

RAČUNALNIŠTVO ZA ELEKTROTEHNIKE II

odgovori na primere ustnih vprašanj

Šolsko leto 2008 / 2009
Izvajalec Iztok Fajfar

Avtor dokumenta /
Sodelavci stromar.si

UREJANJE DOKUMENTA

VERZIJA 01 REVIZIJA 03
DATUM 25. 2. 2009

ZADNJI POPRAVLJAL /
PREGLEDAL /

OPOMBE

POPRAVKI

v dokumentu lahko obstajajo napake

Vse rešitve v tem dokumentu so le moj pristop k problemu, ki ni ne najboljši, niti vedno pravilen. Moje rešitve so le delček vseh mogočih rešitev! Poizkusite poiskati svoje rešitve, moje pa uporabite kot primer, kakšna rešitev je še mogoča. Ko jih boste našli, pripravite dokument podoben temu za naslednje generacije študentov.

Avtor: Lister

Za pripravo na izpit je vsekakor potrebno pridobiti praktično znanje. Nalog se ne učite na pamet, temveč jih rešite sami, saj boste le tako osvojili znanje. Dokument je zelo dober pomočnik, kadar se kje zatakne (nimate ideje, kako sploh začeti, program ne deluje pravilno itd.)

Avtorju se za posredovanje dokumenta lepo zahvaljujemo.

stromar.si

1. Napišite program, ki na prikazovalniku ves čas prikazuje vsebino registra IO0PIN. Različica problema: Prikazovalnik naj prikazuje samo stanje bitov b12 do b15, ki ustrezajo stanjem tipk T0 do T3.

```
#include "io.h"

int main(void)
{
    KEY_INIT;
    _LCDInit();
    _KeyInit();
    _setcursortype(_NOCURS0R);
    int x;

    while(1)
    {
        x = IO0PIN; // x dobi vrednost IO0PIN
        x = x & 0x0000f000; // 0 bite postavi na 0, 1 pa ohranja njihovo
        vrednost // bite zamakne za 12 v desno
        x = x >> 12; // izpiše eno cifro šestnajstiškega zapisa
        printf("%1X \r" ,x);
    }
}
```

Podani primer je rešitev različice problema, z minimalnimi spremembami opravi tudi osnovno nalogo.

2. Napišite program, ki na prikazovalnik ob vsakem pritisku na tipko T0 prikaže en pomišljaj. Tako bo na primer po treh pritiskih na tipko T0 na prikazovalniku videti tri pomišljaje (---). Različica problema: Ob vsakokratnem pritisku na tipko T1 naj zadnji od prikazanih pomišljajev izgine.

```
#include "io.h"

int main(void)
{
    _LCDInit();
    _KeyInit();
    _setcursortype(_NOCURS0R);
    char ooo[] ={'-',0}; // poskrbi za prvi pomišljaj izpisan na zaslonu
    int i = 0;
    int x;

    printf("%s", ooo); // začetni izpis
    while (1)
    {
        x = getch();
        if(x == '0'){ooo[i+1]='-'; ooo[i+2]=0; i++;} // v znakovni niz ooo zapisuje -
        if(x == '1'){if(i >= 0)i--;ooo[i+1]=0;} // v znakovnem nizu postavlja nulti znak
        clrscr();
        printf("%s", ooo);
    }
    return 0;
}
```

Tudi tukaj je rešena samo različica problema.

3. Napišite program, ki na prikazovalnik v šestnajstiškem zapisu izpisuje kode ASCII vseh znakov med 'a' in 'z'. Naenkrat naj se prikazuje le ena koda, ki naj se zamenja ob pritisku na katerokoli tipko. Kode naj se spreminjajo po vrsti, ko pa pridemo do 'z', naj bo spet na vrsti 'a'.

