

13. lekce

Operace v číselných soustavách

Miroslav Jílek

a) Sčítání

Princip: Sčítáme po řádech zprava doleva. Jestli je výsledek větší nebo roven základu, potom základ odečteme, zapíšeme zbytek a k dalšímu řádu přičteme jedničku.

Př.: v desítkové číselné soustavě sečtěte:

$$1954_{(10)} + 751_{(10)}$$

čísla v desítkové číselné soustavě si můžeme představit takto:

$$1954_{(10)}$$

$$0751_{(10)}$$

... a nyní číslice pod sebou budeme sčítat zprava doleva:

$4 + 1 = 5$, výsledek je **5** a do sousedního řádu přičteme **0**, protože číslo 5 je menší než číselný základ soustavy

$5 + 5 + 0 = 10$, výsledek je $10 - 10 = 0$ a do sousedního řádu přičteme **1**, protože 10 je větší nebo rovno základu číselné soustavy

$9 + 7 + 1 = 17$, výsledek je $17 - 10 = 7$ a do sousedního řádu přičteme **1**

$1 + 0 + 1 = 2$, výsledek je **2** a do sousedního řádu už nic nepřičteme

výsledek je **2705**₍₁₀₎

Př.: ve dvojkové číselné soustavě sečtěte:

$$10011_{(2)} + 1101_{(2)}$$

čísla ve dvojkové soustavě si můžeme představit takto:

$$010011_{(2)}$$

$$001101_{(2)}$$

... a nyní číslice pod sebou budeme sčítat zprava doleva:

$1 + 1 = 2$, výsledek je **0** a do sousedního řádu přičteme **1**

$1 + 0 + \mathbf{1} = 2$, výsledek je **0** a do sousedního řádu přičteme **1**

$0 + 1 + \mathbf{1} = 2$, výsledek je **0** a do sousedního řádu přičteme **1**

$0 + 1 + \mathbf{1} = 2$, výsledek je **0** a do sousedního řádu přičteme **1**

$1 + 0 + \mathbf{1} = 2$, výsledek je **0** a do sousedního řádu přičteme **1**

$0 + 0 + \mathbf{1} = 1$, výsledek je **1** a do sousedního řádu nic přičteme

výsledek je **100000**₍₂₎

Př.: ve dvacetisedmičkové číselné soustavě (*soustava se základem 27*) sečtěte:

V číselné soustavě se základem 27 používáme číslice 0..9 a A..Q
Q představuje číslici(!) 26.

$$\text{PMN9JJ}_{(27)} + \text{QHHC5}_{(27)}$$

čísla ve dvacetisedmičkové číselné soustavě si můžeme představit takto:

$$\begin{array}{r} \text{PMN9JJ}_{(27)} \\ 0\text{QHHC5}_{(27)} \end{array}$$

... a nyní číslice pod sebou budeme sčítat zprava doleva:

$$J + 5 = 19 + 5 = 24,$$

výsledek je **O** a do sousedního řádu přičteme **0** (protože číslo 24 není větší nebo rovno číselnému základu soustavy)

$$J + C + 0 = 19 + 12 + 0,$$

výsledek je 31, protože 31 je větší nebo rovno základu soustavy, odečteme základ soustavy $31 - 27 = 4$ a do sousedního řádu přičteme **1**

