

1. Realizovaného pomocí funkce sprintf

```
unsigned long binarni=0x2569;  
unsigned char vystup[5];  
  
void main()  
{    sprintf(vystup,"%d",binarni);  
konec: goto konec;  
}
```

2. Realizovaného pomocí dělení a funkce modulo

```
unsigned long binarni=0x2569;
unsigned char vystup[5];

void main()
{
    volatile unsigned char z;
    for (z=0; z<4; z++)
    {
        vystup[3-z] = binarni%10;
        binarni/=10;
    }
    konec: goto konec;
}
```

3. Realizovaného pomocí dělení a funkce modulo

```
unsigned long binarni=0x2569;
unsigned char vystup[5];

void main()
{
    vystup[0] = binarni/1000;
    vystup[1] = (binarni%1000)/100;
    vystup[2] = ((binarni%1000)%100)/10;
    vystup[3] = ((binarni%1000)%100)%10;
konec: goto konec;
}
```

4. Realizovaného pomocí dekadické předkorekce

```

unsigned long binarni=0x2569;
unsigned char vystup[5];
void main()
{
    volatile unsigned int z, korekce;
    for (z=0; z<16; z++)
    {
        korekce=0;
        if ((binarni&0x000F0000) >=0x00050000)
            korekce|=0x00030000;
        if ((binarni&0x00F00000) >=0x00500000)
            korekce|=0x00300000;
        if ((binarni&0x0F000000) >=0x05000000)
            korekce|=0x03000000;
        if ((binarni&0xF0000000) >=0x50000000)
            korekce|=0x30000000;
        binarni+=korekce;  binarni+=binarni;}
//   Převod komprimovaného BCD do pole vystup
konec: goto konec;

```

4. Realizovaného pomocí dekadickou předkorekcí

```

unsigned long binarni=0x2569;
unsigned char vystup[5];
void main()
{
    volatile unsigned int z, y, korekce, k;
    for (z=0; z<16; z++)
    {
        korekce=binarni>>16; k=0x00030000;
        for (y=0; y<4; y++)
        {
            if ((korekce&0xF)>=5) binarni+=k;
            korekce>>=4; k=k<<4; }
        binarni+=binarni;
    }
    // Převod komprimovaného BCD do pole vystup
    konec: goto konec;
}

```

Řešení o 30% kratší, ale o 120% delší (tj.o 120us)

Program SMYČKA versus OTEVŘENÁ SMYČKA

4. Realizovaného pomocí dekadické předkorekce

```
// Převod komprimovaného BCD do pole vystup
```

Varianta a)

```
vystup[0] = binarni>>28;  
vystup[1] = (binarni&0x0F000000)>>24;  
vystup[2] = (binarni&0x00F00000)>>20;  
vystup[3] = (binarni&0x000F0000)>>16;
```

Varianta b)

```
vystup[0] = * ((char*) (&binarni)+3)>>4;  
vystup[1] = * ((char*) (&binarni)+3) &0x0F;  
vystup[2] = * ((char*) (&binarni)+2)>>4;  
vystup[3] = * ((char*) (&binarni)+2) &0x0F;
```

4. Realizovaného pomocí dekadické předkorekce

```
// Převod komprimovaného BCD do pole vystup
```

```
Union
```

```
{    unsigned int bcd;  
    unsigned char znaky[4];  
}    rozdeleni;
```

```
Varianta c)
```

```
    rozdeleni.bcd=binarni;  
vystup[0] = rozdeleni.znaky[3]>>4;  
vystup[1] = rozdeleni.znaky[3]&0x0F;  
vystup[2] = rozdeleni.znaky[2]>>4;  
vystup[3] = rozdeleni.znaky[2]&0x0F;
```

UKÁZKY PROGRAMOVÝCH ŘEŠENÍ PŘEVODU BINÁRNĚ-DEKADICKÉHO

Vyhodnocení – Počet bytů programu

Řešení	A R M C-byty/D-byty	Stroj. cykly	8 0 5 1 C-byty/D-byty	Stroj. cykly	A V R C-byty/D-byty	Stroj. cykly
1	504 / 16	635	1075 / 39,1	588	1170/9	2253
2	68 / 16	199	283 / 20	4795	180 / 10	6060
3	128 / 16	106	374 / 18	5383	192 / 9	6711
4a	180 / 16	1212	254 / 22	3952	408 / 9	4700
4b	180 / 16	1212	204 / 22	2605	334 / 9	3713
4c	192 / 20	1220	203 / 26	2581	350 / 13	3721

1. Realizovaného pomocí funkce `sprintf`
2. Realizovaného pomocí dělení a funkce `modulo`
3. Realizovaného pomocí dělení a funkce `modulo`
4. Realizovaného pomocí dekadické předkorekce
 - a. Konverze komprimovaného BCD pomocí posunů
 - b. Konverze komprimovaného BCD přetypováním
 - c. Konverze komprimovaného BCD pomocí `unionu`

VLOŽENÉ ASSEMBLEROVSKÉ ŘEŠENÍ PRO PROCESOR 8051

```
int main (void)
{ //      PREVOD TEPLoty Z BIN NA BCD
  bcd[0]=bcd[1]=bcd[2]=0;
  for (i=0; i<16; i++)
  {      teplota+=teplota;
#pragma asm
          MOV A,bcd+2
          ADDC A,ACC
          DA A
          MOV bcd+2,A
          MOV A,bcd+1
          ADDC A,ACC
          DA A
          MOV bcd+1,A
          MOV A,bcd+0
          ADDC A,ACC
          DA A
          MOV bcd+0,A
#pragma endasm
  }
  vystup[4]=bcd[0];vystup[3]=bcd[1]>>4;
  vystup[2]=bcd[1]&0x0F; vystup[2]=bcd[2]>>4;
  vystup[0]=bcd[2]&0x0F;
konec:   goto konec;}

```

Počet bytů programu C-76/D-9 , Strojových cyklů 672