặt 2KBRAM(1x 6116). Hãy giải thích lệnh đọc ngăn nhớ 0xF85A bằng các bit 0/1 trên các bus địa chỉ, dữ liệu và điều khiển .

KHOA CÔNG NGHỆ THÔNG TIN MÔN THI: *KỸ THUẬT SỐ& ỨNG DỤNG*

Đề 4: Thời gian: **75 phút** (*Sinh viên được dùng tài liệu của riêng mình, nhớ ghi tên vào đề thi và nộp lại*)

*****

1) Thiết kế mạch đếm thập phân,không đồng bộ, đếm thuận, dùng mạch flipflop D(CP tác động bằng sườn âm), mã đầu vào tùy chọn, vẽ mạch điện.

gợi ý : *+phân tích yêu cầu thiết kế, xác định đồ hình trạng thái ban đầu, lập bảng Karnaugh chung.*

+vẽ giản đồ thời gian, tìm phương trình định thời

+Tìm phương trình đ ầu ra, phương trình trạng thái

+ kiểm tra tự khởi động và vận hành của mạch

+ tìm phương trình đầu vào kích, vẽ mạch

2) Tìm BTTG của hàm/đảo hàm(dùng bảng Karnaugh)

f(a,b,c,d,e,f)=b/cd/e+ b/ce/f+ b/cd/+ /abc/f+ abc/d/e/f+ b/d/ef+ abce/f+ bd/ef+ abcd/e.

gợi ý:+Tìm BTCTT của hàm, đánh vòng các ô toàn 1 để tìm BTTG của hàm

+Tìm BTCTT của đảo hàm(phủ định 1 lần BTCTT của hàm rồi dùng luật triển khai), đánh vòng các ô toàn 0 để tìm BTTG của đảo hàm.

3) Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điều khiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổng logic) để đặt 8KBRAM(1 x 6264) vào vùng địa chỉ thấp nhất, và 4KBROM(1x 2732) vào vùng địa chỉ cao nhất. Hãy giải thích lệnh ghi -0x5D vào ngăn nhớ 0x21A, bằng các bit 0/1trên các bus địa chỉ, dữ liệu và điều khiển .

KHOA CÔNG NGHỆ THÔNG TIN MÔN THI: *KỸ THUẬT SỐ& ỨNG DỤNG*

Đề 5: Thời gian: **75 phút** (*Sinh viên được dùng tài liệu của riêng mình, nhớ ghi tên vào đề thi và nộp lại*)

*****

1)Thiết kế mạch giải mã LED 7 đoạn(anode chung 0: LED sáng, 1: LED tắt) hiển thị số thập nhị phân(0,1,…,9,A,b) , mã đầu vào là BÙ_1. Thiết kế mạch dạng NOR_AND((*viết chương trình mô phỏng* tín hiệu đầu vào/ đầu ra,*vẽ mạch điện)*

gợi ý: *+lập bảng công tác của mạch*

*+lập 7 bảng **Karnaugh** cho 7 đầu ra (mỗi đầu ra tương ứng với 1 segment)*

+đánh vòng các ô toàn 0, tìm biểu thức tối giản của đảo hàm

+phủ định 1 lần nữa,nhưng không triển khai gì, vẽ mạch

2) Tìm BTTG của hàm/đảo hàm (dùng định lý và bảng Karnaugh)

f(a,b,c,d)= /b/c+/a/c/d+abc+b/cd+a/cd+ab.

***(**Khi dùng định lý logic, triển khai đầy đủ các bước, không làm tắt, không đưa về biểu thức chuẩn tắc tuyển)*

gợi ý: Khi dùng bảng K+Tìm BTCTT của hàm, đánh vòng các ô toàn 1 để tìm BTTG của hàm

+Tìm BTCTT của đảo hàm (phủ định 1 lần BTCTT của hàm rồi dùng luật triển khai), đánh vòng các ô toàn 0 để tìm BTTG của đảo hàm.

3) Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điềukhiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổng logic) để đặt 4KBROM(1 x 2732) vào địa chỉ cơ sở 0x4000 và kế tiếp về phía địa chỉ cao đặt 4KB RAM(1/2 x 6264). Hãy giải thích lệnh đọc -0129 tại ngăn nhớ 0x401B, bằng các bit 0/1 trên các bus địa chỉ, dữ liệu và điều khiển .

KHOA CÔNG NGHỆ THÔNG TIN M ÔN THI: *KỸ THUẬT SỐ& ỨNG DỤNG*

Đề 6: Thời gian: **75 phút** (*Sinh viên được dùng tài liệu của riêng mình, nhớ ghi tên vào đề thi và nộp lại*)

*****

2) Dùng định lý logic tìm BTTG của hàm/đảo hàm (dùng định lý và bảng Karnaugh)

f(a,b,c,d,e)= /a/bd/e+/a/bcd+/ac/d+/abd+abc+ab/cd+a/bd+a/bc/d .

(*Khi dùng định lý logic, triển khai đầy đủ các bước, không làm tắt, không đưa về biểu thức chuẩn tắc tuyển)*

gợi ý: Khi dùng bảng K+Tìm BTCTT của hàm, đánh vòng các ô toàn 1 để tìm BTTG của hàm

+Tìm BTCTT của đảo hàm (phủ định 1 lần BTCTT của hàm rồi dùng luật triển khai), đánh vòng các ô toàn 0 để tìm BTTG của đảo hàm.

1) Thiết kế mạch giải mã LED 7 đoạn(anode chung 0: LED sáng, 1: LED tắt) hiển thị số thập phân, mã đầu vào là mã BCD 84-2-1. Thiết kế mạch dạng NOR_NOR(*viết chươngtrình mô phỏng* tín hiệu đầu vào/ đầu ra,*vẽ mạch điện)*

gợi ý: *+lập bảng công tác của mạch*

*+lập 7 bảng **Karnaugh** cho 7 đầu ra (mỗi đầu ra tương ứng với 1 segment)*

+đánh vòng các ô toàn 0, tìm biểu thức tối giản của đảo hàm

+phủ định 1 lần nữa , triển khai đến cấp biến

+phủ định 2 lần nữa, chỉ triển khai đến cấp số hạng

3) Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điềukhiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổng logic) để đặt 12KBROM(1 x 2732+1x 2764) vào địa chỉ cơ sở 0x8000 và kế tiếp về phía địa chỉ cao đặt 4KB RAM(1/2 x 6264). Hãy giải thích lệnh đọc -0129 tại ngăn nhớ 0x801B, bằng các bit 0/1 trên các bus địa chỉ, dữ liệu và điều khiển .

Khoa Công nghệ Thông tin Môn thi**: KỸ THUẬT SỐ& ỨNG DỤNG**

Đề 7: Thời gian 75**phút** (*Sinh viên được dùng tài liệu của riêng mình, ghi tên vào đề thi và nộp lại*)

2) Tìm biểu thức tối giản của đảo hàm và nguyên hàm (không dùng bảng Karnaugh)

/f(a,b,c,d,e)=c/de+ab/cd+/acd+bcd+a/bcd+a/b/c+a/c/d+/ab. (1đ)

(*Khi dùng định lý logic, triển khai đầy đủ các bước, không làm tắt, không đưa về biểu thức chuẩn tắc tuyển*)

1) Thiết kế mạch đếm thập phân, không đồng bộ, đếm nghịch, dùng mạch flipflop JK(CP tác động bằng sườn âm), mã đầu vào tùy chọn. Không vẽ mạch.(5đ)

+ phân tích yêu cầu thiết kế,xác định đồ hình trạng thái ban đầu, lập bảng Karnaugh chung

+ vẽ giản đồ thời gian, tìm phương trình định thời

+Tìm phương trình đầu ra, phương trình trạng thái

+ kiểm tra quá trình chuyển đổi trạng thái của mạch, kiểm tra tự khởi động của mạch.

+ tìm phương trình đầu vào kích,

3) Hệ CPU 8 bit (16 bit địa chỉ A15⇔A0, 8 bit dữ liệu D7⇔ D0, 2 bit điều khiển đọc/ ghi tích cực thấp /RD, /WR). Hãy dùng cổng logic và IC 74138, để thiết kế mạch giải mã địa chỉ, đặt 2KB ROM (1x 2716) vào vùng địa chỉ thấp nhất, kế tiếp về phía địa chỉ cao đặt 1KBRAM (1/2x 6116). Hãy giải thích lệnh đọc ngăn nhớ có địa chỉ 0xCE (giả thiết nội dung của ngăn nhớ là 0x68) bằng các bit 0 hoặc 1 trên bus địa chỉ, bus dữ liệu và bus điều khiển của hệ(3đ).

(1đ trình bày)

Khoa Công nghệ Thông tin Môn thi**: KỸ THUẬT SỐ& ỨNG DỤNG**

Đề 8: Thời gian 75 **phút** (*Sinh viên được dùng tài liệu của riêng mình, ghi tên vào đề thi và nộp lại*)

2) Tìm biểu thức tối giản của đảo hàm và nguyên hàm (không dùng bảng Karnaugh)

/f(a,b,c,d,e)=ac/de+/a/bd+/ab/cd+bcd+acd+abc+abd+a/c/d+a/b/cd.

(*Khi dùng định lý logic, triển khai đầy đủ các bước, không làm tắt, không đưa về biểu thức chuẩn tắc tuyển)*

1) Thiết kế mạch đếm thập phân, không đồng bộ, đếm nghịch, dùng mạch flipflop JK(CP tác động bằng sườn dương), mã đầu vào tùy chọn. Không vẽ mạch.

+ phân tích yêu cầu thiết kế,xác định đồ hình trạng thái ban đầu, lập bảng Karnaugh chung

+ vẽ giản đồ thời gian, tìm phương trình định thời

+Tìm phương trình đầu ra, phương trình trạng thái

+ kiểm tra quá trình chuyển đổi trạng thái của mạch, kiểm tra tự khởi động của mạch.

+ tìm phương trình đầu vào kích.

3) Hệ CPU 8 bit (16 bit địa chỉ A15⇔A0, 8 bit dữ liệu D7⇔ D0, 2 bit điều khiển đọc/ ghi tích cực thấp /RD, /WR). Hãy dùng cổng logic và IC 74138, để thiết kế mạch giải mã địa chỉ, đặt 2KB RAM (1x 6116) vào địa chỉ nền 0xB800, kế tiếp về phía địa chỉ thấp đặt 1KB ROM (1/2x 2716). Hãy giải thích lệnh đọc ngăn nhớ có địa chỉ 0xB444 (giả thiết nội dung của ngăn nhớ là -0x59) bằng các bit 0 hoặc 1 trên bus địa chỉ, bus dữ liệu và bus điều khiển của hệ.

KHOA CÔNG NGHỆ THÔNG TIN MÔN THI: ***KỸ THUẬT SỐ& ỨNG DỤNG***

Đề 9: Thời gian: **75 phút** (*Sinh viên được dùng tài liệu của riêng mình, nhớ ghi tên vào đề thi và nộp lại*)

*****

1)Thiết kế mạch giải mã LED 7 đoạn(cathode chung 1: LED sáng, 0: LED tắt) hiển thị số thập nhất phân(0,1,..,9,A), mã đầu vào là mã DƯ_3. Thiết kế mạch dạng NAND_NAND(viết chương trình mô phỏng tín hiệu đầu vào/ đầu ra, vẽ mạch điện).

gợi ý: *+lập bảng công tác của mạch*

*+ lập 7 bảng **Karnaugh** cho 7 đầu ra (mỗi đầu ra tương ứng với 1 segment)*

+đánh vòng các ô toàn 1, tìm biểu thức tối giản của hàm,

+phủ đnh 2 lần nữa, triển khai đến cấp số hạng,ta có biểu thức tối giản toàn NAND

2) Tìm BTTG của hàm/đảo hàm(dùng định lý, không dùng bảng Karnaugh)

f(a,b,c,d,e,f)=b/cd/e + b/ce/f+ b/cd/e + /abc/f + abc/d/e/f + b/d/ef+ abce/f + bd/ef+ abcd/e.

(Khi dùng định lý logic, triển khai đầy đủ các bước, không làm tắt, không đưa về biểu thức chuẩn tắc tuyển)

3) Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điều khiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổng logic) để đặt 6KBRAM(1/2x 6264+ 1x 6116) vào địa chỉ cơ sở 0xA000. Hãy giải thích lệnh ghi 100 vào ngăn nhớ 0xA0AA, bằng các bit 0/1 trên các bus địa chỉ, dữ liệu và điều khiển .

KHOA CÔNG NGHỆ THÔNG TIN MÔN THI: *KỸ THUẬT SỐ& ỨNG DỤNG*

Đề 10: Thời gian: **75 phút** (*Sinh viên được dùng tài liệu của riêng mình, nhớ ghi tên vào đề thi và nộp lại*)

*****

1) Thiết kế mạch chuyển mã BÙ_1=> BÙ_2 dạng NAND_NAND(dùng cho hệ đếm thập tứ phân: 0,1,..,9,A,b,C,d). Viết chương trình mô phỏng tín hiệu đầu vào/ đầu ra,vẽ mạch điện)

gợi ý: *+lập bảng công tác của mạch*

*+lập 4 bảng **Karnaugh** cho 4 đầu ra.*

+đánh vòng các ô toàn 0, tìm biểu thức tối giản của đảo hàm,

+phủ định 1 lần nữa, triển khai đến cấp biến,ta có biểu thức tối giản dạng AND_OR

+phủ định 2 lần nữa,nhưng chỉ triển khai đến cấp số hạng.

2) Tìm BTTG của hàm/đảo hàm(dùng định lý logic và bảng Karnaugh)

f(a,b,c,d,e)=b/d/e+ /a/ce+ ab/c/de+ /abd/e+ ab/e+ bc/de+ /acde+ /a/bce.

(Khi dùng định lý logic, triển khai đầy đủ các bước, không làm tắt, không đưa về biểu thức chuẩn tắc tuyển)

gợi ý:+Tìm BTCTT của hàm, đánh vòng các ô toàn 1 để tìm BTTG của hàm

+Tìm BTCTT của đảo hàm(phủ định 1 lần BTCTT của hàm rồi dùng luật triển khai), đánh vòng các ô toàn 0 để tìm BTTG của đảo hàm.

3)Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điều khiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổng logic) để đặt 6KBRAM(3 x 6116) vào địa chỉ cơ sở 0xA000. Hãy giải thích lệnh ghi -10 vào ngăn nhớ 0xA01E, bằng các bit 0/1 trên các bus địa chỉ, dữ liệu và điều khiển .

KHOA CÔNG NGHỆ THÔNG TIN MÔN THI: *KỸ THUẬT SỐ& ỨNG DỤNG*

Đề 11: Thời gian: **75 phút**(*Sinh viên được dùng tài liệu của riêng mình, nhớ ghi tên vào đề thi và nộp lại*)

*****

1) Thiết kế mạch giải mã LED 7 đoạn(anode chung 0: LED sáng, 1: LED tắt) hiển thị số thập ngũ phân(0,1,…,9,A,b,C,d,E), mã đầu vào là mã BÙ-2. Thiết kế mạch dạng NOR_AND(viết chương trình mô phỏng tín hiệu đầu vào/ đầu ra vẽ mạch điện)

gợi ý: *+lập bảng công tác của mạch*

+lập 7 bảng karnaugh cho 7 đầu ra (mỗi đầu ra tương ứng với 1 segment)

+đánh vòng các ô toàn 0, tìm biểu thức tối giản của đảo hàm

+phủ định 1 lần nữa, nhưng không triển khai gì

2)Tối giản hàm logic(dùng định lý Quine Mc_Cluskey, không dùng bảng Karnaugh):

f(a,b,c,d,e)=∑(8,9,11,12,13,14,15,18,21,22,23,24,25,27,28,29,30,31) **.**

3)Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điềukhiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổng logic) để đặt 3KBROM(1 x 2716+1/2 x2716) vào vùng địa chỉ cao nhất. giải thích lệnh đọc ngăn nhớ 0x21A, bằng các bit 0/1 trên các bus địa chỉ, dữ liệu và điều khiển .

KHOA CÔNG NGHỆ THÔNG TIN MÔN THI: *KỸ THUẬT SỐ& ỨNG DỤNG*

Đề 12: Thời gian: **75 phút** (*Sinh viên được dùng tài liệu của riêng mình, nhớ ghi tên vào đề thi và nộp lại*)

*****

1) Thiết kế mạch chuyển mã JOHNSON=>BCD_7421 dạng AND_OR(dùng cho hệ đếm thập tam phân: 0,1,..,9,A,b,C).(Viết chương trình mô phỏng tín hiệu đầu vào/ đầu ra, vẽ mạch điện)

gợi ý: *+lập bảng công tác của mạch*

*+lập 4 bảng **Karnaugh** cho 4 đầu ra.*

+đánh vòng các ô toàn 0, tìm biểu thức tối giản của đảo hàm,

+phủ định 1 lần nữa, triển khai đến cấp biến,ta có biểu thức tối giản dạng AND_OR

2) Tối giản hàm f(a,b,c,d,e)=∑(3,4,5,7,8,9,11,12,15,18,20,21,22) **.**

và điều kiện ràng buộc ∑(0,1,13,23) = 0 **(phương pháp tối giản tùy chọn)**

3)Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điều khiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổng logic) để đặt 4KBRAM(1/2 x 6264) vào địa chỉ cơ sở 0x4000, kế tiếp về phía địa chỉ thấp đặt 4KBROM(1x 2732). Hãy giải thích lệnh ghi -114 vào ngăn nhớ 0x4005, bằng các bit 0/1 trên các bus địa chỉ, dữ liệu và điều khiển .

KHOA CÔNG NGHỆ THÔNG TIN MÔN THI: *KỸ THUẬT SỐ& ỨNG DỤNG*

Đề 13: Thời gian:75 **phút** (*Sinh viên được dùng tài liệu của riêng mình, nhớ ghi tên vào đề thi và nộp lại*)

*****

2) tìm BTTG của hàm/đảo hàm (dùng định lý và bảng Karnaugh)

f(a,b,c,d,e)= /cde+ /a/b/c/d+ b/c/d+ a/c/d+ /ac/d+ /abc+ acd+ a/bc+ abc/d.

(Khi dùng định lý logic, triển khai đầy đủ các bước, không làm tắt, không đưa về biểu thức chuẩn tắc tuyển)

gợi ý: Khi dung bảng K+Tìm BTCTT của hàm, đánh vòng các ô toàn 1 để tìm BTTG của hàm

+Tìm BTCTT của đảo hàm (phủ định 1 lần BTCTT của hàm rồi dùng luật triển khai), đánh vòng các ô toàn 0 để tìm BTTG của đảo hàm.

1) Thiết kế mạch đếm thập phân,không đồng bộ, đếm thuận, dùng mạch flipflop T(CP tác động bằng sườn âm), mã đầu vào tùy chọn, vẽ mạch điện.

gợi ý : *+phân tích yêu cầu thiết kế, xác định đồ hình trạng thái ban đầu, lập bảng K tổng hợp*

+vẽ giản đồ thời gian, tìm phương trình đ ịnh thời

+Tìm phương trình đầu ra, phương trình trạng thái kểm tra tự khởi động của mạch

+ tìm phương trình đ ầu vào kích, vẽ mạch

3) Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điều khiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổng logic) để đặt 12KBROM(1x 2732+1x 2764) vào địa chỉ cơ sở 0xC000. Hãy giải thích lệnh đọc ngăn nhớ 0xC01A bằng các bit 0/1 trên các bus địa chỉ, dữ liệu và điều khiển .