Različica problema: Dopolnite program tako, da se ob pritisku na tipko T0 koda poveča, ob pritisku na tipko T1 pa pomanjša za ena. Kodo ASCII lahko izpišete v šestnajstiškem zapisu z uporabo formatnega določila %x, formatno določilo za dvojiški izpis pa ne obstaja. Program dopolnite tako, da bo kode izpisoval v dvojiškem zapisu.

```
#include "io.h"

int main(void)
{
    _LCDInit();
    _KeyInit();
    _setcursortype(_NOCURSOR);
    char x; // vrednost tipke
    char z = 'a'; // z hrani vrednost trenutno prikazane črke
    printf("%X",z);
    while(1)
    {
        x= getch();
        if(x == '0')z -= 1; // spreminjanje črke
        if(x == '1')z += 1; // spreminjanje črke
        if(z < 'a' || z > 'z')z = 'a'; // omejitev črk

        clrscr();
        printf("%X",z); // šestnajstiški izpis
    }
    return 0;
}
```

Rešen je le osnovni problem, izpis ascii kode v šestnajstiškem načinu.

5. Napišite in v programu uporabite funkcijo, ki kot parameter sprejme realno vrednost in vrne najmanjše celo število, ki je še večje od podanega parametra (zaokroževanje navzgor).

Primer klica funkcije:

```
int x;
```

```
x = zaokrozi_navzgor(3.24); //x dobi vrednost 4
```

```
#include "io.h"

int zaokrozi(double stevilo); // prototip funkcije

int main(void)
{
    _LCDInit();
    _KeyInit();
    double x; // vrednost ki jo zaokrožujemo
    int y; // zaokrožena vrednost
    x=139847.923947; // vnesi zeljeno stevilo
    printf("vase stevilo %f",x); // izpis x
    getch(); // samo da zaustavi program
    clrscr(); // pobriše izpis x
    y = zaokrozi(x); // klic funkcije
    printf("%d",y); // izpis y
    while(1) // zaključna neskončna zanka
    return 0;
}

int zaokrozi(double stevilo) // definicija funkcije
{
    int temp;
    stevilo += 1; // podanemu številu pristojemo 1
    temp = (int) stevilo; // s pretvorbo tipa iz duble v int odrezemo vse za decimalno
    piko // funkcija nam vrne vrednost zaokroževanja navzgor
    return temp;
}
```

6. Napišite program, ki ob pritisku na tipko Tn (0 _ n _ 3) spremeni stanje diode LED LDn.

če je dioda pred pritiskom ugasnjena, naj se prižge, sicer naj se ugasne.

Različica problema: Dodajte še stanje utripanja diode. če pritisnemo tipko, ko je dioda ugasnjena, naj se prižge, če je prižgana, naj začne utripati, če pa utripa, naj ugasne.

```
#include "io.h"

int main(void)
{
    _KeyInit();
    _LEDInit();
    _TimerInit();

    while(1)
    {
        _clrleds(15);
        getch();
        _setleds(15);
        getch();
        while(kbhit() == 0)
        {
            _clrleds(15);
            delay(200);
            _setleds(15);
            delay(200);
        }
        delay(200);
        getch();
    }
    return 0;
}
```

7. Napišite in v programu uporabite funkcijo, ki iz znakovnega niza, ki ji ga podamo kot parameter, izloči vse presledke.

Primer klica funkcije:

```
char formula[] = "x * (a + b)";  
odstrani_presledke(formula); //v nizu formula ostane "x*(a+b)"
```

Različica problema: Napišite tudi funkcijo, ki nadomesti poljuben niz zaporednih presledkov z enim samim presledkom.

Primer klica funkcije:

```
char msg[] = "Joj, to je pa težko!";  
odstrani_presledke(msg); //v nizu msg ostane "Joj, to je pa težko!"
```

```
#include "io.h"  
  
void odstrani(char *k, char *p);           // prototip funkcije izpis  
  
int main(void)  
{  
  _LCDInit();  
  _KeyInit();  
  _setcursortype(_NOCURSOR);  
  char ooo[] = "neki s   presledki";      // znakovni niz s presledki  
  char izpis[] = "";                       // znakovni niz za zapis ooo brez presledkov  
  
  printf("%s", ooo);                       // prvi izpis  
  getch();  
  clrscr();  
  
  odstrani(&ooo[0], &izpis[0]);           // klic funkcije odstrani  
  printf("%s", izpis);                     //izpis znakovnega niza izpis  
  while(1);  
  return 0;  
}  
  
void odstrani(char *k, char *p)           // definicija funkcije izpis  
{  
  int i=0;  
  int j=0;  
  while(1)  
  {  
    if(*(k+i) != ' '){*(p+j) = *(k+i);i++;j++;}  
    // če je element *(k+i) ni enak presledku prepíše element v *(p+j) in poveča oba števca  
  
    if(*(k+i) == ' ')i++;  
    // če je element *(k+i) presledek poveča samo i  
    if(*(k+i) == 0){*(p+j)=0;break;}  
    // ob nultem znaku v *(k+i) v *(p+j) postavi nulni znak  
  }  
  return 0;  
}
```