$$9 + H + 1 = 9 + 17 + 1,$$

výsledek je 27, odečteme $27 - 27 = 0$ a do sousedního řádu přičteme **1**

$$N + H + 1 = 23 + 17 + 1,$$

výsledek je $41 - 27 = 14$, to je **E** a do sousedního řádu přičteme **1**

$$M + Q + 1 = 22 + 26 + 1,$$

výsledek je $49 - 27 = 22$, to je **M** a do sousedního řádu přičteme **1**

$$P + 0 + 1 = 25 + 0 + 1,$$

výsledek je 26, to je **Q** a do sousedního řádu už nic nepřičteme

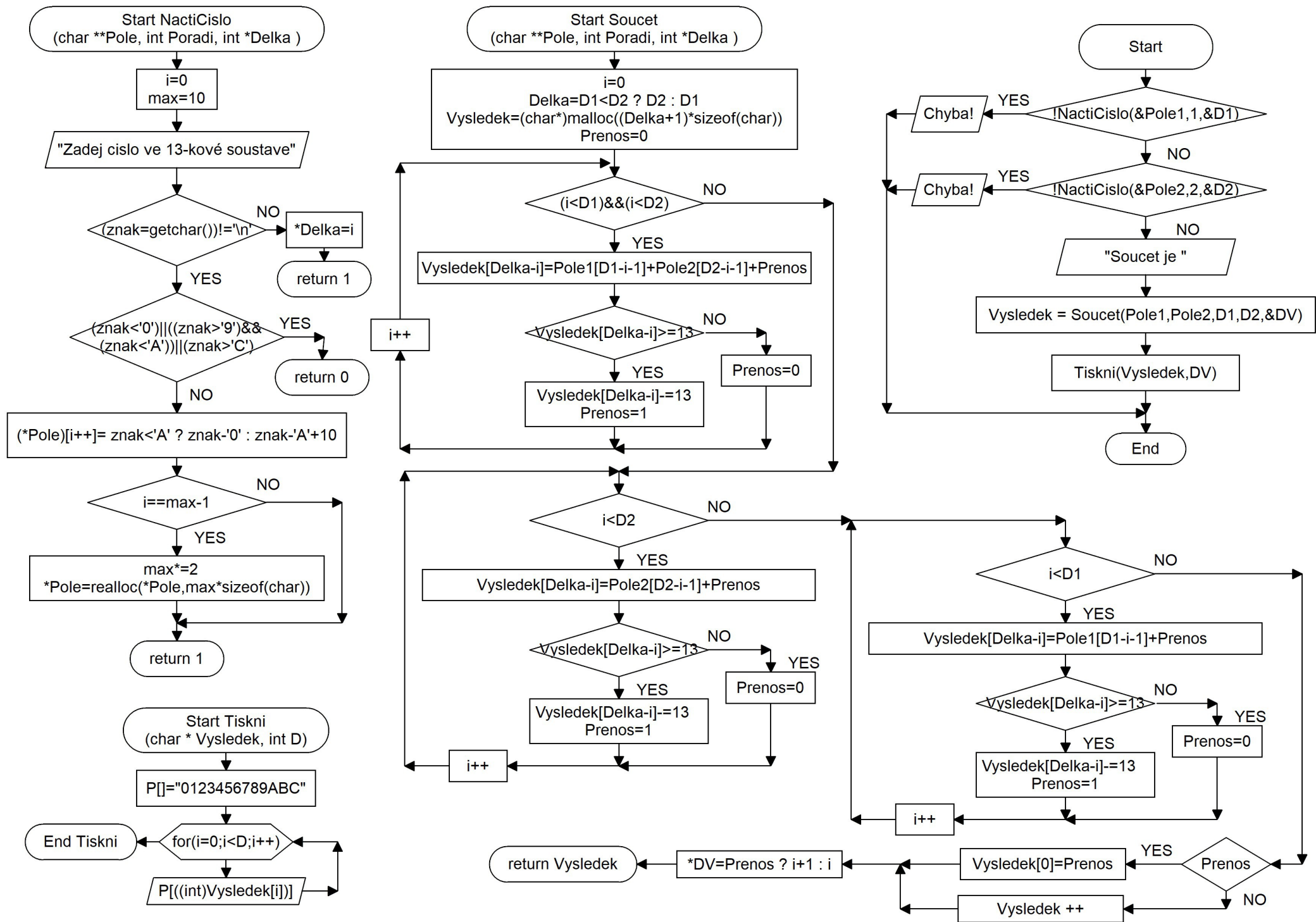
výsledek je **QME04O**₍₂₇₎

Úkol: vytvořte program, který provede součet dvou čísel ve třináctkové číselné soustavě

Nedívejte se na další stranu této prezentace a nejprve zkuste kód napsat (naprogramovat) sami.

Jestli se vám to nepodaří, pak se podívejte na další stránku této prezentace – na ní je vývojový diagram možného řešení.

A jestli vám nepomůže ani vývojový diagram, pak na posledních stránkách této prezentace naleznete řešení – kód. V řešení jsou použity techniky programování, které jsme již několikrát použili v předchozích lekcích...



Součet dvou čísel ve třináctkové soustavě:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int NactiCislo(char **Pole, int Poradi, int *Delka) //ukazatel na ukazatel na char
{
    int i=0, max=10;
    char znak;
    *Pole = (char*)malloc(max*sizeof(char));
    printf("Zadej %d. cele cislo ve trinactkove soustave: ", Poradi); //char vytiskne odpovidajici cislo
    while((znak=getchar())!='\n')
    {
        if ((znak<'0')||((znak>'9')&&(znak<'A'))||(znak>'C')) return 0;
        (*Pole)[i++] = znak<'A' ? znak-'0' : znak-'A'+10;
        if (i==max-1)
        {
            max*=2;
            *Pole=realloc(*Pole,max*sizeof(char));
        }
    }
    *Delka=i;
    return 1;
}

void Tiskni(char * Vysledek, int D)
{
    int i;
    char P[]="0123456789ABC";
    for(i=0;i<D;i++) putchar(P[((int)Vysledek[i]]));
    printf("\n");
}
```

```

char * Soucet(char *Pole1,char *Pole2, int D1, int D2, int * DV)
{
    int i=0,Delka=D1<D2 ? D2 : D1;
    char * Vysledek;
    Vysledek=(char*)malloc((Delka+1)*sizeof(char));
    char Prenos=0;
    while((i<D1)&&(i<D2))
    {
        Vysledek[Delka-i]=Pole1[D1-i-1]+Pole2[D2-i-1]+Prenos;
        if (Vysledek[Delka-i]>=13)
        {
            Vysledek[Delka-i]-=13;
            Prenos=1;
        }
        else Prenos=0;
        i++;
    }
    while(i<D2)
    {
        Vysledek[Delka-i]=Pole2[D2-i-1]+Prenos;
        if (Vysledek[Delka-i]>=13)
        {

```

```

            Vysledek[Delka-i]-=13;
            Prenos=1;
        }
        else Prenos=0;
        i++;
    }
    while(i<D1)
    {
        Vysledek[Delka-i]=Pole1[D1-i-1]+Prenos;
        if (Vysledek[Delka-i]>=13)
        {
            Vysledek[Delka-i]-=13;
            Prenos=1;
        }
        else Prenos=0;
        i++;
    }
    if (Prenos) Vysledek[0]=Prenos; else Vysledek ++;
    *DV=Prenos ? i+1 : i;
    return Vysledek;

```

```

}
```



```
int main (void)
{
    char * Pole1, *Pole2;
    int D1, D2, DV;
    if (!NactiCislo(&Pole1,1,&D1)) // posilame adresu adresy
    {
        printf("Chyba!\n");
        return 1;
    }
    if (!NactiCislo(&Pole2,2,&D2)) // posilame adresu adresy
    {
        printf("Chyba!\n");
        return 1;
    }
    char * Vysledek;
    printf("Soucet: ");
    Vysledek = Soucet(Pole1,Pole2,D1,D2,&DV);
    Tiskni(Vysledek,DV);
    return 0;
}
```