KHOA CÔNG NGHỆ THÔNG TIN **MÔN THI:** ***KỸ THUẬT SỐ& ỨNG DỤNG***

Đề 14: Thời gian: **75 phút** (*Sinh viên được dùng tài liệu của riêng mình, nhớ ghi tên vào đề thi*)

*****

1) Thiết kế mạch giải mã LED 7 đoạn(anode chung 0: LED sáng, 1: LED tắt) hiển thị số Hex(0,1,..,9,A,bC,d,E,F), mã đầu vào là GRAY+3. Thiết kế mạch dạng NOR_AND(vẽ mạch điện, viết chương trình mô phỏng tín hiệu đầu vào/ đầu ra)

gợi ý: *+lập bảng công tác của mạch*

*+lập 7 bảng **Karnaugh** cho 7 đầu ra (mỗi đầu ra tương ứng với 1 segment)*

+đánh vòng các ô toàn 0, tìm biểu thức tối giản của đảo hàm

+phủ định 1 lần nữa, nhưng không triển khai gì

2) Tìm BTTG của hàm/đảo hàm(dùng định lý logic và bảng Karnaugh)

f(a,b,c,d,e)=/a/c/d+ /ab+ b/c/d+ a/b/c/d+/ce+ a/bd+ a/cd+ abcd.

(*Khi dùng định lý logic, triển khai đầy đủ các bước, không làm tắt, không đưa về biểu thức chuẩn tắc tuyển)*

gợi ý:+Tìm BTCTT của hàm, đánh vòng các ô toàn 1 để tìm BTTG của hàm

+Tìm BTCTT của đảo hàm(phủ định 1 lần BTCTT của hàm rồi dùng luật triển khai), đánh vòng các ô toàn 0 để tìm BTTG của đảo hàm.

3)Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điềukhiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổnglogic) để đặt 3KBRAM(1/2x 6116+ 1x 6116) vào vùng địa chỉ thấp nhất. Hãy giải thích lệnh ghi 0xB5, vào ngăn nhớ 0x2A bằng các bit 0/1 trên các bus địa chỉ, dữ liệu và điều khiển .

KHOA CÔNG NGHỆ THÔNG TIN **MÔN THI:** ***KỸ THUẬT SỐ& ỨNG DỤNG***

Đề 15: Thời gian: **75 phút** (*Sinh viên được dùng tài liệu của riêng mình, nhớ ghi tên vào đề thi*)

*****

1) Thiết kế mạch chuyển mã BCD_5421=>BCD_5121 dạng AND_OR (dùng cho hệ đếm thập lục phân: 0,1,..,9,A,b,C,d,E,F). Viết chương trình mô phỏng tín hiệu đầu vào/ đầu ra,vẽ mạch điện

gợi ý: *+lập bảng công tác của mạch*

*+lập 4 bảng **Karnaugh** cho 4 đầu ra.*

+đánh vòng các ô toàn 0, tìm biểu thức tối giản của đảo hàm,

+phủ định 1 lần nữa, triển khai đến cấp biến,ta có biểu thức tối giản dạng AND_OR

+phủ định 2 lần nữa, nhưng chỉ triển khai đến cấp số hạng

2) Tìm BTTG của hàm/đảo hàm (dùng định lý)

f(a,b,c,d,e)= /a/b/d/e+ b/ce+ bcde+ /abc/de+ abc/d+ ad/e+ a/b/e+ ab/c.

(Khi dùng định lý logic, triển khai đầy đủ các bước, không làm tắt, không đưa về biểu thức chuẩn tắc tuyển)

3) Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điều khiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổng logic) để đặt 4KBRAM(1/2 x 6264) vào địa chỉ cơ sở 0xC000, và tiếp theo về phía địa chỉ cao là 4KBROM(1x 2732) Hãy giải thích lệnh đọc ngăn nhớ 0xC111, bằng các bit 0/1 trên các bus địa chỉ, dữ liệu và điều khiển .

KHOA CÔNG NGHỆ THÔNG TIN **MÔN THI: *KỸ THUẬT SỐ& ỨNG DỤNG***

Đề 16: Thời gian: **75 phút** (*Sinh viên được dùng tài liệu của riêng mình, nhớ ghi tên vào đề thi*)

****

1)Thiết kế mạch giải mã LED 7 đoạn(cathode chung 1: LED sáng, 0: LED tắt) hiển thị số thập phân, mã đầu vào là mã BCD_84-2-1. Thiết kế mạch dạng NAND_NAND(viết chương trình mô phỏng tín hiệu đầu vào/ đầu ra, vẽ mạch điện).

gợi ý: *+lập bảng công tác của mạch*

*+lập 7 bảng **Karnaugh** cho 7 đầu ra.*

+đánh vòng các ô toàn 1, tìm biểu thức tối giản của hàm,

+phủ định 2 lần nữa, triển khai đến cấp số hạng,ta có biểu thức tối giản toàn NAND

2) Tối giản hàm f(a,b,c,d,e)=$\sum(3,4,5,7,8,9,11,12,15,18,20,21,22)$ **.**

và điều kiện ràng buộc $\sum(0,1,13,23) = 0$ **(phương pháp tối giản, tùy chọn)**

3) Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điều khiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổng logic) để đặt 8KBROM(1x 2764) vào vùng địa chỉ cao nhất, kế tiếp về phía địa chỉ thấp đặt 4KBRAM(1/2x 6264). Hãy giải thích lệnh đọc -125 tại ngăn nhớ 0xF01A, bằng các bit 0/1 trên các bus địa chỉ, dữ liệu và điều khiển .

KHOA CÔNG NGHỆ THÔNG TIN **MÔN THI: KỸ THUẬT SỐ& ỨNG DỤNG**

Đề 17: Thời gian: **75 phút** (*Sinh viên không được dùng tài liệu, nhớ ghi tên vào đề thi*)

*****

1) Thiết kế mạch chuyển mã BCD_5421=>BCD_5121 dạng AND_OR(dùng cho hệ đếm thập ngũ phân: 0,1,..,9,A,b,C,d,E). (Viết chương trình mô phỏng tín hiệu đầu vào/ đầu ra, vẽ mạch điện)

+lập bảng công tác của mạch

*+lập 4 bảng **Karnaugh** cho 4 đầu ra.*

+đánh vòng các ô toàn 0, tìm biểu thức tối giản của đảo hàm,

+phủ định 1 lần nữa, triển khai đến cấp biến,ta có biểu thức tối giản dạng AND_OR

2)Tối giản hàm logic(dung định lý Quine Mc_Cluskey, không dùng bảng Karnaugh):

f(a,b,c,d,e)=$\sum(4,5,7,16,17,19,20,21,23,24,25,27,28,29,31)$ **.**

3) Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điều khiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổng logic) để đặt 4KBRAM(1/2 x 6264) vào vùng địa chỉ thấp nhất. Hãy giải thích lệnh ghi -109 vào ngăn nhớ 0x20A, bằng các bit 0/1 trên các bus địa chỉ, dữ liệu và điều khiển .

KHOA CÔNG NGHỆ THÔNG TIN **MÔN THI: *KỸ THUẬT SỐ& ỨNG DỤNG***

Đề 18: Thời gian: **75 phút** (*Sinh viên được dùng tài liệu của riêng mình, nhớ ghi tên vào đề thi*)

*****

1) Thiết kế mạch giải mã LED 7 đoạn(anode chung 0: LED sáng, 1: LED tắt) hiển thị số hệ đếm thập ngũ phân: 0,1,..,9,A,b,C,d,E). mã đầu vào là mã BCD_5421. Thiết kế mạch dạng NOR_AND(viết chương trình mô phỏng tín hiệu đầu vào/ đầu ra ,vẽ mạch điện)

gợi ý: *+lập bảng công tác của mạch*

*+lập 7 bảng **Karnaugh** cho 7 đầu ra (mỗi đầu ra tương ứng với 1 segment)*

+đánh vòng các ô toàn 0, tìm biểu thức tối giản của đảo hàm

+phủ định 1 lần nữa,nhưng không triển khai gì, vẽ mạch

2) Tối giản hàm/đảo hàm logic(dùng định lý, và dùng bảng Karnaugh):

f(a,b,c,d,e)=$\sum$(1,2,3,5,6,8,9,10,11,12,13,15,24,25,26,27,28,29)

+Tìm BTCTT của đảo hàm(phủ định 1 lần BTCTT của hàm rồi dùng luật triển khai), đánh vòng các ô toàn 0 để tìm BTTG của đảo hàm.

3)Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điều khiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổng logic) để đặt 4KBRAM(1/2 x 6264) vào địa chỉ cơ sở 0x1000, kế tiếp về phía địa chỉ cao, đặt 4KB ROM(1x 2732). Hãy giải thích lệnh ghi 150 vào ngăn nhớ 0x10AA, bằng các bit 0/1 trên các bus địa chỉ, dữ liệu và điều khiển .

.----------

KHOA CÔNG NGHỆ THÔNG TIN MÔN THI: *KỸ THUẬT SỐ& ỨNG DỤNG*

Đề 19: Thời gian: **75 phút** (*Sinh viên được dùng tài liệu của riêng mình, nhớ ghi tên vào đề thi*)

*****

2) Tìm BTTG của hàm/đảo hàm (dùng định lý và bảng Karnaugh)

f(a,b,c,d,e)= /a/b/d/e+ b/ce+ bcde+ /abc/de+ abc/d+ ad/e+ a/b/e+ ab/c.

(*Khi dùng định lý logic, triển khai đầy đủ các bước, không làm tắt, không đưa về biểu thức chuẩn tắc tuyển)*

gợi ý: Khi dung bảng K+Tìm BTCTT của hàm, đánh vòng các ô toàn 1 để tìm BTTG của hàm

+Tìm BTCTT của đảo hàm(phủ định 1 lần BTCTT của hàm rồi dùng luật triển khai), đánh vòng các ô toàn 0 để tìm BTTG của đảo hàm.

1) Thiết kế mạch đếm thập phân,không đồng bộ, đếm thuận, dùng mạch flipflop JK(CP tác động bằng sườn âm), vẽ mạch điện.Mã vào tùy chọn

gợi ý : *+phân tích yêu cầu thiết kế, xác định đồ hình trạng thái ban đầu, lập bảng K tổng hợp*

+vẽ giản đồ thời gian,tìm phương trình đ ịnh thời

+Tìm phương trình đ ầu ra, phương trình trạng thái

+ kiểm tra tự khởi động của mạch

+ tìm phương trình đầu vào kích, vẽ mạch

3)Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điều khiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổng logic) để đặt 2KBRAM(1x 6116) vào địa chỉ cơ sở 0xE000. Hãy giải thích lệnh ghi 110 vào ngăn nhớ 0xE01A bằng các bit 0/1 trên các bus địa chỉ, dữ liệu và điều khiển .

KHOA CÔNG NGHỆ THÔNG TIN MÔN THI: *KỸ THUẬT SỐ& ỨNG DỤNG*

Đề 20: Thời gian: **75 phút** (*Sinh viên khôngđược dùng tài liệu, nhớ ghi tên vào đề thi và nộp lại*)

*****

1) Thiết kế mạch giải mã LED 7 đoạn(cathode chung 1: LED sáng, 0: LED tắt) hiển thị số thập phân, mã đầu vào là BCD_7421. Thiết kế mạch dạng OR_AND(vẽ mạch điện, viết chương trình mô phỏng tín hiệu đầu vào/ đầu ra)

gợi ý: *+lập bảng công tác của mạch*

*+lập 7 bảng **Karnaugh** cho 7 đầu ra (mỗi đầu ra tương ứng với 1 segment)*

+đánh vòng các ô toàn 1, tìm biểu thức tối giản của hàm, vẽ mạch.

2) cho đảo hàm /f(a,b,c,d)=/abc+b/cd+a/c/d, hãy tìm biểu thức chuẩn tắc tuyển (BTCTT) và

biểu thức chuẩn tắc hội(BTCTH) của hàm f(a,b,c,d).

gợi ý: *+để tìm BTCTT ta dung định luật"* ***triển khai"***

*+ để tìm BTCTH ta dung định luật "****De_Morgan"***

3) Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điều khiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổng logic) để đặt 1KBROM(1/2x 2716) vào địa chỉ cơ sở 0xC000. Hãy giải thích lệnh đọc ngăn nhớ 0xC03D, bằng các bit 0/1 trên các bus địa chỉ, dữ liệu và điều khiển .

KHOA CÔNG NGHỆ THÔNG TIN MÔN THI: *KỸ THUẬT SỐ& ỨNG DỤNG*

Đề 21: Thời gian: **75 phút** (*Sinh viên được dùng tài liệu của riêng mình, nhớ ghi tên vào đề thi*)

*****

1) Thiết kế mạch chuyễn mã JOHNSON=>GRAY+3, dạng NOR_NOR (dùng cho hệ đếm thập nhất phân: 0,1,..,9,A). (viết chương trình mô phỏng tín hiệu đầu vào/ đầu ra,vẽ mạch điện)

gợi ý: *+lập bảng công tác của mạch*

+lập 4 bảng ***Karnaugh*** *cho 4 đầu ra.*

+đánh vòng các ô toàn 0, tìm biểu thức tối giản của đảo hàm,

+phủ định 1 lần nữa, triển khai đến cấp biến,ta có biểu thức tối giản dạng AND_OR

+phủ định 2 lần nữa, chỉ triển khai đến cấp số hạng.

2) Tìm BTTG của hàm/đảo hàm(dùng định lý)

f(a,b,c,d,e,f)=/a/b/e/f+/a/b/ce/f+/a/b/de/f+/a/bcde/f+a/be/f+a/bc/e/f+a/b/d/e/f+a/b/cd/e/f+b/c/de/f+ab/ce/f+

/ab/cde/f .

(*Khi dùng định lý logic, triển khai đầy đủ các bước, không làm tắt, không đưa về biểu thức chuẩn tắc tuyển)*

3) Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điều khiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổnglogic) để đặt 4KBRAM(1/2 x 6264) vào địa chỉ cơ sở 0x1000, và kế tiếp về phía địa chỉ caođặt 8KBROM(1x 2764). Hãy giải thích lệnh đọc ngăn nhớ 0x201A bằng các bit 0/1 trêncác bus địa chỉ, dữ liệu và điều khiển .

KHOA CÔNG NGHỆ THÔNG TIN MÔN THI: *KỸ THUẬT SỐ& ỨNG DỤNG*

Đề 22: Thời gian: **75 phút** ((*Sinh viên được dùng tài liệu của riêng mình, nhớ ghi tên vào đề thi*)

*****

1)Thiết kế mạch giải mã LED 7 đoạn(cathode chung 1: LED sáng, 0: LED tắt) hiển thị số thập phân, mã đầu vào là mã BCD_5121. Thiết kế mạch dạng NAND_NAND(viết chương trình mô phỏng tín hiệu đầu vào/ đầu ra, vẽ mạch điện).

gợi ý: *+lập bảng công tác của mạch*

+lập 7 bảng ***Karnaugh*** *cho 7 đầu ra(mỗi đầu ra tương ứng với 1 segment).*

+đánh vòng các ô toàn 1, tìm biểu thức tối giản của hàm,+phủ định 2 lần nữa, triển khai đến cấp số hạng,ta có biểu thức tối giản toàn NAND

2) Tối giản hàm f(a,b,c,d,e)=$\sum$(0,1,4,5,9,11,13,15,16,17,18,20,22,24,25,26,27,28,29,30,31) **. (phương pháp tối giản tùy chọn)**

3) Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điều khiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổng logic) để đặt 32KBRAM(1 x 62256) vào địa chỉ cơ sở 0x8000. Hãy giải thích lệnh ghi -65 vào ngăn nhớ 0xA01A bằng các bit 0/1 trên các bus địa chỉ, dữ liệu và điều khiển .

KHOA CÔNG NGHỆ THÔNG TIN MÔN THI: KỸ THUẬT *SỐ& ỨNG DỤNG*

Đề 23: Thời gian: **75 phút** (*Sinh viên được dùng tài liệu của riêng mình, nhớ ghi tên vào đề thi*)

*****

1) Thiết kế mạch chuyển mã BCD_5421=>BCD_7421 dạng AND_OR (dùng cho hệ đếm thập tam phân: 0,1,..,9,A,b,C)(Viết chương trình mô phỏng tín hiệu đầu vào/ đầu ra, vẽ mạch điện)

gợi ý: *+lập bảng công tác của mạch*

*+lập 4 bảng **Karnaugh** cho 4 đầu ra.*

+đánh vòng các ô toàn 0, tìm biểu thức tối giản của đảo hàm,

+phủ định 1 lần nữa, triển khai đến cấp biến,ta có biểu thức tối giản dạng AND_OR

2)Tối giản hàm logic(dùng định lý Quine Mc_Cluskey, không dùng bảng Karnaugh):

f(a,b,c,d,e)=∑(8,9,11,12,13,14,15,18,21,22,23,24,25,27,28,29,30,31) **.**

3)Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điều khiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổng logic) để đặt 24KBROM(1 x 27128+1x 2764) vào địa chỉ cơ sở 0x8000. Hãy giải thích lệnh đọc 117, tại ngăn nhớ 0x800B, bằng các bit 0/1 trên các bus địa chỉ, dữ liệu và điều khiển .

KHOA CÔNG NGHỆ THÔNG TIN MÔN THI: *KỸ THUẬT SỐ& ỨNG DỤNG*

Đề 24: Thời gian: **75 phút** (*Sinh viên được dùng tài liệu của riêng mình, nhớ ghi tên vào đề thi*)

*****

1) Thiết kế mạch giải mã LED 7 đoạn(anode chung 0: LED sáng, 1: LED tắt) hiển thị số thập nhị phân(0,1,..,9,A,b), mã đầu vào là mã GRAY. Thiết kế mạch dạng NOR_AND(viết chương trình mô phỏng tín hiệu đầu vào/ đầu ra,vẽ mạch điện)

gợi ý: *+lập bảng công tác của mạch*

*+lập 7 bảng **Karnaugh** cho 7 đầu ra (mỗi đầu ra tương ứng với 1 segment)*

+đánh vòng các ô toàn 0, tìm biểu thức tối giản của đảo hàm

+phủ định 1 lần nữa, nhưng không triển khai gì, vẽ mạch-

2) tìm BTTG của hàm/đảo hàm (dùng định lý và bảng Karnaugh)

f(a,b,c,d,e)= /a/b/c+/ab/c/d+ab/c+abd+b/cd+/abcd+a/bc+/a/bce+a/bc.

(*Khi dùng định lý logic, triển khai đầy đủ các bước, không làm tắt, không đưa về biểu thức chuẩn tắc tuyển)*

gợi ý: Khi dùng bảng K+Tìm BTCTT của hàm, đánh vòng các ô toàn 1 để tìm BTTG của hàm

+Tìm BTCTT của đảo hàm (phủ định 1 lần BTCTT của hàm rồi dùng luật triển khai), đánh vòng các ô toàn 0 để tìm BTTG của đảo hàm.

3) Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điều khiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổng logic) để đặt 10KBRAM(1x 6264+ 1x6116) vào địa chỉ cơ sở 0xC000. Hãy giải thích lệnh ghi -100 vào ngăn nhớ 0xC666 bằng các bit 0/1 trên các bus địa chỉ, dữ liệu và điều khiển .

KHOA CÔNG NGHỆ THÔNG TIN MÔN THI: *KỸ THUẬT SỐ& ỨNG DỤNG*

Đề 25: Thời gian: **75 phút** (*Sinh viên được dùng tài liệu của riêng mình, nhớ ghi tên vào đề thi i*)

*****

2)Tối giản hàm f(a,b,c,d,e,f)=∑(0,1,2,4,5,6,8,9,10,12,14,18,22,30,33,34,36,37,38,40,41,42,44,45,46,50,54) (***phương pháp tối giản tùy chọn***).

