

```

/*
V minulém programu se nám podařilo rozblikat řadu LED diod pomocí cyklu,
čekání a dvou konstant (STAV1 a STAV2) které nám určovaly, jaké ledky mají
svítit. Problém je v tom, že po napsání první konstanty musíme vytvořit tu
druhou přesně opačně (kde byly jedničky, budou nuly, a naopak), protože jinak
by nám některé ledky buďto vůbec nesvítily, nebo by svítily stále. Nyní se
proto pokusíme vytvořit podobný program jako minule, který ale bude potřebovat
nedefinovat pouze jeden stav a druhý si vypočte sám.
K tomuto účelu se přesně hodí takzvaný bitový komplement. bitový komplement je
operace, při níž se z jedniček stanou nuly, a naopak.
*/

```

```

/*
Teď ještě vědět, jak takovou operaci zapsat.
Na tomto místě se hodí uvést tabulku operátorů jazyka C, mezi které patří
i náš bitový komplement.
*/

```

```

/*
Operátory v C:

```

```

+   sčítání
-   odčítání
*   násobení
/   dělení
%   modulus      (zbytek po celočíselném dělení)
++  inkrementace (zvětšení o 1)
--  dekrementace (zmenšení o 1)
=   přiřazení
==  porovnání    ( if(A==B)...
~   bitový komplement ( 0b11110000 -> 0b00001111)
!   not
!=  není rovno
<   menší než
>   větší než
<=  menší nebo rovno
>=  větší nebo rovno
&   bitové AND   (také adresa proměnné)
&&  logické AND
|   bitové OR
||  logické OR
^   bitové exclusive OR
?   podmínkový operátor
<<  bitový posun doleva (A = B << 2 posun bitů o dvě místa doleva,
                                zprava doplňuje nuly)
>>  bitový posun doprava
+=  A += B totéž co A = A + B
-=  A -= B totéž co A = A - B
/=  podobně...
%=
&=
*=
^=
|=
>>=
<<=
*/
/*
Nyní se tedy pokusíme napsat program:
*/

```

```

#define F_CPU 1000000UL // 1 MHz (základní frekvence)

#include <avr/io.h> //Knihovna vstupů a výstupů (PORT, DDR)
#include <util/delay.h> //Knihovna čekacích funkcí

```

```
#define CEKANI 500    //kolik milisekund budou ledky svítit
#define STAV1 0b01010101//které ledky se budou střídat

int main (void)
{
  DDRB = 0b11111111;
  PORTB= STAV1;
  for(;;)
  {
    PORTB=~PORTB;    //Zapíšeme na portB bitový komplement původní hodnoty
                    //Vyměníme ledky - ty které svítily zhasnou, a naopak
    _delay_ms (CEKANI); //čekání
  }
}

/*
Pozorní si jistě všimli, že se konstrukce programu zjednodušila.
Po nahrání programu do přípravku "ATmega8 - LED panel" budou střídavě blikat
LED diody tak, jak je to definováno v konstantě STAV1, rychlostí určenou
v konstantě CEKANI. Tyto konstanty lze libovolně měnit...
*/

//Pro radioklub OK1KVK naspal Vašek Král
```