Prvi program je rešitev osnovnega problema, iz niza ooo[] odstrani vse presledke,

```

#include "io.h"

void odstrani(char *k, char *p);

int main(void)
{
    _LCDInit();
    _KeyInit();
    _setcursortype(_NOCURSOR);
    char ooo[] = "neki velik dalsga      s  presledki";
    char izpis[] = "";

    printf("%s", ooo);
    getch();
    clrscr();

    odstrani(&ooo[0], &izpis[0]);
    printf("%s", izpis);
    while(1);
    return 0;
}

void odstrani(char *k, char *p)
{
    int i=0;
    int j=0;
    while(1)
    {
        if(*(k+i) == 0){*(p+j)=0; break;}
        if(*(k+i) != ' '){*(p+j) =*(k+i); i++; j++;}
        if(*(k+i) == ' ' & *(k+i+1) != ' '){*(p+j) =*(k+i); i++; j++;}
        if(*(k+i) == ' ' & *(k+i+1) == ' ')i++;
    }
    return 0;
}

```

Drugi program je rešitev različice problema, znakovnemu nizu odstrani le presledke ki sledijo drugemu presledku.

8. Napišite program, ki ob pritisku na tipko Tn (0 _ n _ 3) na prikazovalniku povzroči utripajoče prikazovanje besedila "Tipka Tn". Najkasneje po treh sekundah naj prikaz izgine, če pa kdo v tem času pritisne kakšno drugo tipko, naj se takoj popravi utripajoče besedilo (v resnici se spremeni samo n, ki predstavlja zaporedno številko pritisnjene tipke.)

```
#include "io.h"

short blink(void);
// funkcija ki daje takt utripanju

void printC(int px);
// izpisuje vrednost tipke dokler je blink 1, če vmes ne pritisnemo nobene tipke

int main(void)
{
    _LCDInit();
    _KeyInit();
    _TimerInit();
    _setcursortype(_NOCURSOR);
    int px; // hrani vrednost tipke
    int t;
    while(1)
    {
        if(kbhit()){px=getch();t = clock();}
        // če pritisneš tipko jo vpise v px in shrani čas vpisa v t

        if((clock() - t) <= 3000)printC(px);
        // omeji čas izvajanja printC na 3000 milisekund

        clrscr();
        // po izvajanju pobriše zaslon
    }
}

// funkcije

short blink(void)
{
    if((clock()/500) % 2 == 0)return 1; // pogoj za takt
    return 0;
}

void printC(int px)
{
    while(blink())
    {
        printf("tipka %c \r", px); // izpis
        if(kbhit())break; // prekine ob pritisku
    }
}
```