1) Thiết kế mạch đếm thập phân,không đồng bộ, đếm thuận, dùng mạch flipflop D(CP tác động bằng sườn âm), vẽ mạch điện.

gợi ý : *+phân tích yêu cầu thiết kế, xác định đồ hình trạng thái ban đầu, lập bảng K tổng hợp*

+vẽ giản đồ thời gian, chọn xung cp(clock pulse)

+Tìm phương trình đ ầu ra, phương trình trạng thái

+ kiểm tra tự khởi động của mạch

+ tìm phương trình đ ầu vào kích, vẽ mạch

3) Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điều khiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổng logic) để đặt 4KBRAM(1/2 x 6264) vào địa chỉ cơ sở 0x3000, kế tiếp về phía địa chỉ cao đặt 8KBROM(1x 2732). Hãy giải thích lệnh đọc ngăn nhớ 0x5555, bằng các bit 0/1 trên các bus địa chỉ, dữ liệu và điều khiển .

KHOA CÔNG NGHỆ THÔNG TIN MÔN THI: *KỸ THUẬT SỐ& ỨNG DỤNG*

Đề 26: Thời gian: **75 phút** (*Sinh viên được dùng tài liệu của riêng mình, nhớ ghi tên vào đề thi*)

*****

1) Thiết kế mạch giải mã LED 7 đoạn(anode chung 0: LED sáng, 1: LED tắt) hiển thị số thập nhất phân (0,1,..,9,A), mã đầu vào là BCD_2421.Thiết kế mạch dạng AND_OR(vẽ mạch điện, viết chương trình mô phỏng tín hiệu đầu vào/ đầu ra)

gợi ý: *+lập bảng công tác của mạch*

*+lập 7 bảng **Karnaugh** cho 7 đầu ra (mỗi đầu ra tương ứng với 1 segment)*

+đánh vòng các ô toàn 0, tìm biểu thức tối giản của đảo hàm

+phủ định 1 lần nữa, triển khai đến cấp biến, ta có biểu thức tối giản dạng AND_OR.

2) Tối giản hàm f(a,b,c,d,e)=$\sum(0,1,5,7,8,9,11,12,15,18,20,21,23)$ **.**

và điều kiện ràng buộc $\sum(3,4,13,22) = 0$ **(phương pháp tối giản tùy chọn)**

3) Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điều khiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổng logic) để đặt 16KBRAM(2 x 6264) vào địa chỉ cơ sở 0x8000. Hãy giải thích lệnh ghi -0x7A vào ngăn nhớ 0xD01A bằng các bit 0/1 trên các bus địa chỉ, dữ liệu và điều khiển .

KHOA CÔNG NGHỆ THÔNG TIN MÔN THI: *KỸ THUẬT SỐ& ỨNG DỤNG*

Đề 27: Thời gian: **75 phút**(*Sinh viên được dùng tài liệu của riêng mình, nhớ ghi tên vào đề thi*)

*****

1) Thiết kế mạch chuyễn mã DƯ_3=>BCD 5121 (dùng cho hệ đếm thập ngũ phân: 0,1,..,9,A,b,C,D,E), dạng NOR_NOR(viết chương trình mô phỏng tín hiệu đầu vào/ đầu ra,vẽ mạch điện)

Gợi ý: *+lập bảng công tác của mạch*

*+lập 4 bảng **Karnaugh** cho 4 đầu ra.*

+đánh vòng các ô toàn 0, tìm biểu thức tối giản của đảo hàm,

+phủ định 1 lần nữa, triển khai đến cấp biến,ta có biểu thức tối giản dạng AND_OR

+phủ định 2 lần nữa,chỉ triển khai đến cấp số hạng. vẽ mạch

2)Tìm BTTG của hàm/ đảo hàm(phương pháp tối giản tùy chọn):

f(a,b,c,d,e,f)=$\sum(2,6,10,14,16,17,18,20,21,22,34,38,42,46,48,49,50,52,53,54,58,62)$ **.**

3) Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điều khiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổng logic) để đặt 32KBROM(2 x 27128) vào vùng địa chỉ thấp nhất. Hãy giải thích lệnh đọc -0x4A tại ngăn nhớ 0x701A bằng các bit 0/1 trên các bus địa chỉ, dữ liệu và điều khiển .

KHOA CÔNG NGHỆ THÔNG TIN **MÔN THI: *KỸ THUẬT SỐ& ỨNG DỤNG***

Đề 28: Thời gian: **75 phút** (*Sinh viên được dùng tài liệu của riêng mình, nhớ ghi tên vào đề thi*)

*****

1)**Thiết kế mạch giải mã LED 7 đoạn(cathode chung 1: LED sáng, 0: LED tắt) hiển thị số thập phân, mã đầu vào là mã JOHNSON. Thiết kế mạch dạng NAND_NAND(viết chương trình mô phỏng tín hiệu đầu vào/ đầu ra, vẽ mạch điện).**

+lập bảng công tác của mạch

*+lập 7 bảng **Karnaugh** cho 7 đầu ra(mỗi đầu ra tương ứng với 1 segment).*

+đánh vòng các ô toàn 1, tìm biểu thức tối giản của hàm,

+phủ định 2 lần nữa, triển khai đến cấp số hạng,ta có biểu thức tối giản toàn NAND

2)Tìm BTTG của hàm/ đảo hàm(phương pháp tối giản tùy chọn):

f(a,b,c,d,e)=∑(0,1,3,4,5,7,16,17,19,20,21,23,26,30,31) **.**

3) Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điều khiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổng logic) để đặt 4KBRAM(1/2 x 6264) vào vùng địa chỉ thấp nhất, và 8KBROM(1x 2764) vào vùng địa chỉ cao nhất. Hãy giải thích lệnh ghi 0150 vào ngăn nhớ 0xAAA bằng các bit 0/1 trên các bus địa chỉ, dữ liệu và điều khiển .

KHOA CÔNG NGHỆ THÔNG TIN **MÔN THI: *KỸ THUẬT SỐ& ỨNG DỤNG***

Đề 29: Thời gian: **75 phút** (*Sinh viên được dùng tài liệu của riêng mình, nhớ ghi tên vào đề thi*)

*****

1) Thiết kế mạch chuyển mã BCD_5121=>BÙ_1 dạng AND_OR(dùng cho hệ đếm thập nhị phân: 0,1,..,9,A,b). Viết chương trình mô phỏng tín hiệu đầu vào/ đầu ra, vẽ mạch điện.

gợi ý: *+lập bảng công tác của mạch*

*+lập 4 bảng **Karnaugh** cho 4 đầu ra.*

+đánh vòng các ô toàn 0, tìm biểu thức tối giản ủa đảo hàm,

+phủ định 1 lần nữa, triển khai đến cấp biến,ta có biểu thức tối giản dạng AND_OR

2) Tìm BTTG của hàm/đảo hàm(dùng định lý logic và bảng Karnaugh)

f(a,b,c,d,e)= /abcd+/a/bce+/a/b/c+/ab/c/d+ab/c+abd+b/cd+a/bc+bc/d.

***(**Khi dùng định lý logic, triển khai đầy đủ các bước, không làm tắt, không đưa về biểu thức chuẩn tắc tuyển)*

gợi ý:+Tìm BTCTT của hàm, đánh vòng các ô toàn 1 để tìm BTTG của hàm

+Tìm BTCTT của đảo hàm(phủ định 1 lần BTCTT của hàm rồi dùng luật triển khai), đánh vòng các ô toàn 0 để tìm BTTG của đảo hàm.

3) Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điều khiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổng logic) để đặt 8KBRAM(1x 6264) vào vùng địa chỉ thấp nhất, kế tiếp về phía địa chỉ cao đặt 8KBROM(1x 2764). Hãy giải thích lệnh ghi 100 vào ngăn nhớ 0xFFF bằng các bit 0/1 trên các bus địa chỉ, dữ liệu và điều khiển .

KHOA CÔNG NGHỆ THÔNG TIN **MÔN THI: *KỸ THUẬT SỐ& ỨNG DỤNG***

Đề 30: Thời gian: **75 phút** (*Sinh viên được dùng tài liệu của riêng mình, nhớ ghi tên vào đề thi*)

*****

1) Thiết kế mạch giải mã LED 7 đoạn(anode chung 0: LED sáng, 1: LED tắt) hiển thị số

HEX(0,1,..,9,A,b,C,d,E,F),mã đầu vào là mã BÙ_2. Thiết kế mạch dạng NOR_AND(viết chương trình mô phỏng tín hiệu đầu vào/ đầu ra,vẽ mạch điện)

gợi ý: *+lập bảng công tác của mạch*

*+lập 7 bảng **Karnaugh** cho 7 đầu ra (mỗi đầu ra tương ứng với 1 segment)*

+đánh vòng các ô toàn 0, tìm biểu thức tối giản của đảo hàm.

+phủ định 1 lần nữa,nhưng không triển khai, vẽ mạch

2) Tìm BTTG của hàm/đảo hàm(dùng định lý logic)

f(a,b,c,d,e,f)=a/d/e/f+/a/b/c/d+/a/b/cdf+d/ef +b/d/e/f+/a/b/ce+bd/e/f+/d/ef+/ac/d/e/f+/ad/e/f+a/bd/e/f .

(*Khi dùng định lý logic, triển khai đầy đủ các bước, không làm tắt, không đưa về biểu thức chuẩn tắc tuyển)*

3)Trong hệ CPU 8 bit, có 16 bit địa chỉ(A15 A0), 8 bit dữ liệu(D0 D7) và 2 bit điềukhiển đọc/ ghi(/WR và /RD). Thiết kế mạch giải mã địa chỉ(dùng IC 74138 và các cổng logic) để đặt 12KBROM(1x 2764+ 1x 2732) vào địa chỉ cơ sở 0x2000. Hãy giải thích lệnh đọc ngăn nhớ 0x2222 bằng các bit 0/1 trên các bus địa chỉ, dữ liệu và điều khiển .

Khoa Công nghệ Thông tin

Bộ môn **HỆ THỐNG NHÚNG**

IT FACULTY

BÀI TẬP
KỸ THUẬT SỐ & ƯD

1. NHIỆM VỤ CỦA SINH VIÊN :

- *Nắm chắc lý thuyết đại số Boole, các định lý logic, các cổng logic, dạng thức chuẩn tắc tuyển, phương pháp tối giản biểu thức logic bằng định lý logic và bằng bảng KARNAUGH.*
- *Phân tích bài tập được giao, để xây dựng bảng công tác cho mạch logic cần thiết kế.*
- *Lập biểu thức logic cho các đầu ra.*
- *Tiến hành tối giản biểu thức logic (dùng cả 2 phương pháp định lý logic và bảng karnaugh).*
- *Thiết kế mạch chuyển đổi mã với các dạng logic: OR – AND, AND – OR, NOR – AND, NOR – NOR và NAND – NAND).*
- *Vẽ mạch logic.*
- *Bắt buộc phải có chương trình mô phỏng tín hiệu logic của mạch bằng ngôn ngữ lập trình C và kết quả chạy chương trình.*
- *Sinh viên nhóm xxa có số thứ tự n nhận bài tập thứ n.(**số thứ tự trong danh sách thi giữa kỳ**)*
- *Sinh viên nhóm xxb có số thứ tự n nhận bài tập thứ n+55.(**số thứ tự trong danh sách thi giữa kỳ+55**)*
- *Số đề bài phải được ghi rõ ở trang bìa bản báo cáo vd:* ***Đề bài: số 56***
- *Bản báo cáo phải được soạn thảo bằng máy tính (cấm viết tay). Nộp báo cáo vào ngày học cuối cùng của học kỳ.*

2. BÀI TẬP: ◙ **Thiết kế mạch CHUYỂN ĐỔI MÃ**

A) Bài tập dành cho nhóm học tập xxa từ bài số 1 => đến bài số 55:

+BÀI 1: dùng cho hệ đếm thập phân(0,1,…,9)

- Mã đầu vào: BCD 8421 => Mã đầu ra: BCD 7421

+BÀI 2: dùng cho hệ đếm thập nhất phân(0,1,…,9,A)

- Mã đầu vào: BCD 8421 => Mã đầu ra: BCD 2421

+**<u>BÀI 3</u>:** dùng cho hệ đếm thập nhị phân(0,1,…,9,A,B)

- **Mã đầu vào: BCD 8421 => Mã đầu ra: GRAY**

+**<u>BÀI 4</u>:** dùng cho hệ đếm thập tam phân(0,1,…,9,A,B,C)

- **Mã đầu vào: BCD 8421 => Mã đầu ra: Dư 3**

+**<u>BÀI 5</u>:** dùng cho hệ đếm thập tứ phân(0,1,…,9,A,B,C,D)

- **Mã đầu vào: BCD 8421 => Mã đầu ra: BCD 5121**

+**<u>BÀI 6</u>:** dùng cho hệ đếm thập ngũ phân(0,1,…,9,A,B,C,D,E)

- **Mã đầu vào: BCD 8421 => Mã đầu ra: BCD 8-4-2-1**

+**<u>BÀI 7</u>:** dùng cho hệ đếm thập lục phân(0,1,…,9,A,B,C,D,E,F)

- **Mã đầu vào: BCD 8421 => Mã đầu ra: Bù 1**

+**<u>BÀI 8</u>:** dùng cho hệ đếm thập phân(0,1,…,9)

- **Mã đầu vào: BCD 8421 => Mã đầu ra: Bù 2**

+**<u>BÀI 9</u>:** dùng cho hệ đếm thập nhất phân(0,1,…,9,A)

- **Mã đầu vào BCD: 8421 => Mã đầu ra: GRAY+ 3**

+**<u>BÀI 10</u>:** dùng cho hệ đếm thập nhị phân(0,1,…,9,A,B)

- **Mã đầu vào: BCD 8421 => Mã đầu ra: BCD 5421**

+**<u>BÀI 11:</u>** dùng cho hệ đếm thập tam phân(0,1,…,9,A,B,C)

- **Mã đầu vào: BCD 2421 => Mã đầu ra: BCD 7421**

+**<u>BÀI 12</u>:** dùng cho hệ đếm thập tứ phân(0,1,…,9,A,B,C,D)

- **Mã đầu vào: BCD 2421 => Mã đầu ra: BCD 8421**

+**<u>BÀI 13</u>:** dùng cho hệ đếm thập ngũ phân(0,1,…,9,A,B,C,D,E)

- **Mã đầu vào: BCD 2421 => Mã đầu ra: GRAY**

+**<u>BÀI 14</u>:** dùng cho hệ đếm thập lục phân(0,1,…,9,A,B,CD,E,F)

- **Mã đầu vào: BCD 2421 => Mã đầu ra: Dư 3**

+**<u>BÀI 15</u>:** dùng cho hệ đếm thập phân(0,1,…,9)

- **Mã đầu vào: BCD 2421 => Mã đầu ra: BCD 5121**

+**<u>BÀI 16</u>:** dùng cho hệ đếm thập nhất phân(0,1,…,9,A)

- **Mã đầu vào: BCD 2421 => Mã đầu ra: BCD 8-4-2-1**

+**<u>BÀI 17</u>:** dùng cho hệ đếm thập nhị phân(0,1,…,9,A,B)

- **Mã đầu vào: BCD 2421 => Mã đầu ra: Bù 1**

+**<u>BÀI 18</u>:** dùng cho hệ đếm thập tam phân(0,1,…,9,A,B,C)

- **Mã đầu vào: BCD 2421 => Mã đầu ra: Bù 2**

+**<u>BÀI 19</u>:** dùng cho hệ đếm thập tứ phân(0,1,…,9,A,B,C,D)

- **Mã đầu vào BCD: 2421 => Mã đầu ra: GRAY+ 3**

+**<u>BÀI 20</u>:** dùng cho hệ đếm thập ngũ phân(0,1,…,9,A,B,C,D,E)

- **Mã đầu vào: BCD 2421 => Mã đầu ra: BCD 5421**

+**<u>BÀI 21:</u>** dùng cho hệ đếm thập lục phân(0,1,…,9,A,B,C,D,E,F)

- **Mã đầu vào: BCD 5121 => Mã đầu ra: BCD 7421**

+**<u>BÀI 22</u>:** dùng cho hệ đếm thập phân(0,1,…,9)

- Mã đầu vào: BCD 5121 => Mã đầu ra: BCD 2421

+**<u>BÀI 23</u>:** dùng cho hệ đếm thập *nhất phân(0,1,…,9,A)*

- Mã đầu vào: BCD 5121 => Mã đầu ra: GRAY

+**<u>BÀI 24</u>:** dùng cho hệ đếm thập nhị phân(0,1,…,9,A,B)

- Mã đầu vào: BCD 5121 => Mã đầu ra: Dư 3

+**<u>BÀI 25</u>:** dùng cho hệ đếm thập tam phân(0,1,…,9,A,B,C)

- Mã đầu vào: BCD 5121 => Mã đầu ra: BCD 8421

+**<u>BÀI 26</u>:** dùng cho hệ đếm thập tứ phân(0,1,…,9,A,B,C,D)

- Mã đầu vào: BCD 5121 => Mã đầu ra: BCD 8-4-2-1

+**<u>BÀI 27</u>:** dùng cho hệ đếm thập ngũ phân(0,1,…,9,A,B,C,D,E)

- Mã đầu vào: BCD 5121 => Mã đầu ra: Bù 1

+**<u>BÀI 28</u>:** dùng cho hệ đếm thập lục phân(0,1,…,9,A,B,C,D,E,F)

- Mã đầu vào: BCD 5121 => Mã đầu ra: Bù 2

+**<u>BÀI 29</u>:** dùng cho hệ đếm thập phân(0,1,…,9)

- Mã đầu vào BCD: 5121 => Mã đầu ra: GRAY+ 3

+**<u>BÀI 30</u>:** dùng cho hệ đếm thập nhất phân(0,1,…,9,A)

- Mã đầu vào: BCD 5121 => Mã đầu ra: BCD 5421

+**<u>BÀI 31:</u>** dùng cho hệ đếm thập nhị phân(0,1,…,9,A,B)

- Mã đầu vào: BCD 5421 => Mã đầu ra: BCD 7421

+**<u>BÀI 32</u>:** dùng cho hệ đếm thập tam phân(0,1,…,9,A,B,C)

- Mã đầu vào: BCD 5421 => Mã đầu ra: BCD 2421

+**<u>BÀI 33</u>:** dùng cho hệ đếm thập tứ phân(0,1,…,9,A,B,C,D)

- Mã đầu vào: BCD 5421 => Mã đầu ra: GRAY

+**<u>BÀI 34</u>:** dùng cho hệ đếm thập ngũ phân(0,1,…,9,A,B,C,D,E)

- Mã đầu vào: BCD 5421 => Mã đầu ra: Dư 3

+**<u>BÀI 35</u>:** dùng cho hệ đếm thập lục phân(0,1,…,9,A,B,C,D,E,F)

- Mã đầu vào: BCD 5421 => Mã đầu ra: BCD 5121

+**<u>BÀI 36</u>:** dùng cho hệ đếm thập phân(0,1,…,9)

- Mã đầu vào: BCD 5421 => Mã đầu ra: BCD 8-4-2-1

+**<u>BÀI 37</u>:** dùng cho hệ đếm thập nhất phân(0,1,…,9,A)

- Mã đầu vào: BCD 5421 => Mã đầu ra: Bù 1

+**<u>BÀI 38</u>:** dùng cho hệ đếm thập nhị phân(0,1,…,9,A,B)

- Mã đầu vào: BCD 5421 => Mã đầu ra: Bù 2

+**<u>BÀI 39</u>:** dùng cho hệ đếm thập tam phân(0,1,…,9,A,B,C)

- Mã đầu vào BCD: 5421 => Mã đầu ra: GRAY+ 3

+**<u>BÀI 40</u>:** dùng cho hệ đếm thập tứ phân(0,1,…,9,A,B,C,D)

- Mã đầu vào: BCD 5421 => Mã đầu ra: BCD 8421

+**<u>BÀI 41:</u>** dùng cho hệ đếm thập ngũ phân(0,1,…,9,A,B,C,D,E)

- **Mã đầu vào: BCD 7421 => Mã đầu ra: BCD 8421**

+**<u>BÀI 42</u>:** dùng cho hệ đếm thập lục phân(0,1,…,9,A,B,C,D,E,F)

- **Mã đầu vào: BCD 7421 => Mã đầu ra: BCD 2421**

+**<u>BÀI 43</u>:** dùng cho hệ đếm thập phân(0,1,…,9)

- **Mã đầu vào: BCD 7421 => Mã đầu ra: GRAY**

+**<u>BÀI 44</u>:** dùng cho hệ đếm thập nhất phân(0,1,…,9,A)

- **Mã đầu vào: BCD 7421 => Mã đầu ra: Dư 3**

+**<u>BÀI 45</u>:** dùng cho hệ đếm thập nhị phân(0,1,…,9,A,B)

- **Mã đầu vào: BCD 7421 => Mã đầu ra: BCD 5121**

+**<u>BÀI 46</u>:** dùng cho hệ đếm thập tam phân(0,1,…,9,A,B,C)

- **Mã đầu vào: BCD 7421 => Mã đầu ra: BCD 8-4-2-1**

+**<u>BÀI 47</u>:** dùng cho hệ đếm thập tứ phân(0,1,…,9,A,B,C,D)

- **Mã đầu vào: BCD 7421 => Mã đầu ra: Bù 1**

+**<u>BÀI 48</u>:** dùng cho hệ đếm thập ngũ phân(0,1,…,9,A,B,C,D,E)

- **Mã đầu vào: BCD 7421 => Mã đầu ra: Bù 2**

+**<u>BÀI 49</u>:** dùng cho hệ đếm thập lục phân(0,1,…,9,A,B,C,D,E,F)

- **Mã đầu vào BCD: 7421 => Mã đầu ra: GRAY+ 3**

+**<u>BÀI 50</u>:** dùng cho hệ đếm thập phân(0,1,…,9)

- **Mã đầu vào: BCD 7421 => Mã đầu ra: BCD 5421**

+**<u>BÀI 51:</u>** dùng cho hệ đếm thập nhất phân(0,1,…,9,A)

- **Mã đầu vào: BCD 84-2-1 => Mã đầu ra: BCD 7421**

+**<u>BÀI 52</u>:** dùng cho hệ đếm thập nhị phân(0,1,…,9,A,B)

- **Mã đầu vào: BCD 84-2-1 => Mã đầu ra: BCD 2421**

+**<u>BÀI 53</u>:** dùng cho hệ đếm thập tam phân(0,1,…,9,A,B,C)

- **Mã đầu vào: BCD 84-2-1 => Mã đầu ra: GRAY**

+**<u>BÀI 54</u>:** dùng cho hệ đếm thập tứ phân(0,1,…,9,A,B,C,D)

- **Mã đầu vào: BCD 84-2-1 => Mã đầu ra: Dư 3**

+**<u>BÀI 55</u>:** dùng cho hệ đếm thập ngũ phân(0,1,…,9,A,B,C,D,E)

- **Mã đầu vào: BCD 84-2-1 => Mã đầu ra: BCD 5121**

<u>B) Bài tập dành cho nhóm học tập xxb từ bài số 56 => đến bài số 110</u>:

+**<u>BÀI 56</u>:** dùng cho hệ đếm thập lục phân(0,1,…,9,A,B,C,D,E,F)

- **Mã đầu vào: BCD 84-2-1 => Mã đầu ra: BCD 8421**

+**<u>BÀI 57</u>:** dùng cho hệ đếm thập phân(0,1,…,9)

- **Mã đầu vào: BCD 84-2-1 => Mã đầu ra: Bù 1**

+**<u>BÀI 58</u>:** dùng cho hệ đếm thập nhất phân(0,1,…,9,A)

- **Mã đầu vào: BCD 84-2-1 => Mã đầu ra: Bù 2**

+**<u>BÀI 59</u>:** dùng cho hệ đếm thập nhị phân(0,1,…,9,A,B)

- **Mã đầu vào BCD: 84-2-1 => Mã đầu ra: GRAY+ 3**

+**<u>BÀI 60</u>:** dùng cho hệ đếm thập tam phân(0,1,…,9,A,B,C)

- **Mã đầu vào: BCD 84-2-1 => Mã đầu ra: BCD 5421**

+**<u>BÀI 61:</u>** dùng cho hệ đếm thập tứ phân(0,1,…,9,A,B,C,D)

- **Mã đầu vào: GRAY => Mã đầu ra: BCD 7421**

+**<u>BÀI 62</u>:** dùng cho hệ đếm thập ngũ phân(0,1,…,9,A,B,C,D,E)

- **Mã đầu vào: GRAY => Mã đầu ra: BCD 2421**

+**<u>BÀI 63</u>:** dùng cho hệ đếm thập lục phân(0,1,…,9,A,B,C,D,E,F)

- **Mã đầu vào: GRAY => Mã đầu ra: 8421**

+**<u>BÀI 64</u>:** dùng cho hệ đếm thập phân(0,1,…,9)

- **Mã đầu vào: GRAY => Mã đầu ra: Dư 3**

+**<u>BÀI 65</u>:** dùng cho hệ đếm thập nhất phân(0,1,…,9,A)

- **Mã đầu vào: GRAY => Mã đầu ra: BCD 5121**

+**<u>BÀI 66</u>:** dùng cho hệ đếm thập nhị phân(0,1,…,9,A,B)

- **Mã đầu vào: GRAY => Mã đầu ra: BCD 8-4-2-1**

+**<u>BÀI 67</u>:** dùng cho hệ đếm thập tam phân(0,1,…,9,A,B,C)

- **Mã đầu vào: GRAY => Mã đầu ra: Bù 1**

+**<u>BÀI 68</u>:** dùng cho hệ đếm thập tứ phân(0,1,…,9,A,B,C,D)

- **Mã đầu vào: GRAY => Mã đầu ra: Bù 2**

+**<u>BÀI 69</u>:** dùng cho hệ đếm thập ngũ phân(0,1,…,9,A,B,C,D,E)

- **Mã đầu vào: GRAY => Mã đầu ra: GRAY+ 3**

+**<u>BÀI 70</u>:** dùng cho hệ đếm thập lục phân(0,1,…,9,A,B,C,D,E,F)

- **Mã đầu vào: GRAY => Mã đầu ra: BCD 5421**

+**<u>BÀI 71:</u>** dùng cho hệ đếm thập phân(0,1,…,9)

- **Mã đầu vào: BÙ_1 => Mã đầu ra: BCD 7421**

+**<u>BÀI 72</u>:** dùng cho hệ đếm thập nhất phân(0,1,…,9,A)

- **Mã đầu vào: BÙ_1 => Mã đầu ra: BCD 2421**

+**<u>BÀI 73</u>:** dùng cho hệ đếm thập nhị phân(0,1,…,9,A,B)

- **Mã đầu vào: BÙ_1 => Mã đầu ra: GRAY**

+**<u>BÀI 74</u>:** dùng cho hệ đếm thập tam phân(0,1,…,9,A,B,C)

- **Mã đầu vào: BÙ_1 => Mã đầu ra: Dư 3**

+**<u>BÀI 75</u>:** dùng cho hệ đếm thập tứ phân(0,1,…,9,A,B,C,D)

- **Mã đầu vào: BÙ_1 => Mã đầu ra: BCD 5121**

+**<u>BÀI 76</u>:** dùng cho hệ đếm thập ngũ phân(0,1,…,9,A,B,C,D,E)

- **Mã đầu vào: BÙ_1 => Mã đầu ra: BCD 8-4-2-1**

+**<u>BÀI 77</u>:** dùng cho hệ đếm thập lục phân(0,1,…,9,A,B,C,D,E,F)

- **Mã đầu vào: BÙ_1 => Mã đầu ra: 8421**

+**<u>BÀI 78</u>:** dùng cho hệ đếm thập phân(0,1,…,9)

- **Mã đầu vào: BÙ_1 => Mã đầu ra: Bù 2**

+**<u>BÀI 79</u>:** dùng cho hệ đếm thập nhất phân(0,1,…,9,A)

- **Mã đầu vào BÙ_1 => Mã đầu ra: GRAY+ 3**

+**<u>BÀI 80</u>:** dùng cho hệ đếm thập nhị phân(0,1,…,9,A,B)

- **Mã đầu vào: BÙ_1 => Mã đầu ra: BCD 5421**

+**<u>BÀI 81:</u>** dùng cho hệ đếm thập tam phân(0,1,…,9,A,B,C)

- **Mã đầu vào: BÙ_2 => Mã đầu ra: BCD 7421**

+**<u>BÀI 82</u>:** dùng cho hệ đếm thập tứ phân(0,1,…,9,A,B,C,D)

- **Mã đầu vào: BÙ_2 => Mã đầu ra: BCD 2421**

+**<u>BÀI 83</u>:** dùng cho hệ đếm thập ngũ phân(0,1,…,9,A,B,C,D,E)

- **Mã đầu vào: BÙ_2 => Mã đầu ra: GRAY**

+**<u>BÀI 84</u>:** dùng cho hệ đếm thập lục phân(0,1,…,9,A,B,C)

- **Mã đầu vào: BÙ_2 => Mã đầu ra: Dư 3**

+**<u>BÀI 85</u>:** dùng cho hệ đếm thập phân(0,1,…,9)

- **Mã đầu vào: BÙ_2 => Mã đầu ra: BCD 5121**

+**<u>BÀI 86</u>:** dùng cho hệ đếm thập nhất phân(0,1,…,9,A)

- **Mã đầu vào: BÙ_2 => Mã đầu ra: BCD 8-4-2-1**

+**<u>BÀI 87</u>:** dùng cho hệ đếm thập nhị phân(0,1,…,9,A,B)

- **Mã đầu vào: BÙ_2 => Mã đầu ra: Bù 1**

+**<u>BÀI 88</u>:** dùng cho hệ đếm thập tam phân(0,1,…,9,A,B,C)

- **Mã đầu vào: BÙ_2 => Mã đầu ra: Bù 2**

+**<u>BÀI 89</u>:** dùng cho hệ đếm thập tứ phân(0,1,…,9,A,B,C,D)

- **Mã đầu vào: BÙ_2 => Mã đầu ra: GRAY+ 3**

+**<u>BÀI 90</u>:** dùng cho hệ đếm thập ngũ phân(0,1,…,9,A,B,C,D,E)

- **Mã đầu vào: BÙ_2 => Mã đầu ra: BCD 5421**

+**<u>BÀI 91:</u>** dùng cho hệ đếm thập lục phân(0,1,…,9,A,B,C,D,E,F)

- **Mã đầu vào: GRAY+ 3 => Mã đầu ra: BCD 7421**

+**<u>BÀI 92</u>:** dùng cho hệ đếm thập phân(0,1,…,9)

- **Mã đầu vào: GRAY+ 3 => Mã đầu ra: BCD 2421**

+**<u>BÀI 93</u>:** dùng cho hệ đếm thập nhất phân(0,1,…,9,A)

- **Mã đầu vào: GRAY+ 3 => Mã đầu ra: GRAY**

+**<u>BÀI 94</u>:** dùng cho hệ đếm thập nhị phân(0,1,…,9,A,B)

- **Mã đầu vào: GRAY+ 3 => Mã đầu ra: Dư 3**

+**<u>BÀI 95</u>:** dùng cho hệ đếm thập tam phân(0,1,…,9,A,B,C)

- **Mã đầu vào: GRAY+ 3 => Mã đầu ra: BCD 5121**

+**<u>BÀI 96</u>:** dùng cho hệ đếm thập tứ phân(0,1,…,9,A,B,C,D)

- **Mã đầu vào: GRAY+ 3 => Mã đầu ra: BCD 8-4-2-1**

+**BÀI 97**: dùng cho hệ đếm thập ngũ phân(0,1,…,9,A,B,C,D,E)

- Mã đầu vào: GRAY+ 3 => Mã đầu ra: Bù 1

+**BÀI 98**: dùng cho hệ đếm thập lục phân(0,1,…,9,A,B,C,D,E,F)

- Mã đầu vào: GRAY+ 3 => Mã đầu ra: Bù 2

+**BÀI 99**: dùng cho hệ đếm thập phân(0,1,…,9)

- Mã đầu vào: GRAY+ 3 => Mã đầu ra: 8421

+**BÀI 100**: dùng cho hệ đếm thập nhất phân(0,1,…,9,A)

- Mã đầu vào: GRAY+ 3 => Mã đầu ra: BCD 5421

+**BÀI 101:** dùng cho hệ đếm thập nhị phân(0,1,…,9,A,B)

- Mã đầu vào: DƯ_3 => Mã đầu ra: BCD 7421

+**BÀI 102**: dùng cho hệ đếm thập tam phân(0,1,…,9,A,B,C)

- Mã đầu vào: DƯ_3 => Mã đầu ra: BCD 2421

+**BÀI 103**: dùng cho hệ đếm thập tứ phân(0,1,…,9,A,B,C,D)

- Mã đầu vào: DƯ_3 => Mã đầu ra: GRAY

+**BÀI 104**: dùng cho hệ đếm thập ngũ phân(0,1,…,9,A,B,C,D,E)

- Mã đầu vào: DƯ_3 => Mã đầu ra: 8421

+**BÀI 105**: dùng cho hệ đếm thập lục phân(0,1,…,9,A,B,C,D,E,F)

- Mã đầu vào: DƯ_3 => Mã đầu ra: BCD 5121

+**BÀI 106**: dùng cho hệ đếm thập phân(0,1,…,9)

- Mã đầu vào: DƯ_3 => Mã đầu ra: BCD 8-4-2-1

+**BÀI 107**: dùng cho hệ đếm thập nhất phân(0,1,…,9,A)

- Mã đầu vào: DƯ_3 => Mã đầu ra: Bù 1

+**BÀI 108**: dùng cho hệ đếm thập nhị phân(0,1,…,9,A,B)

- Mã đầu vào: DƯ_3 => Mã đầu ra: Bù 2

+**BÀI 109**: dùng cho hệ đếm thập tam phân(0,1,…,9,A,B,C)

- Mã đầu vào: DƯ_3 => Mã đầu ra: GRAY+ 3

+**BÀI 110**: dùng cho hệ đếm thập tứ phân(0,1,…,9,A,B,C,D)

- Mã đầu vào: DƯ_3 => Mã đầu ra: BCD 5421

****

Khoa Công nghệ Thông tin

Bộ môn **HỆ THỐNG NHÚNG**

ĐỀ BÀI

LẬP TRÌNH HỆ THỐNG

Mỗi đồ án gồm 2 đề bài, bài 1) viết chương trình bằng ngôn ngữ lập trình C và bài 2) viết bằng ngôn ngữ lập trình Assembler

1) Đồ án được thực hiện theo nhóm(Mỗi nhóm chỉ gồm 2 sinh viên)

2) Cấm 2 nhóm đồ án trong 1 nhóm học tập xxA (hoặc xxB) làm chung 1 đề bài, muốn thế sinh viên có sốTT là 1, thì thực hiện đề số 1. Sinh viên có số TT là 2, thì thực hiện đề số 2, vv…1 sinh viên ở ½ nhóm học tập phía dưới, ghép với sinh viên đã có đồ án ở ½ phía trên nhóm học tập.

3) Chia đề bài ra làm nhiều phần nhỏ (tùy ý), xây dựng thuật toán chi tiết cho từng phần, viết chương trình con (hàm, thủ tục),macro, để thực hiện phần nhỏ nầy, hoặc viết ISR(Interupt Service Rountine) rồi gọi liên kết vào chương trình chính.

4) Các đề bài ngôn ngữ C đã có thuật tóan cơ bản, các đề tài ngôn ngữ ASM đều dùng các chức năng của ngắt 21h, ngắt 10h,…trong bài giảng ***lập trình hệ thống****.*

*5)**Phần mở rộng:**(phần bắt buộc) Nhóm đồ án tự xây dựng thêm các chương trình con,macro,ISR.bổ sung vào đồ án, sao cho hợp ngữ cảnh.*

6) Khi viết chương trình cần thường xuyên giải thích, chương trình trở nên sáng tỏ, rỏ ràng.(Quy định số dòng giải thích ít nhất bằng ½ số dòng lệnh).

7) In kết quả chạy chương trình vào đồ án.

8) Nhóm tự cho điểm:1 sinh viên điểm 10, và 1 sinh viên điểm 9.

Lưu ý:

Giáo viên sẽ hướng dẫn tại lớp(trong giờ hướng dẫn đồ án môn học) vì thế sinh viên cần tìm hiểu đề tài , xây dựng các thuật toán và học nâng cao kỹ thuật lập trình C, Assembler để thảo luận, phản biện cùng giáo viên.

****

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án: LẬP TRÌNH HỆ THỐNG

Đề 1

****

1) lập trình C:

Viết 1 chương trình C thực hiện các công việc sau:

1)Viết hàm **void* NHAP(),** để nhập số nguyên kiểu int ***(nhập số thập phân, hàm cho phép xóa dấu âm hoặc chữ số thập phân(0⇔9) khi gõ nhầm, bằng cách ấn phím Backspace, hàm không cho hiển thị ký tự khác ký tự thập phân lên màn hình, đồng thời hàm cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị -32768 ⇔32767.(dùng <dos.h>, union REGS và software interrupt 0x21, 0x10)***,gọi hàm NHAP(),để nhập ma trận **int *a ;** (n*n phần tử với 6<=**n** <=10)

2) Hãy sao chép ma trận int*a thành mảng 1 chiều int *x, và ngược lại.

3)Tìm int **min=mini+ minj,** trong đó mini là phần tử nhỏ nhất ở hàng i, và minj là phần tử nhỏ nhất ở cột j(cột j ứng với phần tử lớn nhất cuối cùng trên hàng i ma trận a).