9. Napišite program, ki uporabniku omogoča, da s pomočjo štirih tipk vnese račun oblike <desetiska cifra><operator><desetiska cifra> Po zaključnem vnosu naj program izpiše pravilen rezultat. Na primer, če uporabnik vtipka 4*9, naj se na zaslonu pokaže 4*9=36. Različica problema: Dopolnite program tako, da bo deloval tudi za večmestna cela števila.

```
#include "io.h"

int main(void)
{
    _LCDInit();
    _KeyInit();
    _setcursortype(_NOCURSOR);
    _TimerInit();
    int x;           // prva številka
    int y;           // druga številka
    char o;          // cifra operatorja
    short k;         // vrednost tipke
    double r;        // rezultat

    while(1)
    {
        x = 0;
        y = 0;
        o = 0;
        while(1)          // izbor prve številke
        {
            clrscr();
            printf("prva številka: %i",x);
            k = getch();
            switch(k)
            {
                case '0' : x *= -1; break;           // s tipko 0 zamenjamo predznak
                case '1' : x += 10; break;          // s tipko 1 prištejemo 10
                case '2' : x += 1; break;           // s tipko 2 prištejemo 1
            }
            clrscr();
            if(k == '3')break;                       // s tipko 3 prekinemo izbor prve številke
        }
        delay(100);
        while(1)          // izbor operatorja
        {
            if(k == '0' || k == '1' || k == '2')o++; // tipke 0, 1, 2 povečujejo vrednost o
            if(o >= 4)o = 0;                          // če je o več ali enako 4 ga vrne na 0
            clrscr();
            switch(o)          // izpisovanje operatorja glede na vrednost o
            {
                case 0 : printf("+");break;
                case 1 : printf("-");break;
                case 2 : printf("*");break;
                case 3 : printf("/");
            }
            k = getch();
            if(k == '3')break; // tipka 3 prekine izbor operatorja
        }
        while(1)          // izbor druge številke
        {
            clrscr();
            printf("druga številka: %i",y);
            k = getch();
            switch(k)
            {
                case '0' : y *= -1; break;           // s tipko 0 zamenjamo predznak
                case '1' : y += 10; break;          // s tipko 1 prištejemo 10
                case '2' : y += 1; break;           // s tipko 2 prištejemo 1
            }
            if(k == '3')break; // s tipko 3 prekinemo izbor druge številke
        }
        switch(o)          // izbira operacije glede na cifro operatorja (o)
        {
            case 0 : r=(double)x+(double)y;break;
            case 1 : r=(double)x-(double)y;break;
            case 2 : r=(double)x*(double)y;break;
            case 3 : r=(double)x/(double)y;break;
        }
        clrscr();
    }
}
```

```
    printf("rezultat je %f", r);  
    getch();  
    clrscr();  
}  
return 0;  
}
```

Program bi bil lepši s funkcijami, ker pa v času njegovega nastanka s funkcijami še nisem bil v dobrih odnosih je program tak kot je. Tudi sama izvedba upravljanja na šarmu ni najboljša. Tipka t0 spremeni predznak cifri s katero trenutno upravljamo, tipka t1 trenutni številki prišteje 10, tipka t2 pa prišteje 1. Tako lahko s spreminjanjem predznaka in prištevanjem 1 ali 10, dosežemo katerokoli številko tudi če smo želeno vrednost prekoračili.

Primer na ustnem izpitu je bil sledeč: Napiši program ki bo na zaslonu šarma izpisoval uro, ki jo bo mogoče natančno nastaviti s tipkami.

Rešitev tega problema je podana že v učbeniku, tu pa je moja rešitev:

```
# include "io.h"

void display(int add);           // skrbi za izpis ure
int blink(void);                // skrbi za utripanje prikaza ure v funkciji change()
int change(int k);              // omogoči prilagoditev ure

int main(void)
{
    _LCDInit();
    _KeyInit();
    _TimerInit();
    _setcursortype(_NOCURSOR);
    int zzz = 0;                 // število sekund prilagoditve ure

    while(1)
    {
        display(zzz);
        if(kbhit()){delay(100);if(kbhit())zzz = change(zzz);}
    }

    // funkcije

    void display(int add)        // prikazuje uro glede na clock() in vrednost add
    {
        int s;
        int min;
        int h;

        s = clock()/1000 + add;  // dobi število sekund
        min = s/60;              // dobi število minut
        h = min/60;              // dobi število ur
        printf("%2u : %2u : %2u \r",h%24,min%60,s%60);
        // operator % vrne le število s/ min/ h ki ostanejo pri deljenju z 24/ 60 /60
    }

    int blink(void)              // skrbi za utripanje izpisa
    {
        if((clock()/500)%2)return 0;
        else return 1;
    }

    int change(k)
    {
        int x;

        int min;
        int h;
        int s;

        while(x != '3')
        {
            x = 0;
            if(kbhit())x = getch();
            switch(x)
            {
                case '0': k+= 60*60; break;
                case '1': k+= 60; break;
                case '2': k+= 1; break;
            }
            s = clock()/1000 + k;
            min = s/60;
            h = min/60;
            if(k >= 24*60*60)k = 0;
            // poskrbi da popravek ne preseže največje smiselne vrednosti
            if(blink())printf("%2u : %2u : %2u \r",h%24,min%60,s%60);
            else clrscr();
        }
        return k; // vraca stevilo sekund prilagositve
    }
}
```