4) Đổi int **min** ra chuỗi ký tự số hex : char *hex ;(min có ở câu b).Nếu **min** là số âm, thì viết theo 2 quy cách : số bù 2 và số có dấu ’-’ đặt trước.

vd : min=-15 => char*hex=”0xFFF1”= ”-0xf”

5)Tính **k** là tổng các phần tử nào của chuỗi hex, mà chữ số hex của nó có 2 bit tận cùng bên trái là bit 1, nếu không k là tổng các phần tử có 2bit tận cùng bên phải là 10 hoặc 01.(dùng phép toán thao tác bit).

gợi ý : *- ma trận int*A ; có n*m phần tử.*
- độ lớn kiểu dữ liệu char(1 byte), int(2 byte), float(4 byte),long(4 byte),double(8 byte)..
*- ma trận int *A ; được bố trí trong bộ nhớ hết hàng 0, đến hàng 1,…đến hàng n-1 (trật tự hàng), con trỏ A, trỏ vào phần tử đầu tiên của vùng nhớ nầy.*
hoặc bố trí hết cột 0, đến cột 1,….,đến cột m-1(trật tự cột)
- Tìm địa chỉ phần tử A[i][j], phải tìm hàng i và phần tử thứ j trên hàng i
- (mỗi hàng có m phần tử, mỗi cột có n phần tử)
*'hàng 0: vị trí bắt đầu là A , hàng 1 có vị trí bắt đầu là A+m*2.(độ lớn kiểu int là 2)*
*hàng 2: vị trí bắt đầu là A+2*m*2,… , hang i có vị trí bắt đầu là A+i*m*2.*
*- phần tử thứ j trên 1 hàng cách vị trí đầu hàng j*2 byte*
*- như vậy phần tử A[i][j] có địa chỉ là: A+(i*m+j)*2(trật tự hàng)*
*tương tự lưu trữ theo trật tự cột là: A+(i+j*n)*2*
*- con trỏ C,được bố trí để trỏ theo độ lớn kiểu dữ liệu, như vậy địa chỉ phần tử A[i][j] là: A+i*m+j(trật tự hàng), hoặc A+i*n+j(trật tự cột)*

---

2) lập trình ASSEMBLER :

Viết chương trình ASM, với ISR(chương trình con phục vụ ngắt) chặn ngắt 1Ch, để hiện dòng ''KHOA CONG NGHE THONG TIN_DAI HOC BACH KHOA DA NANG thứ, ngày, tháng, năm, giờ, phút, giây'' hiện hành ở góc trái màn hình.

gợi ý: bản mạch điều khiển màn hình(đơn vị Input/Output : I/O) trong máy tính được thiết kế theo phương pháp 'IO mapped MEMORY',đặt tại địa chỉ segment là 0B800H, địa chỉ offset tùy thuộc vị trí đặt dòng thông báo. Để hiển thị 1 ký tự cần bố trí 2 byte(dùng AX), byte thấp(AL) chứa mã ASCII của ký tự, byte cao(AH) chứa thuộc tính,và theo phương pháp thiết kế nầy ta dùng lệnh : MOV để niêm yết ký tự lên màn hình.(chú ý : lệnh OUT chỉ dùng cho đơn vị I/O thiết kế theo phương pháp : IO izolated MEMORY như cổng COM địa chỉ nền 3F8h,địa chỉ nền của PIC(Priority Interrupt Controler)8259A tại 20h vv..).

gọi ISR : + mov AH, 2Ah

int 21h ;trả về : CX<=năm, DH<=tháng, DL<=ngày, AL<=thứ(0 :chủ nhật,1 :thứ hai,..6 :thử bảy)

+ mov AH, 2CH ; int 21h ; trả về :CH<=giờ, CL<=phút , DH<=giây

==========

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án: LẬP TRÌNH HỆ THỐNG

Đề 2

****

1) lập trình C:

Viết 1 chương trình C để thực hiện các công việc sau:

1)Viết hàm **void* NHAP(),** để nhập số nguyên kiểu int ***(nhập số thập nhất phân, hàm cho phép xóa dấu âm hoặc chữ số thập nhất phân(0⇔A hoặc a) khi gõ nhầm, bằng cách ấn phím Backspace, hàm không cho hiển thị ký tự khác ký tự thập nhất phân lên màn hình, đồng thời hàm cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị : -2268a(-32768) ⇔22689(32767).(dùng <dos.h>, union REGS và software interrupt 0x21, 0x10)***,gọi hàm NHAP(),để nhập n và ma trận **int *a ;** (n*n+1 phần tử với 5<=**n** <=10).Taọ tệp tin kiểu nhị phân "MATRIX.BIN", mà mẫu tin đầu tiên được ghi vào là n(5<=n<=10) tiếp theo là n*(n+1)phần tử kiểu int,của ma trận int*a;

b) Ghi lại tệp tin kiểu nhị phân ''MATRIX.BIN'' thành tệp tin kiểu văn bản ''MTRAN.TXT''

c) Đọc tệp "MTRAN.TXT", lấy phần tử đầu tiên đưa vào int n ; tiếp theo là n*(n+1)phần tử lưu vào ma trận int *a ; hãy xóa cột chứa phần tử nhỏ nhất cuối cùng, để có int*a là ma trận vuông n*n phần tử.

d)Tạo chuỗi ký tự số bát phân char *oct ; mà *(oct+i), là ký tự số bát phân tận cùng bên trái của phần tử cột giữa trên hàng i ma trận a.(chú ý đóng chuỗi bằng ký tự '\0' (NULL)). ***vd : *(a+i*n+(n-1)/2)=140 =0210=> *(oct+i)= '2'=062***.

e) Xét xem chuỗi char *oct; có phải là chuỗi đối xứng ?

gợi ý : ***ma trận int*A ; có n*m phần tử(xem gợi ý ở đề 1).***

gợi ý: để đổi 1 chuỗi số hệ đếm bất kỳ về trị thập phân (int d ;) dùng phương pháp Horner

d=(…((*(s+0)*cs+*(s+1))*cs+*(s+2))*cs+…+*(s+k-1))*cs+*(s+k)

cs: cơ số của hệ đếm

****(s+i) là ký tự thứ i (hệ đếm cơ số cs), cần đổi ra trị tương ứng***

hoặc viết theo thao tác bit. vd: char* s; lưu chuỗi ký tự số bát phân, đổi ra trị thập phân tương ứng

int d=0; while(*(s+i)) d=(d<<3) |(*(s+i++) &0x07); chú ý: nếu char *s; là chuỗi ký tự số bát phân biểu diễn số âm, thì phải bù 2(d), và đặt dấu âm vào.

---

2) lập trình ASSEMBLER :

Viết một chương trình, có thủ tục NHAP (*để nhập N kiểu DB xóa được khi gõ nhầm chữ số, không hiển thị ký tự khác ký tự thập phân và xét điều kiện N chứa 1 chữ số BCD không nén(unpacked BCD) 00h ⇔09h,nếu gỏ giá trị nằm ngoài miền nầy, thủ tục sẽ tự động xóa và cho phép nhập lại* **) ,gọi thủ tục NHAP, để nhập 2 mảng 1 chiều A[24], B[20] chứa số unpacked BCD , tính C[25]= A[24]+B[20]. Hãy đổi 4 byte thấp nhất của C[25] ra số thập phân tương ứng, lưu trong biến N kiểu DW.In C[25] và N**

Thuật toán:hiệu chỉnh thập phân sau khi cọng 2 số BCD không nén:

if ((AL AND 0Fh>9)OR(AF=1))

. AL=AL+6 .AF=1=CF AL=AL and 0Fh

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án: LẬP TRÌNH HỆ THỐNG

Đề 3

****

1) lập trình C:

Viết hàm **char* NHAP()**, để nhập số nguyên kiểu char***(hàm cho phép xóa chữ số gõ nhầm, không cho hiển thị ký tự khác ký tự thập phân, đồng thời thủ tục cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị 0x30⇔0x31)***, gọi hàm NHAP(), để nhập chuỗi dài 31 byte, lưu 31 ký tự bit(char *m1;) và chuỗi m2 dài 25 byte(char *m2; lưu 25 ký tự bit), thực hiện m3= m1+m2. (char *m3; m3 dài 32 byte chứa 32 ký tự bit) , hãy đổi 16 byte thấp của m3 ra số thập phân N kiểu int. In kết quả.

gợi ý: tính phần tử(ký tự bit)*(m3+i)=?

tam= *(m2+i)&1+*(m1+i)&1+cf_{i-1} (cf_{i-1} là bit nhớ của lần cọng trước, tam có thể là 0,1,2,3)

****(m3+i)=tam%2 \0x30;***

Cf_i=tam/2;(cf_i là bit nhớ của lần cọng hiện tại)

---

2) lập trình ASSEMBLER :

Viết một chương trình, có thủ tục NHAP(*hàm cho phép xóa chữ số gõ nhầm, bằng cách ấn phím Backspace, hàm không cho hiển thị ký tự khác ký tự thập phân lên màn hình, đồng thời hàm cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị -32768⇔32767),***gọi thủ tục NHAP để nhập ma trận A[5][5]và B[5][5]kiểu DW.Tìm ma trận tích C[5][5]= A[5][5]*B[5][5]. In kết quả**

gợi ý : - *ma trận A[n][m] ; có n*m phần tử.*

- *độ lớn kiểu dữ liệu DB(1 byte), DW(2 byte), DD(4 byte),..*
- *ma trận A được bố trí trong bộ nhớ hết hàng 0, đến hàng 1,…đến hàng n-1 (trật tự hàng)*
  hoặc hết cột 0, đến cột 1,….,đến cột m-1(trật tự cột)
- *Tìm địa chỉ A[i][j], phải tìm hàng i và phần tử thứ j trên hàng i*
- *(mỗi hàng có m phần tử, mỗi cột có n phần tử)*
  *'hàng 0: vị trí bắt đầu là A , hang 1 có vị trí bắt đầu là A+m*2.(độ lớn kiểu DW là 2)*
  *hàng 2: vị trí bắt đầu là A+2*n*2,… , hang i có vị trí bắt đầu là A+i*m*2.*

*-phần tử thứ j trên 1 hàng cách vị trí đầu hàng j*2 byte*

*- như vậy phần tử A[i][j] có địa chỉ là: A+(i*m+j)*2(trật tự hàng)*
*tương tự lưu trữ theo trật tự cột là: A+(i+j*n)*2*

==========

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án: LẬP TRÌNH HỆ THỐNG
Đề 4
****

1) lập trình C:

Viết 1 chương trình C thực hiện các công việc sau:

1)Viết các hàm mô phỏng các hàm chuẩn của string.h: strlen(), strstr(), strchr(), strrchr().

2) Viết các hàm để xử lý bài toán tự tạo, có gọi các hàm mô phỏng nầy để thử nghiệm.

3) Cho trước 1 chuỗi **char *s ='' 2, 4,7, 4 ,8 ,9,9, 3,6, 4 , 0 '';** hãy tạo ra chuỗi số thập phân, (bằng cách xóa hết các ký tự trống , dấu phẩy và số 0)

4) Chia các phần tử của chuỗi char *s ; thành 2 nhóm sao cho hiệu của tổng các phần tử nhóm nầy và tổng các phần tử nhóm kia là 1 số dương nhỏ nhất.

5) Đổi chuỗi **s** ra số nguyên long n=2478936; rồi đổi n ra chuỗi số nhị phân char*bin; và chuỗi số bù 2: char*bu2 của char*bin.

gợi ý : +Tìm cặp x0, y0(x0>y0) có hiệu nhỏ nhất, cặp x1, y1(x1>y1) có hiệu nhỏ thứ 2,…như vậy hiệu(a0 +a1+..)- (b0+b1+..) sẽ nhỏ nhất.

- vd: viết hàm mô phỏng hàm strlen() và thử nghiệmừ

```
int len( char *s)
{ int i=0; for(; *(s+ ++i); ); return i;}// hoặc{ const char *st=s; while(*++st){} return(st-s);}
 char *ch="Bach_Khoa";  printf ("Do dai that =%d=%d\n",strlen(ch),len(ch));
```

---

2) lập trình ASSEMBLER :

Viết 1 chương trình, có thủ tục NHAP, để nhập số nguyên kiểu DB(*thủ tục cho phép xóa chữ số gõ nhầm, không cho hiển thị ký tự khác ký tự thập phân, đồng thời thủ tục cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị 0⇔1*)**, gọi thủ tục NHAP, để nhập mảng m1 DB 31 dup(0) và m2 DB 25 dup(0)(mỗi byte chứa 1 bit), rồi thực hiện m3= m1+m2.(m3 DB 32 dup(0)), hãy đổi 16 byte thấp của m3 ra số thập phân N kiểu DW.In kết quả M3 và N.**

gợi ý : *so2 DB 2*

Al= m1[i]+m2[i]+C_{i-1}(bit nhớ từ lần cọng trước đó)

Ah=0

div so2

m3[i]= Ah

C= Al(bit nhớ của lần cọng hiện tại)

==========

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án: LẬP TRÌNH HỆ THỐNG
Đề 5
****

1) lập trình C:

Viết 1 chương trình C, thực hiện các công việc sau:

1) Viết hàm **void* NHAP(),** để nhập số nguyên kiểu int không âm(***hàm cho phép xóa chữ số gõ nhầm,hàm không cho hiển thị ký tự khác ký tự thập phân, đồng thời hàm cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị 0⇔15*),** gọi hàm void *NHAP() để nhập mảng 1 chiều n phần tử **int *a={7,7,15,15,13,13,2,2,2,2,14, 9,9,9,0,0,11,11};** là các trị chữ số hex.

2) hãy xỏa các phần tử giống nhau chỉ để lại 1 phần tử, rồi xóa tiếp phần tử nào bằng 0 =>

int*a={7,15,13,2,14,9,11};.

3) đổi mảng int*a, ra số thập phân kiểu long k.

4) Hãy trích 4 chữ số tận cùng bên phải của k để đổi ra h kiểu unsigned int.

5) viết hàm **char*THU()** để trả về thứ mấy trong tuần của ngày x/11/2017. Với x=h%31, biết rằng ngày 1/1/2017 là chủ nhật.

gợi ý: ***để đổi 1 dãy số hệ đếm bất kỳ về trị thập phân (int d ;) dùng phương pháp Horner***

d=(…((*(a+0)*cs+*(a+1))*cs+*(a+2))*cs+…+*(a+k-1))*cs+*(a+k)

cs: cơ số của hệ đếm . *(a+i) là phần tử thứ i

---

2) lập trình ASSEMBLER :

Viết chương trình ASM để mô phỏng lệnh TYPE

==========

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án: LẬP TRÌNH HỆ THỐNG

Đề 6

****

1) lập trình C:

Viết 1 chương trình C để thực hiện các công việc sau:

Viết hàm **char NHAP()**, để nhập mã ASCII số hex***(hàm cho phép xóa khi gõ nhầm ký tự hex, không cho hiển thị ký tự khác ký tự hex và nếu nhập giá trị nằm ngoài miền trị 30h⇔39h,hoặc nằm ngoài miền trị 61h⇔ 7Ah,và 41h⇔5Ah thì hàm cho phép nhập lại.***) gọi hàm NHAP() để nhập chuỗi **char hex;** gồm 16 ký tự hex, trong đó có ít nhất 4 ký tự thập phân.(chú ý đóng chuỗi bằng '\0')

1) Hãy đổi 4 phần tử ở cuối chuỗi hex ra số thập phân int d;

2) Đổi d%10 ra ký tự thập phân, rồi chèn vào giữa chuỗi hex

3) Hãy xóa tất cả các chữ cái , rồi tiếp tục loại bỏ các chữ số sao cho 4 chữ số còn lại cuối cùng(theo đúng thứ tự nhập và chèn) tạo nên 1 số lớn nhất (k).

4) Xét xem d và k, trị nào có số bit 1 nhiều hơn, thì in số nhị phân của trị đó ra (không đổi ra số nhị phân).

5) Xét xem k và d có phải 2 số hữu nghị? Nếu số nầy bằng tổng các ước số thật sự của số kia.

gợi ý + k là số hoàn hảo, khi k bằng tổng các ước số tự nhiên của nó, không kể nó vd: 28=1+2+4+7+14

=>56 và 28 là 2 số hữu nghị 56=1+2+4+7+14+28

gợi ýcâu 1 : để đổi 1 chuỗi số hệ đếm bất kỳ về trị thập phân (int d ;) dùng phương pháp Horner

d=(…((*(s+0)*cs+*(s+1))*cs+*(s+2))*cs+…+*(s+k-1))*cs+*(s+k)

cs: cơ số của hệ đếm

****(s+i) là ký tự thứ i (hệ đếm cơ số cs), cần đổi ra trị tương ứng***

hoặc viết theo thao tác bit. vd: char* s; lưu chuỗi ký tự số hex, đổi ra trị thập phân tương ứng

d=0; while(*(s+i))d=(d<<4)|(*(s+i)<='9'?*(s+i++)&0x0f:*(s+i)<='F'?*(s+i++)-0x37:*(s+i++)-'a'+10); chú ý: nếu char *s; là chuỗi ký tự số hex biểu diễn số âm, thì phải bù 2(d), và đặt dấu âm vào.

gợi ý câu 3: số abcd có trị lớn nhất khi a>=b>=c>=d, tiến hành tuyển lần lượt các số lớn nhất, lớn thứ 2,…từ đầu chuỗi đến cuối để tạo chuỗi còn lại có trị lớn nhất.chú ý giới hạn cuối của vòng lặp tìm số lớn nhất

---

2) lập trình ASSEMBLER:

Viết 1 chương trình .ASM, để thực hiện các công việc sau:

Viết thủ tục NHAP(*thủ tục cho phép xóa chữ số gõ nhầm, bằng cách ấn phím Backspace, thủ tục không cho hiển thị ký tự khác ký tự thập phân lên màn hình, đồng thời thủ tục cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị -32768⇔-1),* **gọi thủ tục NHAP để nhập 1 số âm rồi đổi ra mảng số hex.**

Viết ra cả 2 cách biểu diễn số âm : có dấu âm – đặt trước hoặc dạng số bù 2 vd : -15= -Fh =FFF1h.

==========

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án: LẬP TRÌNH HỆ THỐNG
Đề 7
****

1) lập trình C:

Viết một chương trình C để thực hiện các công việc sau:
Viết hàm **void* NHAP(),** để nhập số nguyên kiểu int ***(nhập số thập nhị phân, hàm cho phép xóa dấu âm hoặc chữ số thập nhị phân(0⇔B(b)) khi gõ nhầm, bằng cách ấn phím Backspace, hàm không cho hiển thị ký tự khác ký tự thập nhị phân lên màn hình, đồng thời hàm cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị -16b68(-32768 ⇔ 16B67(32767).(dùng <dos.h>, union REGS và software interrupt 0x21, 0x10).***

gọi hàm NHAP() để nhập số nguyên, rồi chèn vào danh sách liên kết: (có thể dùng cách chèn node đầu, chèn node cuối, hoặc chèn đệ quy)

```
struct SO{
            int  data;
               struct SO *link;
        };
typedef struct SO* ctro;
```

+dùng các phương pháp sắp xếp để sắp xếp các phần tử trong danh sách. ***gợi ý: nhập đệ quy, không dùng hàm chèn node, không cần khởi tạo NULL cho con trỏ đầu danh sách***

```
ctro dq()
  { it x; ctro p;
      x=nhap(); if(!x)return NULL;
       p=(ctro) malloc(sizeof( struct SO));
        p->data=x;
        p->link=dq(); return p;
   }
```

---

2) lập trình ASSEMBLER :

Viết chương trình ASM để mô phỏng lệnh DEL

==========

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án: LẬP TRÌNH HỆ THỐNG
Đề 8
****

1) lập trình C:

Viết 1 chương trình C để thực hiện các công việc sau:

a) Viết hàm **void* NHAP(),** để nhập ký tự chữ cái in thường kiểu char***(hàm cho phép xóa chữ cái gõ nhầm,hàm không cho hiển thị ký tự khác ký tự chữ cái in thường, đồng thời hàm cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị 'a'⇔'z'),***gọi hàm NHAP(), để nhập chuỗi **char *ch;**

b) Hãy nén chuỗi như sau: với chuỗi con gồm n ký tự 'x' giống nhau, thì thay bằng nx vd: aaabbbbb= 3a5b.

c) Hãy giải nén để có lại chuỗi ban đầu.

d) Hãy xóa các ký tự giống nhau, chỉ để lại 1 ký tự.

e) Xét xem có phải là chuỗi đối xứng ?

gợi ý : + dùng hàm chuẩn sprintf(char *ctr, const char*format,…) hoạt động tương tự như printf(), nhưng không xuất ra thiết bị xuất chuẩn(stdout), mà trả về số int là chiều dài của chuỗi do hàm tạo thành(không tính NULL). đối số đầu tiên là địa chỉ chuỗi đích, các đối số tiếp theo tương tự printf() ;

+dùng hàm chuẩn void* memset(void *ctr, int ch,size_t n) trị ch được sao chep đến n byte đầu tiên của vùng nhớ, dược trỏ bởi *ctr.

vd : giải nén chuỗi char*s, đưa vào chuỗi char*st

```
size_t d=0,i=0; for( ;*(s+i) ;++i ) if (isdigit(*(s+i)){ memset(st+d,*(s+i++),*(s+i)&0x0f) ; d+=*(s+i)&0x0f ; i++ ;}else d+=sprintf( st+d, "%c",*(s+i));
```

---

2) <u>lập trình ASSEMBLER</u> :

Viết một chương trình, có thủ tục NHAP (*để nhập N kiểu DB xóa được khi gõ nhầm chữ số, không hiển thị ký tự khác ký tự thập phân và xét điều kiện N chứa 1 chữ số BCD không nén(unpacked BCD) 00h⇔09h,nếu gỏ giá trị nằm ngoài miền nầy, thủ tục sẽ tự động xóa và cho phép nhập lại* **) ,gọi thủ tục NHAP, để nhập 2 mảng 1 chiều A[10], B[10] chứa số unpacked BCD , tính C[20]= A[10]*B[10]. In kết quả.**

<u>Gợi ý</u> : giống khi nhân số thập phân, ta tách từng chữ số BCD từ phải sang trái nhân với số hạng thứ nhất , kết quả mỗi lần nhân dịch qua trái 1 đơn vị cho đến hết, sau đó ta cọng các kết quả đó lại.

==========

KHOA CÔNG NGHỆ THÔNG TIN

<u>Đồ án</u>: LẬP TRÌNH HỆ THỐNG

<u>Đề 9</u>

****

1) <u>lập trình C</u>:

Viết <u>1 chương trình C</u>, thực hiện các công việc sau:

a) Tạo 1 ma trận vuông **int *a;** (6<=n<=15) có dạng zích zắc như sau

1 2 3

6 5 4

7 8 9

b) Tìm mảng 1 chiều int*x, mà *(x+i) là phần tử "YÊN_NGỰA" của hàng i đó là phần tử lớn nhất/nhỏ nhất hàng i , đồng thời là phần tử nhỏ nhất/ lớn nhất cột j, cột j là cột chứa nó.nếu hàng i không có phần tử "YÊN_NGỰA", thì *(x+i) là tổng các phần tử nào của cột i có byte thấp có số bit 1 là số chẵn, hãy tính k là tổng các phần tử đứng trước phần tử lẻ cuối cùng của int*x;

c)Hãy chuyển các phần tử chẵn mảng x sang trái, và chuyển các phần tử lẽ sang phải.

d) xóa đi các phần tử nào của mảng **x<=k%9**.(không được dùng mảng phụ và không được sắp xếp)

e)Hãy tính định thức ma trận int*a..

<u>gợi ý</u>: +dung lệnh void*memmove(void *dest, const void*src,size_t n);chép n byte từ vùng nhớ được trỏ bởi *src, đến vùng nhớ được trỏ bởi *dest, trả về 1 con trỏ trỏ đến dest.

+so sánh s với các phần tử *(x+i) từ đầu mảng(i=0), nếu *(x+i)<k,thì chuyển(n-(i+1)*sizeof(kiểu)) byte, về phía trái mảng 1 vị trí (lệnh memmove (x+i, x+i+1,*n-(i+1)*sizeof(int))) , giảm n đi 1), nếu không thì tăng i.

---

2) <u>lập trình ASSEMBLER</u> :

Viết 1 chương trình, có thủ tục NHAP, để nhập số thập phân, kiểu DW không dấu(*thủ tục cho phép xóa chữ số gõ nhầm, không cho hiển thị ký tự khác ký tự thập phân, đồng thời thủ tục cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị 0⇔65535*)**, gọi thủ tục NHAP, để nhập M1 DW 10 dup(0) , tìm tổng các phần tử nào là số Palindrome, số hoàn hảo, số chính phương ….in kết quả(dùng ngắt)**

==========

KHOA CÔNG NGHỆ THÔNG TIN

<u>Đồ án</u>: LẬP TRÌNH HỆ THỐNG

<u>Đề 10</u>

****

1) <u>lập trình C</u>:

Viết hàm **void* NHAP(),** để nhập ký tự chữ cái in thường kiểu char**(*hàm cho phép xóa chữ cái gõ nhầm,hàm không cho hiển thị ký tự khác ký tự chữ cái, đồng thời hàm cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị 'a'⇔'z' và 'A' ⇔'Z'*),**gọi hàm NHAP(), để nhập họ và tên sinh viên, để quản lý hồ sơ sinh viên, tổ hợp dữ liệu theo danh sách liên kết, với mỗi sinh viên cần quản lý các thông tin như: họ và tên, điểm của 5 môn thi trong học kỳ, điểm trung bình, bao gồm các công việc chính sau đây:

a) hàm void tao_lap(), để nhập dữ liệu của sinh viên vào danh sách.

b) hàm void chen(), để chén dữ liệu của sinh viên mới vào danh sách.

c) hàm void loai_bo(), để loại bỏ sinh viên khỏi danh sách
d) hàm void sua_doi(), để chĩnh sữa thông tin của sinh viên
e) hàm void sap_xep(), để sắp xếp danh sách theo bảng chữ cái
In kết quả

vi dụ: ***struct svien***
{ char hten[35];
float toan,ly,ltht,do_an,thdc,dtb;
struct svien*link;
};
typedef svien *ctro;
int n;// số sinh viên
ctro fd; // con trỏ đầu danh sách, cần khởi tạo NULL

vidu: sắp xếp theo bảng chữ cái, khai báo ctro p,q; for(p=fd;p->link!=NULL;p=p->link) for(q=p->link; q!=NULL;q=q->link) if(strcmp(p->hten,q->hten)>0)thì hoán đổi 2 bản ghi cho nhau.

---

2) lập trình ASSEMBLER :

Viết chương trình ASM dể Mô phỏng lệnh COPY

==========

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án: LẬP TRÌNH HỆ THỐNG
Đề 11
****

1) lập trình C:

Viết 1 chương trình C, thực hiện các công việc sau đây:
Viết hàm **char* NHAP(),** để nhập ký tự số thập phân kiểu char***(hàm cho phép xóa chữ số gõ nhầm,hàm không cho hiển thị ký tự khác ký tự thập phân, đồng thời hàm cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị '0'⇔'9')***,gọi hàm NHAP(), để nhập 2 chuỗi ký tự thập phân **char *a;** và **char *b;**
Thực hiện nhân chuỗi a với chuỗi b(trình bày giống như bài toán nhân tay).
gợi ý: + từng số của số bị nhân(chuỗi a)sẽ nhân với chuỗi số nhân (chuỗi b) từ phải sang trái, kết quả lưu trử thành 1 dòng nhân(chuỗi c) ngược lại (từ trái qua phải) với quá trình tính, cần kiểm tra số nhớ cuối cùng để đưa vào chuỗi c, và gán NULL vào cuối chuỗi c.
+cọng dòng nhân c vào chuỗi kết quả d, từ trái sang phải, bắt đầu tại vị trí i, với dòng nhân thứ i. chưa gán NULL vào cuối chuỗi d, vì còn cọng các dòng nhân c khác,nhưng phải dùng x để đánh dấu vị trí cuối chuỗi.Khi cọng xog tất cả các dòng nhân c vào d,thì gán '\0' vào d ở vị trí x nầy(chú ý khi in nhớ đảo ngược c và d)

---

2) lập trình ASSEMBLER :

Viết chương trình ghi dữ liệu vào tệp, đánh vào từ bàn phím và kết thúc bằng CTRL+Z. Tên tệp cũng vào từ bàn phím và kết thúc bằng ENTER

==========

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án: LẬP TRÌNH HỆ THỐNG
Đề 12
****

1) lập trình C:

Viết 1 chương trình C để thực hiện các công việc sau:

1) Hãy tạo tệp nhị phân "MATRIX.BIN" mà mẫu tin đầu tiên ghi n là số hàng/cột của 1 ma trận vuông , tiếp theo là n*n mẫu tin ghi n*n phần tử ma trận int***a** với (5<=**n** <=12).

2) Hãy đọc tệp nhị phân "MATRIX.BIN", lưu mẫu tin đầu tiên vào int n; tiếp theo là n*n phần tử của ma trận int*a; rồi taọ mảng 1 chiều **int *x** mà *(x+j) là tổng các phần tử đứng sau phần tử lẻ đầu tiên ở cột j ma trận **a,** nếu không, *(x+j) là tổng các phần tử nào trên cột j có ½ byte(là 1 chữ số hex) tận cùng bên trái là 1 chữ số BCD nén, nếu đúng thì chuyển đổi với chữ số hex ở giữa(k-1/2, với k là số chữ số hex của phần tử.)

vd: *(a+i*n+j)= 1AC9h, chuyển đổi 1 ⇔12(C)=> CA19h

3)Tính s là tổng các phần tử mảng x,mà byte cao có ½ byte phần thấp bằng giá trị đảo bit của ½ byte phần cao. vd : *(x+i)=0xC3xx, 5Axx,…

4)Tìm kiếm s %55 có trong mảng **int* x**? (Tìm kiếm với mảng đã sắp xếp).

5) Hãy hoán đổi chữ số tận cùng bên trái với chữ số tận cùng bên phải của s.

gợi ý : + Một số nguyên kiểu int có ½ byte tận cùng bên trái là 1 số BCD nén sẽ có giá trị từ 00xxxh ⇔9Fxxxh

---

2) lập trình ASSEMBLER :

Viết chương trình ASM để minh họa sử dụng chuột, hiển thị tọa độ chuột

==========

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án: LẬP TRÌNH HỆ THỐNG
Đề 13
****

1) lập trình C:

Viết chương trình thường trú, hiện đồng hồ ở góc phải phía trên màn hình.

gợi ý: ***+Để chặn ngắt, ta viết ISR_new (hàm phục vụ ngắt mới) cú pháp****: interrupt ISR_new(tham số)*
cất địa chỉ hàm ngắt chuẩn, trên bảng vecto ngắt , mà ta chặn bằng hàm chuẩn:
*void interrupt(*getvect(sh_ngat))(); với sh_ngat là số hiệu ngắt mà ta chặn(đề bài nầy là 0x1C)*
rồi đem địa chỉ của hàm ISR_new đặt vào bảng vecto, đè lên địa chỉ ngắt chuẩn 0x1C, như vậy khi gọi ngắt 0x1C thì hàm ISR_new được phục vụ:
*void setvect(int sh_ngat, void interrupt(*ISR_new)());*

---

2) lập trình ASSEMBLER:

Viết một chương trình, có thủ tục NHAP (*để nhập N kiểu DB xóa được khi gõ nhầm chữ số, không cho hiển thị ký tự khác ký tự thập phân và xét điều kiện N chứa 1 chữ số unpacked BCD 00h⇔09h*) **,gọi thủ tục NHAP, để nhập 2 mảng 1 chiều A[25], B[25] gồm 25 phần tử kiểu DB, chứa 25 chữ số BCD không nén, tính C[26]=A[25]-B[25]. Hãy đổi 4 byte thấp nhất của C[26] ra số thập phân tương ứng, lưu trong biến N kiểu DW. In C[26] và N .**

1) xét dấu: so sánh nội dung 2 ô nhớ đầu mảng A[1]với B[1]:

-nếu lớn hơn thì thực hiện A[25]- B[25], hiệu chỉnh thập phân, lưu kết quả vào C[26],
tiếp tục từ phải sang trái,....nạp 00h vào C[1](kết quả dương).

-nếu nhỏ hơn thì thực hiện B[25]- A[25], hiệu chỉnh thập phân, lưu kết quả vào C[26],
tiếp tục từ phải sang trái,...nạp '-' vào c[1](kết quả âm)

-nếu bằng nhau, thì so sánh nội dung 2 phần tử tiếp theo A[2] với B[2](kết quả như trên nhưng chỉ xử lý với 24 cặp phần tử còn lại..., cứ thế tiếp tục

2)Thuật toán hiệu chỉnh thập phân sau khi trừ 2 số BCD kông nén

if((AL AND 0Fh>9)OR(AF=1))
+ AL= AL-6
+AF=1=CF
+AL=AL AND 0Fh

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án: LẬP TRÌNH HỆ THỐNG
Đề 14
****

1) lập trình C:

Viết 1 chương trình C để thực hiện các công việc sau:

Viết hàm **char* NHAP(),** để nhập số nguyên kiểu unsigned char***(hàm cho phép xóa chữ số gõ nhầm,hàm không cho hiển thị ký tự khác ký tự thập phân, đồng thời hàm cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị 0 ⇔15),***gọi hàm NHAP(), để nhập mảng 1 chiều số hex **char *a;** gồm 8 phần tử.

a) Hãy tạo mảng 1 chiều char* x, mà *(x+i) cũng chính là *(a+i), nhưng có giá trị được tính trong hệ nhị phân có 4 bit thấp chuyển đổi vị trí. vd: *(a+i)=14=00001110 =>*(x+i) =00000111=7

b) Tạo mảng 1 chiều int *x; mà *(x+i) cũng chính là phần tử *(s+i) của chuỗi char*s, nhưng có giá trị được tính trong hệ thập lục phân là ½ byte cao bằng giá trị đảo bit của ½ byte thấp, còn ½ byte thấp bằng ½ byte cao(dùng phép toán thao tác bit)

vd: *(s+i)='5'=0x35=> *(x+i)=10100011 =0xA3 …

c) Viết hàm sắp xếp mảng theo phương pháp Bubble sort, để sắp xếp mảng char *x theo thứ tự giảm dần.

c) Hãy chèn mảng char*a vào mảng char*x, sao cho mảng char*x cũng có thứ tự giảm dần.

d) Hãy đổi char*x ra char*oct(mảng hệ bát phân).

gợi ý: +hiệu quả nhất là chuyển mảng x về mảng số nhị phân char*b; rồi đổi mảng b về mảng s8.

gợi ý: dùng lệnh void*memmove(void *dest, const void*src,size_t n);chép n byte từ vùng nhớ được trỏ bởi *src, đến vùng nhớ được trỏ bởi *dest, trả về 1 con trỏ trỏ đến dest.

+ so sánh *(a+i) với các phần tử *(x+i) từ đầu mảng(i=0), đến vị trí i mà *(a+i)>*(x+i), thì chuyển (n-i)*sizeof(kiểu) byte, về phía cuối mảng 1 vị trí (lệnh memmove (x+i+1,x+i,(*n-i)*sizeof(char))) , rồi chèn *(a+i) vào vị trí i, tăng n lên 1, tương tự như vậy với các phần tử *(a+i)

---

2) lập trình ASSEMBLER:

Viết một chương trình, có thủ tục NHAP (*thủ tục cho phép xóa chữ số gõ nhầm, bằng cách ấn phím Backspace, thủ tục không cho hiển thị ký tự khác ký tự thập phân lên màn hình, đồng thời thủ tục cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị -32768 ⇔32767),* **gọi thủ tục NHAP để nhập 2 ma trận A[5][5] và B[5][5] kiểu DW.**

Tính ma trận tổng C[5][5]= A[5][5] +B[5]. Tìm mảng 1 chiều x[5], mà x[i] là tổng các phần tử đứng sau phần tử lẻ đầu tiên hàng i, nếu không x[i] là phần tử giữa hàng i. In kết quả

*Thuật toán: - ma trận A gồm **nxm** phần tử*

- độ lớn kiểu dữ liệu DB(1 byte), DW(2 byte), DD(4 byte),..
Bố trí trong bộ nhớ hết hàng 0, đến hàng 1,…đến hàng n-1(trật tự hàng)
hoặc hết cột 0, đến cột 1,….,đến cột m-1(trật tự cột)
- Tìm địa chỉ A[i][j], phải tìm hàng i và phần tử thứ j trên hàng i
- (mỗi hàng có m phần tử)
*'hàng 0: vị trí bắt đầu là A , hang 1 có vị trí bắt đầu là A+m*2.(độ lớn kiểu DW là 2)*
*hàng 2: vị trí bắt đầu là A+2*n*2,… , hang i có vị trí bắt đầu là A+i*m*2.*
*-phần tử thứ j trên 1 hàng cách vị trí đầu hàng j*2 byte*
*- như vậy phần tử A[i][j] có địa chỉ là: A+(i*m+j)*độ lớn kiểu(trật tự hàng)*
*tương tự lưu trữ theo trật tự cột là: A+(i+j*n)*độ lớn kiểu*

==========

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án: LẬP TRÌNH HỆ THỐNG
Đề 15
****

1) lập trình C:

Viết hàm void* NHAP(), để nhập số nguyên kiểu char(***hàm cho phép xóa chữ số gõ nhầm, không cho hiển thị ký tự khác ký tự thập phân, đồng thời hàm cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị 0x30 ⇔0x31***), gọi hàm NHAP(), để nhập 2 chuỗi ký tự nhị phân dài 31 byte(m1 và m2), thực hiện m3= m1-m2. (char* m3;dài

32byte, byte đầu tiên chứa dấu, nếu m1>m2 thì byte đầu chứa 0, nếu không byte đầu chứa '-') , hãy đổi 16 byte thấp của m3 ra số hex N kiểu int. In kết quả.

gợi ý:có thể chuyển phép trừ về phép cọng, bằng cách giữ nguyên số bị trừ cọng với số bù 2 của số trừ.(phép cọng xem gợi ý ở đề bài số 3)

+ m1>m2:kết quả của phép cọng, chính là kết quả của phép trừ

+ m1<m2:kết quả của phép cọng,được thực hiện bù 2, rồi đặt dấu âm vào

---

2) lập trình ASSEMBLER:

Viết 1 chương trình .ASM, để thực hiện các công việc sau:

Viết thủ tục NHAP(*để nhập N kiểu DW xóa được khi gõ nhầm chữ số,bằng cách ấn phím Backspace,không cho hiển thị ký tự khác ký tự thập phân và xét điều kiện N có 3 chữ số thập phân 100 ⇔999,nếu N nằm ngoài miền trị nầy, thủ tục cho phép nhập lại***), gọi thủ tục là NHAP, để nhập mảng 1 chiều A[15] gồm 15 phần tử kiểu DW ở trên. Tính S tổng các phần tử nào của mảng A[15] là số keprker(***vd: 495=954- 459***), nếu không S lá tổng các phần tử đầu, cuối và giữa mảng, hãy đổi S ra mã ASCII số hex để lưu vào vùng nhớ bắt đầu từ CS:2AB0h**.

Thuật toán: N là số keprker:.

-lấy từng chữ số thập phân của N cất vào mảng b[3]

- tính trị thập phân X, của mảng b[3] theo thứ tụ giảm dần, đồng thời cất phần tử của b[3]vào stack.

- tính trị thập phânY,của mảng b[3] theo thứ tụ tăngdần(chỉ việc lấy các phần tử từ stack ra là có thứ tự tăng dần).

- nếu X=Y =>N là số keprker

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án: LẬP TRÌNH HỆ THỐNG

Đề 16

****

1) lập trình C:

Viết 1 chương trình, C thực hiện các công việc sau:

1)Viết hàm **void* NHAP(),** để nhập số nguyên kiểu int ***(nhập số thập tam phân, hàm cho phép xóa dấu âm hoặc chữ số thập tam phân(0⇔C(c)) khi gõ nhầm, bằng cách ấn phím Backspace, hàm không cho hiển thị ký tự khác ký tự thập tam phân lên màn hình, đồng thời hàm cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị -11B68(-32768) ⇔11b67(32767).(dùng <dos.h>, union REGS và software interrupt 0x21, 0x10)***

Gọi hàm NHAP(), để nhập 2 ma trận A, B được khai báo dưới dạng kiểu struct. Hãy viết chương trình thực hiện các bài toán trên ma trận như : nhập ,xuất ,cọng , trừ, nhân, chia,..

vd : typedef struct{ int hang ;

int cot ;

float ptu[0][0] ; } matran ;

---

2) lập trình ASSEMBLER:

Viết 1 chương trình .ASM, để thực hiện các công việc sau:

Viết thủ tục NHAP(*thủ tục cho phép xóa chữ số gõ nhầm, bằng cách ấn phím Backspace, thủ tục không cho hiển thị ký tự khác ký tự thập phân lên màn hình, đồng thời thủ tục cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị số packed BCD 00h ⇔99h),* **gọi thủ tục NHAP, để nhập 2 mảng 1 chiều A[25], B[20]:lưu số BCD nén, tính C[26]= A[25]+B[20]. Hãy đổi 2 byte thấp nhất của C[26] ra số thập phân tương ứng N kiểu DW. In C[26] và N.**

Thuật toán:hiệu chỉnh thập phân sau khi cọng 2 số BCD nén:

1) if ((AL AND 0Fh>9)OR(AF=1))

. AL=AL+6

.AF=1

2) if((AL >9Fh)OR (CF=1))

.AL=AL+60h

.CF=1

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án: LẬP TRÌNH HỆ THỐNG
Đề 17
****

1) lập trình C:

Viết 1 chương trình C thực hiện các công việc sau:

1)Viết các hàm mô phỏng các hàm chuẩn của string.h: strlen(), strcmp(), strstr(), strcat().

2)Viết các hàm để xử lý 1 bài toán nào đó, có gọi các hàm mô phỏng để thử nghiệm.

3) Cho trước 1 chuỗi **char *s =" dai hoc da nang ";** hãy tạo ra chuỗi ký tự chuẩn.(giữa các từ chỉ còn 1 ký tự trống)

4) Viết hoa 1 ký tự đầu từ , các ký tự phía sau là chữ in thường. Hãy chèn thêm chuỗi **" Truong**" ở đầu chuỗi sao cho chuỗi s vẫn là chuỗi ký tự chuẩn, rồi đếm số từ của chuỗi s.

5) Viết hàm căn lề phải.

vd: viết hàm mô phỏng hàm strlen() và thử nghiệm

```
int len( char *s)
{ int i=0; for(; *s; i++); return i;}// hoặc{ const char *st=s; while(*st)++st; return(st-s);}
 char *ch="Bach_Khoa";  printf ("Do dai that =%d=%d\n",strlen(ch),len(ch));
```

---

2) lập trình ASSEMBLER:

Viết 1 chương trình .ASM, để thực hiện các công việc sau:

Viết thủ tục NHAP(*thủ tục cho phép xóa chữ số gõ nhầm, bằng cách ấn phím Backspace, thủ tục không cho hiển thị ký tự khác ký tự thập phân lên màn hình, đồng thời thủ tục cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị 10⇔250)***, gọi thủ tục NHAP để nhập N và M (kiểu DB). Hãy tính X=N*M bằng phương pháp dịch và cọng. rôi xét X có phải là số nguyên tố?(***X là số nguyên tố khi X không chia hết lần lượt các giá trị từ 2,3,4,...x/2)***thì in X ra số thập phân và dòng thông báo:"SO NGUYÊN_TỐ". Xét tiếp X có phải số MERSEN ?, nếu không phải số MERSEN thì đổi X ra mảng số bát phân (A DB 6 DUP (0)).**

*gợi ý***:**

+ Đổi X ra số nhị phân ,nếu chỉ gồm toàn bit 1, thì đếm số bit 1 là K
*+ kiểm tra nếu X,K đều là số nguyên tố và $X=2^K-1$ =>X là số MERSEN***.**

gợi ý: đặt KQ=0. Lặp lại xét bit LSB của M,nếu là bit 1: thực hiện KQ+=N; N<<=1;M>=1;nếu không(LSB là bit 0): thực hiện N<<=1;M>>=1;.cho đến khi M==0 **).**

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án: LẬP TRÌNH HỆ THỐNG
Đề 18
****

1) lập trình C:

Viết 1 chương trình C để thực hiện các công việc sau:

1. Hãy đọc tệp nhị phân "MATRIX.BIN" mà mẫu tin đầu tiên ghi n là số hàng/cột của 1 ma trận vuông , tiếp theo là n*n mẫu tin ghi n*n phần tử ma trận int***a** với (5<=**n** <=12).Hãy taọ mảng 1 chiều **int *x** mà *(x+i) là tổng các phần tử nào trên cột i có ½ byte tận cùng bên trái là 1 chữ số BCD nén, nếu không thì chuyển đổi bit tận cùng bên trái với bit tận cùng bên phải của phần tử nầy, rồi cọng vào *(x+i).

2. Tính **t** là tổng các phần tử nào của hàng giữa ma trận int*a; là số nguyên tố**?N**ếu không(hàng giữa không có số nguyên tố) thì tính **t** là tổng tất cả các phần tử đứng sau phần tử lẻ đầu tiên trên cột giữa ma trận a

3.Tính s là tổng các phần tử của mảng x là số đối xứng, nếu không s là tổng các phần tử nào của mảng x, có chữ số hex tận cùng bên trái >=8.

4) Xét xem t và s có phải 2 số hữu nghị? Nếu số nầy bằng tổng các ước số thật sự của số kia.

5)Xem xét t có phải là số Mersen?

---

gợi ý: *+ t là số Mersen, nếu $t=2^s-1$, đồng thời cả t và s đều là 2 số nguyên tố*

+t là số nguyên tố, khi t không chia hết lần lượt từ 2,3,…sqrt(n), có nghĩa là nó chỉ chia hết cho 1 và chính nó.

+ s là số hoàn hảo, khi s bằng tổng các ước số tự nhiên của nó, không kể nó vd: 28=1+2+4+7+14 =>56 và 28 là 2 số hữu nghị 56=1+2+4+7+14+28

---

2) lập trình ASSEMBLER:

Viết một chương trình .ASM, có thủ tục NHAP để nhập N kiểu DB(*thủ tục cho phép xóa chữ số gõ nhầm, bằng cách ấn phím Backspace, thủ tục không cho hiển thị ký tự khác ký tự thập phân lên màn hình, đồng thời thủ tục cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị số packed BCD 00h⇔99h),***gọi thủ tục NHAP, để nhập 2 mảng 1 chiều A[10], B[10] gồm 10 phần tử kiểu DB, chứa 20 chữ số BCD nén, tính C[11]= A[10]-B[10]. Hãy đổi 2 byte thấp nhất ra số thập phân tương ứng N kiểu DW? in kết quả C[11] và N .**

Thuật toán:

1) xét dấu: so sánh nội dung 2 ô nhớ đầu mảng A[1]với B[1]:

-nếu lớn hơn thì thực hiện A10[]- B[10], hiệu chỉnh thập phân, lưu kết quả vào C[11], tiếp tục từ phải sang trái,....nạp 00h vào C[1](kết quả dương).

-nếu nhỏ hơn thì thực hiện B[10]- A[10], hiệu chỉnh thập phân, lưu kết quả vào C[11], tiếp tục từ phải sang trái,...nạp '-' vào c[1](kết quả âm)

-nếu bằng nhau, thì so sánh nội dung 2 phần tử tiếp theo A[2] với B[2](kết quả như trên nhưng chỉ xử lý với n-1 cặp phần tử còn lại...

2)Thuật toán hiệu chỉnh thập phân sau khi trừ 2 số BCD kông nén

if((AL AND 0Fh>9)OR(AF=1))

+ AL-=6

+AF=1=CF

+AL=AL AND 0Fh

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án: LẬP TRÌNH HỆ THỐNG

Đề 19

****

1) lập trình C:

Viết 1 chương trình C thực hiện các công việc sau:

1)Viết các hàm mô phỏng các hàm chuẩn của string.h: strlen(), strstr(), strchr(), strrchr(),hãy thử nghiệm các hàm nầy.

2) Cho trước 1 chuỗi ký tự **char *s =" -2 4,8'';** hãy thực hiện nén chuỗi, để tạo chuỗi ký tự thập phân(xóa các dấu trống và dấu phẩy) char *s="-248"

3) Đổi chuỗi xâu s ra số nguyên long n=-248;

4) Đổi long n ra mảng số Hex int *a={15,15,15,15,15,15,0,8}; rồi đổi mảng a ra lại chuỗi số hex(char*s16).

5)Hãy đổi char*s16 ra chuỗi nhị phân bù 2.

In kết quả.

vd: viết hàm mô phỏng hàm strlen() và thử nghiệm

```
int len( char *s)
{ int i=0; for(; *s; i++); return i;}// hoặc{ const char *st=s; while(*st)++st; return(st-s);}
char *ch="Bach_Khoa";  printf ("Do dai that =%d=%d\n",strlen(ch),len(ch));
```

2) <u>lập trình ASSEMBLER</u>:

Viết 1 chương trình .ASM, để thực hiện các công việc sau:

Viết thủ tục NHAP(*thủ tục cho phép xóa chữ số gõ nhầm, bằng cách ấn phím Backspace, thủ tục không cho hiển thị ký tự khác ký tự thập phân lên màn hình, đồng thời thủ tục cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị -32768⇔32767),* **gọi thủ tục NHAP để nhập ma trận A[5][6] , B[6][7]kiểu DW.Tính ma trận tích C[5][7]= A[5][6]*B[6][7].**

In kết quả

<u>Thuật toán</u>: *- ma trận A gồm **nxm** phần tử*

- độ lớn kiểu dữ liệu DB(1 byte), DW(2 byte), DD(4 byte),..

Bố trí trong bộ nhớ hết hàng 0, đến hàng 1,…đến hàng n-1(trật tự hàng)

hoặc hết cột 0, đến cột 1,….,đến cột m-1(trật tự cột)

- Tìm địa chỉ A[i][j], phải tìm hàng i và phần tử thứ j trên hàng i

- (mỗi hàng có m phần tử)

*'hàng 0: vị trí bắt đầu là A , hang 1 có vị trí bắt đầu là A+m*2.(độ lớn kiểu DW là 2)*

*hàng 2: vị trí bắt đầu là A+2*n*2,… , hang i có vị trí bắt đầu là A+i*m*2.*

*-phần tử thứ j trên 1 hàng cách vị trí đầu hàng j*2 byte*

*- như vậy phần tử A[i][j] có địa chỉ là: A+(i*m+j)*độ lớn kiểu(trật tự hàng)*

*tương tự lưu trữ theo trật tự cột là: A+(i+j*n)*độ lớn kiểu*

==========

KHOA CÔNG NGHỆ THÔNG TIN

<u>Đồ án</u>: LẬP TRÌNH HỆ THỐNG

<u>Đề 20</u>

****

1) <u>lập trình C</u>:

Viết <u>1 chương trình</u> C để thực hiện các công việc sau:

a)Viết hàm **void* NHAP(),** để nhập số nguyên kiểu int ***(nhập số thập lục phân, hàm cho phép xóa dấu âm hoặc chữ số thập lục phân(0⇔F(f)) khi gõ nhầm, bằng cách ấn phím Backspace, hàm không cho hiển thị ký tự khác ký tự thập lục phân lên màn hình, đồng thời hàm cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị -0x8000(-32768) ⇔0x7fff(32767).(dùng <dos.h>, union REGS và software interrupt 0x21, 0x10)***gọi hàm NHAP(), để nhập ma trận vuông **int *a;**(gồm n*n phần tử: 7<=n<=12).

b)Tạo ma trận unsigned char *b ; bằng cách hoán đổi cột/hàng thứ 0 với cột/hàng thứ n-1, cột/ hàng thứ 1 với cột/hàng thứ n-2…

c)Tạo ma trận x là ma trận hoán vị của b.

d)Tính c là tổng các phần tử nằm phía trên đường chéo chính ma trận a và d là tổng các phần tử nằm phía dưới đường chéo phụ ma trận x. Tính w=(float)c/d ;

e) đổi w ra chuỗi số thực char *th ; đổi phần nguyên và phần lẻ của w ra chuỗi ký tự thập phân, và gán ký tự '.' ở giữa 2 chuỗi.

khi đổi phần lẻ, chú ý phần lẻ tuần hoàn, nên phải có điều kiện số chữ số phần lẻ cần lấy <u>vd</u> : 0.6 => char *ple =" 100110011001…. "

---

2) <u>lập trình ASSEMBLER</u>:

Viết 1 chương trình .ASM, để thực hiện các công việc sau:

Viết thủ tục NHAP để nhập N kiểu DW*(thủ tục cho phép xóa chữ số gõ nhầm, bằng cách ấn phím Backspace, thủ tục không cho hiển thị ký tự khác ký tự thập phân lên màn hình, đồng thời thủ tục cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị -32768⇔32767),* **đổi N ra chuỗi số thập lục phân(HEX DB 4 dup (0)), hãy tìm ký tự 'A' có trong chuỗi HEX. In N, chuỗi HEX và kết quả tìm kiếm.**

<u>Xử lý chuỗi</u>:

+ CLD(DF=0): xử lý chuỗi theo chiều tăng

STD(DF=1): xử lý chuỗi theo chiều giảm

+ CX chứa số phần tử cần truy xuất

+ Nạp địa chỉ của chuỗi nguồn vào DS:SI

Nạp địa chỉ của chuỗi đích vào ES:DI

==========

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án: LẬP TRÌNH HỆ THỐNG
Đề 21
****

1) lập trình C:

Viết hàm **void* NHAP(),** để nhập ký tự chữ cái in thường kiểu char***(hàm cho phép xóa chữ cái gõ nhầm,hàm không cho hiển thị ký tự khác ký tự chữ cái in thường, đồng thời hàm cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị 'a'⇔'z'),***gọi hàm NHAP(), để nhập họ và tên sinh viên, để quản lý hồ sơ sinh viên, tổ hợp dữ liệu theo danh sách liên kết, với mỗi sinh viên cần quản lý các thông tin như: họ và tên, năm sinh, điểm của 6 môn thi trong học kỳ, điểm trung bình, bao gồm các công việc sau:

a) hàm void tao_lap(), để nhập dữ liệu của sinh viên vào danh sách.
b) hàm void bo_sung(), để bổ sung dữ liệu của sinh viên mới vào danh sách.
c) hàm void loai_bo(), để loại bỏ sinh viên khỏi danh sách
d) hàm void tim_kiem(), để tìm kiếm sinh viên theo họ tên
e) hàm void sap_xep(), để sắp xếp danh sách theo thứ tự giảm dần điểm trung bình.
In kết quả

vi dụ: ***struct svien***
{ char hten[35];
int nsinh;
float toan,ly,ltht,do_an,thdc,ctmt,dtb;
struct svien*link;
};
typedef svien *ctro;
int n;// số sinh viên
ctro fd; // con trỏ đầu danh sách, cần khởi tạo NULL

vidu: sắp xếp theo điểm trung bình , khai báo ctro p,q; for(p=fd;p->link!=NULL;p=p->link) for(q=p->link; q!=NULL;q=q->link) if(p->dtb<q->dtb)thì hoán đổi 2 bản ghi cho nhau.

---

2) lập trình ASSEMBLER:

Viết 1 chương trình .ASM, để thực hiện các công việc sau:

Viết thủ tục NHAP, để nhập số nguyên kiểu DB(*thủ tục cho phép xóa chữ số gõ nhầm, không cho hiển thị ký tự khác ký tự thập phân, đồng thời thủ tục cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị 0⇔1*)**, gọi thủ tục NHAP, để nhập mảng m1 DB 31 dup(0) và m2 DB 25 dup(0)(mỗi byte chứa 1 bit), rồi thực hiện m3= m1-m2.(m3 DB 32 dup(0)), hãy đổi 16 byte thấp của m3 ra số hex N kiểu DW.In kết quả M3 và N.**

==========

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án: LẬP TRÌNH HỆ THỐNG
Đề 22
****

1) lập trình C:

Viết 1 chương trình C để thực hiện các công việc sau:

1. Viết hàm la_ma(), để nhập 1 số la mã, dưới dạng chuỗi ký tự, trả về số nguyên tương ứng(int x;)

2.Tạo mảng 1 chiều int *a; mà *(a+i) chính là chữ số oct(bát phân)thứ i của x.

vd: int x=142=0216=>*(a+0)=2, *(a+1)=1 ,*(a+2)=6 , với (n=3)

3.Tìm xem trong int*a có đoạn tăng nào có tổng giá trị là lớn nhất.

4) Hãy duyệt mảng int*a từ trái sang phải, nếu *(a+i) lẻ thì xóa 1 phần tử bên phải nó, sau khi xóa không xét phần tử *(a+i) đó nữa.

5)Hãy tìm kiếm 5 có trong int*a.(tìm kiếm nhị phân).

gợi ý: M:1000 X=10 ký hiệu biểu diễn 10^x không được đứng ngay trước 10^{x+1}
D=500 V=5 XCIX=99 (IC : sai)
C=100 I=1 L=50

---

2) <u>lập trình ASSEMBLER</u>:

Viết 1 chương trình .ASM, để thực hiện các công việc sau:

Viết thủ tục NHAP(*thủ tục cho phép xóa chữ số gõ nhầm, bằng cách ấn phím Backspace, thủ tục không cho hiển thị ký tự khác ký tự thập phân lên màn hình, đồng thời thủ tục cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị -32768⇔32767),* **gọi thủ tục NHAP để nhập ma trận A[5][5] kiểu DW.Tìm ma trận chuyển vị của A[5][5]. Hảy chuyển phần tử lớn nhất của từng cột ma trận A lên đường chéo phụ. In kết quả**

*<u>Thuật toán</u>: - ma trận A gồm **nxm** phần tử*

- độ lớn kiểu dữ liệu DB(1 byte), DW(2 byte), DD(4 byte),..

Bố trí trong bộ nhớ hết hàng 0, đến hàng 1,…đến hàng n-1(trật tự hàng)

hoặc hết cột 0, đến cột 1,….,đến cột m-1(trật tự cột)

- Tìm địa chỉ A[i][j], phải tìm hàng i và phần tử thứ j trên hàng i

- (mỗi hàng có m phần tử)

*'hàng 0: vị trí bắt đầu là A , hang 1 có vị trí bắt đầu là A+m*2.(độ lớn kiểu DW là 2)*

*hàng 2: vị trí bắt đầu là A+2*n*2,… , hang i có vị trí bắt đầu là A+i*m*2.*

*-phần tử thứ j trên 1 hàng cách vị trí đầu hàng j*2 byte*

*- như vậy phần tử A[i][j] có địa chỉ là: A+(i*m+j)*độ lớn kiểu(trật tự hàng)*

*tương tự lưu trữ theo trật tự cột là: A+(i+j*n)*độ lớn kiểu*

==========

KHOA CÔNG NGHỆ THÔNG TIN

<u>Đồ án</u>: LẬP TRÌNH HỆ THỐNG

<u>Đề 23</u>

****

1) <u>lập trình C</u>:

Hãy viết 1 chương trình C để xử lý các công việc sau đây:

1) Viết hàm **void* NHAP(),** để nhập số nguyên kiểu long có 9 chữ số (*hàm cho phép xóa chữ số gõ nhầm,hàm không cho hiển thị ký tự khác ký tự thập phân, đồng thời hàm cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị 100000000 ⇔999999999*),gọi hàm NHAP(),để nhập số **n** kiểu long, rồi đổi **n** ra chuỗi ký tự thập phân (TP) char ***s** ;

2) Hãy sắp xếp giảm dần các phần tử trong chuỗi **s(** phương pháp **Quicksort)**.Tính **nt** là tổng các trị chữ số TP của ký tự thập phân nào có 2 bit ở giữa là 10, nếu không thì chuyển đổi vị trí 3 bit cuối cho nhau, rồi cọng vào nt.

<u>vd</u> : *(s+i)='5' => trị chữ số TP là 5(0101), đúng 2 bit giữa là 10 (cọng 5 vào nt)

****(s+i)='6'=>trị chữ số TP là 6(0110), hai bit giữa không phải là 10. phải***

chuyển đổi vị trí 3 bit cuối 0110 =>0011(cọng 3 vào nt)

3) Xét xem **nt (**ở câu **2)** có phải là phần tử Fibonaci(Fi) thứ k=n%10.

cho trước k=0 hoặc k=1=> phần tử Fi= 1, nếu không Fi(k)=Fi(k-1)+Fi(k-2)

4) Tạo mảng 1 chiều int *x; mà *(x+i) cũng chính là phần tử *(s+i) của chuỗi char *s, nhưng có giá trị được tính trong hệ thập lục phân là ½ byte cao bằng giá trị đảo bit của ½ byte thấp, còn ½ byte thấp bằng ½ byte cao(dùng phép toán thao tác bit)

<u>vd</u>: *(s+i)='5'=0x35=> *(x+i)=10100011 =0xA3 …

5)Xét **int nt;** có phải số keprker?

<u>vd</u>: nt=495 =954- 459 là số keprker.

---

2) <u>lập trình ASSEMBLER</u> :

Viết một chương trình, có thủ tục NHAP (*để nhập N kiểu DB xóa được khi gõ nhầm chữ số, không hiển thị ký tự khác ký tự thập phân và xét điều kiện N chứa 2 chữ số packed BCD 00h ⇔99h,nếu gỏ miền trị nằm ngoài miền trị nầy thì thủ tục cho phép gỏ lại***) ,gọi thủ tục NHAP, để nhập 2 mảng 1 chiều A[20], B[20] chứa 40 chữ số BCD nén, tính C[21]= A[20]-B[20]. Hãy đổi 2 byte thấp nhất của C[21] ra số thập phân tương ứng, lưu trong biến N kiểu DW.In C[21] và N.**

==========

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án: LẬP TRÌNH HỆ THỐNG
Đề 24

****

1) lập trình C:

Viết 1 chương trình C, thực hiện các công việc sau:

1) Viết hàm **void* NHAP(),** để nhập số nguyên kiểu int(***hàm cho phép xóa chữ số gõ nhầm,hàm không cho hiển thị ký tự khác ký tự thập phân, đồng thời hàm cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị 0⇔9*)**, gọi hàm void *NHAP() để nhập mảng 1 chiều

int *a={2,2,2,1,1,4,7,7,7,4,4,8,3,0,0,3,3,6,2,2,6,6,6};

2) hãy xỏa các phần tử giống nhau chỉ để lại 1phần tử, rồi xóa tiếp phần tử nào bằng 0 => **int *a={2,1,4,7,4,8,3,6,2,6};**

3) chuyển các phần tử chẵn mảng a sang phải, các phần tử lẻ sang trái, rồi tạo chuỗi char *st; mà *(st+i) là mã ASCII của chữ số thập phân thứ i mảng a.

4)đổi chuỗi st ra lại số thập phân **long n**. Tìm **int x** là tổng các ký tự nào của chuỗi char*st là số nguyên tố.

5).Xét **int x;** có phải số Amstrong? ***vd: x=153=13+53+33 là số Amstrong: tổng các lũy thừa bậc n của các chữ số của nó bằng chính nó.***

gợi ý: + để đổi 1 dãy số về trị thập phân (int d ;) dùng phương pháp Horner

d=(…((*(a+0)*10+*(a+1))*10+*(a+2))*10+…+*(a+k-1))*10+*(a+k) *(a+i) là phần tử thứ i + đổi 1 chuỗi số thập phân(tương tự)chú ý đổi ký tự thập phân ra chữ số thập phân.

---

2) lập trình ASSEMBLER:

Viết một chương trình, có thủ tục NHAP (*để nhập N kiểu DB xóa được khi gõ nhầm chữ số, không cho hiển thị ký tự khác ký tự thập phân và xét điều kiện N chứa 1 chữ số unpacked BCD 00h⇔09h*) **,gọi thủ tục NHAP, để nhập 2 mảng 1 chiều A[15], B[15] gồm 15 phần tử kiểu DB, chứa 30 chữ số BCD nén, tính C[30]= A[15] *B[15].**

Gợi ý : giống khi nhân số thập phân, ta tách từng chữ số BCD từ phải sang trái nhân với số hạng thứ nhất , kết quả mỗi lần nhân dịch qua trái 1 đơn vị cho đến hết, sau đó ta cọng các kết quả đó lại.Với số BCD nén ta dùng thêm các mảng M1, M2 và M3. M1 dùng trong phép tính cọng dồn.M2 dùng để lưu kết quả tạm thời(tích tạm) của phép nhân từng chữ số BCD của B với A. M3 chứa kết quả.

==========

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án: LẬP TRÌNH HỆ THỐNG
Đề 25

****

1) lập trình C:

Viết 1 chương trình C thực hiện các công việc sau:

a) Hãy tạo tệp văn bản ”M1CINP.TXT” mà dòng đầu tiên ghi n (15<=n<=35)là số phần tử mảng 1 chiều int ***x;** tiếp theo là n dòng các phần tử của mảng.

b) Đọc tệp văn bản ”M1CINP.TXT” , lưu dòng đầu tiên của tệp vào int n; tiếp theo là n phần tử của mảng 1 chiều lưu vào int*x; Tính **s** là tổng các phần tử nào của mảng **x** có chữ số thập phân tận cùng bên trái> chữ số tận cùng bên phải, nếu đúng thì hoán chuyển 2 chữ số thập phân nầy, trước khi cọng vào tổng s.

vd: *(x+i)=5062 =>chuyển đổi 5⇔2 =>*(x+i)=2065

c) Tạo mảng 1 chiều int ***y;** mà *(**y**+i) cũng chính là phần tử *(**x**+i) , nhưng giá trị trong hệ đếm thập lục phân được viết đảo các chử số lại.

vd: *(**x**+i)=0x6DA =>*(y+i)=0xAD6.

d) Hãy sắp xếp mảng **y** theo thứ tự tăng dần(phương pháp shake). rồi chèn **s=s%55** cùng mảng **x** vào **y** sao cho mảng **y** vẫn có thứ tự tăng dần.

e) Hãy dịch phải xoay vòng mảng int*y (n%9+1) lần, in kết quả.

gợi ý: +dung hàm void*memmove(void *dest, const void*src,size_t n);chép n byte từ vùng nhớ được trỏ bởi *src, đến vùng nhớ được trỏ bởi *dest, trả về 1 con trỏ trỏ đến dest.

+ so sánh s với các phần tử *(y+i) từ đầu mảng(i=0), đến vị trí i mà s<*(y+i), thì chuyển (n-i)*sizeof(kiểu) byte, về phía cuối mảng 1 vị trí (lệnh memmove (y+i+1,y+i,(*n-i)*sizeof(int))) , rồi chèn s vào vị trí i, tăng n lên 1, tương tự như vậy với các phần tử *(x+i)

---

2) <u>lập trình ASSEMBLER</u>:

Viết 1 chương trình .ASM, để thực hiện các công việc sau:

Viết thủ tục NHAP(*để nhập N kiểu DW:nhập số thập phân có dấu xóa được khi gõ nhầm chữ số* **), đổi N ra chuỗi số bát phân(OCT DB 4 dup (0)), hãy tìm ký tự '5' có trong chuỗi OCT In N, chuỗi OCTvà kết quả tìm kiếm.**

<u>Xử lý chuỗi</u>:

+ *CLD(DF=0): xử lý chuỗi theo chiều tăng*
  STD(DF=1): xử lý chuỗi theo chiều giảm
+ *CX chứa số phần tử cần truy xuất*
+ *Nạp địa chỉ của chuỗi nguồn vào DS:SI*
  Nạp địa chỉ của chuỗi đích vào ES:DI

==========

KHOA CÔNG NGHỆ THÔNG TIN

<u>Đồ án</u>: LẬP TRÌNH HỆ THỐNG

<u>Đề 26</u>

****

1) <u>lập trình C</u>:

Viết một chương trình C để thực hiện các công việc sau:

Viết hàm **int NHAP()**, để nhập số nguyên kiểu unsigned char ***(0 ⇔ 255) ,nếu nhập 1 số nằm ngoài miền trị nầy thì hàm cho phép nhập lại, hàm cho phép xóa khi gõ nhầm ký tự thập phân, không cho hiển thị các ký tự khác ký tự thập phân lên màn hình).***

gọi hàm NHAP() để nhập số nguyên, rồi chèn vào danh sách liên kết đơn (có thể dùng cách chèn phần tử đầu, chèn phần tử cuối, hoặc chèn đệ quy)

```
struct SO{      int  data;
                struct SO *link;
          };
  typedef struct SO* ctro;
```

a) sắp xếp tăng dần các phần tử trong danh sách

b) chèn 1 phần tử mới vào danh sách, sao cho danh sách vẫn có thứ tự tăng dần.

c) chuyển đổi các phần tử chứa trị lẽ lên đầu danh sách, các phần tử chứa trị chẵn vào cuối danh sách.

d) Hãy xóa các phần tử chứa trị <=9.

e) Tìm 1 phần tử trong danh sách tại 1 vị trí bất kỳ.

<u>gợi ý</u>: nhập đệ quy, không dùng hàm chèn node, không cần khởi tạo NULL cho con trỏ đầu danh sách

```
ctro dq()
{ it x; ctro p;
   x=nhap(); if(!x)return NULL;
   p=(ctro) malloc(sizeof( struct SO));
    p->data=x;
    p->link=dq(); return p;
 }
```

---

2) <u>lập trình ASSEMBLER</u>:

Viết 1 chương trình .ASM, để thực hiện các công việc sau:

Viết thủ tục NHAP (*để nhập N kiểu DW xóa được khi gõ nhầm chữ số, không hiển thị ký tự khác ký tự thập phân và xét điều kiện nếu gỏ miền trị nằm ngoài 0<=>65535 thì thủ tục cho phép gỏ lại*),**gọi thủ tục NHAP để nhập ma trận A[5][5] kiểu DW. Tìm ma trận chuyễn vị của A[5][5] và tìm mảng 1 chiều x[5], mà x[i] là phần tử lớn nhất - nhỏ nhất cột i. In kết quả**

*<u>Thuật toán</u>: - ma trận A gồm **nxm** phần tử*

- độ lớn kiểu dữ liệu DB(1 byte), DW(2 byte), DD(4 byte),..
Bố trí trong bộ nhớ hết hàng 0, đến hàng 1,…đến hàng n-1(trật tự hàng)
hoặc hết cột 0, đến cột 1,….,đến cột m-1(trật tự cột)

- *Tìm địa chỉ A[i][j], phải tìm hàng i và phần tử thứ j trên hàng i*
- *(mỗi hàng có m phần tử)*

*'hàng 0: vị trí bắt đầu là A , hang 1 có vị trí bắt đầu là A+m*2.(độ lớn kiểu DW là 2)*
*hàng 2: vị trí bắt đầu là A+2*n*2,… , hang i có vị trí bắt đầu là A+i*m*2.*

*-phần tử thứ j trên 1 hàng cách vị trí đầu hàng j*2 byte*

*- như vậy phần tử A[i][j] có địa chỉ là: A+(i*m+j)*độ lớn kiểu(trật tự hàng)*
*tương tự lưu trữ theo trật tự cột là: A+(i+j*n)*độ lớn kiểu*

==========

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án: LẬP TRÌNH HỆ THỐNG
Đề 27
****

1) lập trình C:

Viết 1 chương trình C thực hiện các công việc sau:

Hãy đọc tệp nhị phân "M1CINP.BIN" mà mẫu tin đầu tiên ghi n phần tử mảng 1 chiều int ***x ;** tiếp theo là n mẫu tin các phần tử của mảng.

a) Tạo 1 ma trận xoắn **int *a** , được hình thành từ các phần tử mảng x, được xoắn theo thứ tự giảm dần từ ngoài vào trong .

vd**:** **22 21 20**
7 - 6 19
9 12 17

b) Tạo mảng 1 chiều int*w ; w**[i]**= maxi + maxj , với maxi là phần tử lớn nhất hàng i, maxj là phần tử lớn nhất cột j(cột j ứng với phần tử nhỏ nhất cuối cùng của hàng i.

c) Hãy tạo các mảng 1 chiều int ***y;** mà *(**y**+i) cũng chính là phần tử *(**x**+i) , nhưng giá trị trong hệ đếm thập phân được viết đảo các chử số lại.

d) Hãy trộn mảng int*y vào mảng int*x, sao cho các phần tử của mảng int*y xen kẻ với các phần tử của mảng int*x ; In kết quả (dùng ngắt).

vd:*(x+i)=129 => *(y+i)=921

(y+i)=0 ; do{ *(y+i)*=10+*(x+i)%10 ;}while(*(x+i)/=10) ;

e)Trên mảng int*x, hãy sắp xếp sao cho các vị trí chẵn trên mảng vẫn chứa số chẵn nhưng có thứ tự tăng, các vị trí lẻ trên mảng vẫn chứa số lẻ nhưng có thứ tự giảm

---

2) lập trình ASSEMBLER:

Viết chương trình ASM để mô phỏng lệnh DIR

==========

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án: LẬP TRÌNH HỆ THỐNG
Đề 28
****

1) lập trình C:

Viết 1 chương trình C để thực hiện các công việc sau:

a)Viết hàm **void* NHAP(),** để nhập số nguyên kiểu unsigned int ***(nhập số thập ngũ phân, hàm cho phép xóa chữ số thập ngũ phân(0⇔E(e)) khi gõ nhầm, bằng cách ấn phím Backspace, hàm không cho hiển thị ký tự khác ký tự thập ngũ phân lên màn hình, đồng thời hàm cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị 0⇔9A97(32767).(dùng <dos.h>, union REGS và software interrupt 0x21, 0x10)***, gọi hàm NHAP(), để nhập ma trận **int*a ; (6<=n <=10**),hãy xóa đi cột và hàng chứa phần tử nhỏ nhất cuối cùng của ma trận **a**.

b)Tạo mảng 1 chiều int *x, mà *(x+i) là tổng các phần tử cột i nào có byte cao lưu 2 chữ số BCD nén, nếu không *(x+i) là phần tử giữa hàng i.

c)Tạo ma trận xoắn int*c, bằng cách xoắn các phần tử của ma trận a theo thứ tự giảm dần từ ngoài vào trong.

d)Tạo ma trận zích zắc tăng dần theo cột int*d, từ các phần tử của mảng int*x.

e) kiểm tra xem tổng các số chẵn ở vị trí lẻ trên mảng int* x, có bằng tổng các số lẻ ở vị trí chẵn?

gợi ý: 1 giá trị nguyên không âm (int w;) có byte cao chứa 2 chữ số BCD nén:

int z=w; dịch phải z 8 bit, int l=z; l>>=4; z&=0x0f; if(l<=9 && z<=9) thì w có byte cao chứa 2 chữ số BCD nén

---

2)lập trình ASSEMBLER:

Viết chương trình ASM để mô phỏng lệnh TIMER

.

==========

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án: LẬP TRÌNH HỆ THỐNG

Đề 29

****

1) lập trình C:

Viết hàm **void* NHAP(),** để nhập ký tự chữ cái in thường kiểu char**(*hàm cho phép xóa chữ cái gõ nhầm,hàm không cho hiển thị ký tự khác ký tự chữ cái , đồng thời hàm cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị 'a'⇔'z' và 'A'⇔ 'Z'*),**gọi hàm NHAP(), để nhập họ và tên,địa chỉ, phòng làm việc của công chức, để quản lý hồ sơ công chức, tổ hợp dữ liệu theo danh sách liên kết, với mỗi công chức cần quản lý các thông tin như: họ và tên, địa chỉ, phòng làm việc, năm sinh, bậc lương, bao gồm các công việc sau:

a) hàm void tao_lap(), để nhập dữ liệu của công hức vào danh sách.

b) hàm void chen(), để chén dữ liệu của công chức mới vào danh sách.

c) hàm void loai_bo(), để loại bỏ công chức khỏi danh sách

d) hàm void sua_doi(), để thay đổi thông tin của cán bộ

e) hàm void sap_xep(), để sắp xếp danh sách theo thứ tự giảm dần bậc lương.

In kết quả

vi dụ: ***struct cnv***

```
{ char hten[35];
  char dchi[35];
  char  plv[20];
  int nsinh;
   float bac_luong;
   struct cnv*link;
  };
typedef  cnv *ctro;
int n;// số công nhân viên
ctro fd; // con trỏ đầu danh sách, cần khởi tạo NULL
```

vidu: sắp xếp theo bậc lương, khai báo ctro p,q; for(p=fd;p->link!=NULL;p=p->link) for(q=p->link; q!=NULL;q=q->link) if(p->bac_luong<q->bac_luong)thì hoán đổi 2 bản ghi cho nhau.

---

2) lập trình ASSEMBLER :

Viết 1 chương trình, có thủ tục NHAP, để nhập số nguyên kiểu DB(*thủ tục cho phép xóa chữ số gõ nhầm, không cho hiển thị ký tự khác ký tự thập phân, đồng thời thủ tục cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị 0⇔1*)**, gọi thủ tục NHAP, để nhập mảng m1 DB 31 dup(0) và m2 DB 31 dup(0)(mỗi byte chứa 1 bit), rồi thực hiện m3= m1-m2.(m3 DB 32 dup(0)), hãy đổi 16 byte thấp của m3 ra số bát phân N kiểu DW.In kết quả M3 và N.**

==========

Đồ án: LẬP TRÌNH HỆ THỐNG

Đề 30

****

1) lập trình C:Viết 1 chương trình C, để thực hiện các công việc sau:

1)Viết hàm **void* NHAP(),** để nhập số nguyên kiểu int ***(nhập số thập tứ phân, hàm cho phép xóa dấu âm hoặc chữ số thập tứ phân(0⇔D(d)) khi gõ nhầm, bằng cách ấn phím Backspace, hàm không cho hiển thị ký tự khác ký tự thập tứ phân lên màn hình, đồng thời hàm cho phép nhập lại, nếu giá trị nhập nằm ngoài miền trị :-Bd28 (-32768) ⇔BD27(32767).(dùng <dos.h>, union REGS và software interrupt 0x21, 0x10)***

b) gọi hàm ***NHAP()** để nhập ma trận **int *a,** gồm n x m phần tử, với (5<=n,m<=12), hãy tìm mảng 1 chiều int*x, mà *(x+i) là tổng các phần tử nào trên cột i có tổng trị vị trí của 4 bit giữa >190 , nếu cột i không có phần tử như vậy, thì chuyển đổi vị trí 4 bit giữa nầy cho nhau.

c) Tìm K là min trong các số nhỏ nhất trên hàng i(ma trận a), bắt đầu từ phần tử đầu tiên đến phần tử trên đường chéo chính.

d)Tạo ma trận int*b từ ma trận int*a sao cho b[i][j]bằng tổng các phần tử khôngthuộc dòng i, cột j ma trận a.

e) Chuyển ma trận b thành ma trận chuyển vị **bt** và in ra

(ma trận bt kích thước m x n với btji= bij được gọi là ma trận chuyển vị của ma trận b kích thước n x m)

vi dụ: K=-22 =>char *hex="0xfFeA".=>char *oct="0177752"

hiệu quả nhất để đổi chuỗi hex ra chuỗi oct là đổi qua trung gian chuỗi nhị phân char*bin="B1111111111101010".

gợi ý: trị vị trí 4 bit giữa($b_9b_8b_7b_6$) là 2^9, 2^8, 2^7. 2^6. chuyển đổi vị trí 4 bit giữa $b_9 \Leftrightarrow b_6$, $b_8 \Leftrightarrow b_7$.

---

2) lập trình ASSEMBLER:

Viết 1 chương trình .ASM, để thực hiện các công việc sau:

Viết thủ tục NHAP(*để nhập N kiểu DB xóa được khi gõ nhầm chữ số và xét điều kiện N chứa 2 chữ số packed BCD 00h⇔99h*) ,gọi thủ tục NHAP, để nhập số BCD nén vào BX và DL. Thực hiện DL-BX, hiệu chỉnh thập phân và cất kêt quả vào BX.(AL chứa 00h nếu kết quả dương,nếu không AL chứa -1 nếu kết quả âm).Hãy đổi BX ra chuỗi số hex, lưu trong chuỗi HEX DB 4 dup(0), sau đó đổi chuỗi số Hex ra chuỗi số bát phân, lưu trong chuỗi BF DB 6 dup(0). In kết quả.

1) xét dấu: so sánh nội dung 2 thanh ghi Bx và DL.

-nếu nhỏ hơn thì thực hiện DL- Bl, hiệu chỉnh thập phân, lưu kết quả v ào AL, nạp -1 vào AH(kết quả âm)

2) Thuật toán hiệu chỉnh thập phân sau khi trừ 2 số BCD nén

1)if(AL AND 0Fh>9)OR(AF=1)

+AL=AL-6

+AF=1)

if(AL>9Fh)OR(CF=1)

+AL=AL-60H

+CF=1