KATEDRA INFORMATIKY
PŘÍRODOVĚDECKÁ FAKULTA
UNIVERZITA PALACKÉHO V OLOMOUCI

Vícerozměrná pole

2. cvičení

Jiří Zacpal

KMI/ZP2 – Základy programování 2

Jednorozměrné pole – připomenutí

- deklarace:
```
int pole[5]={1,2,3,4,5};
```
- přístup k prvkům:
```
pole[4]=3;
printf("%i", pole[2]);
```
- vztah pointerů a polí:
`*(pole+i)` odpovídá zápisu `pole[i]`
- typ pole (resp. prvků v poli) může být libovolný
- nabízí se tedy následující otázky:
  - Lze vytvořit pole, jehož prvky budou pole?
  - Jak s tímto „vícerozměrným" polem pracovat?

Deklarace vícerozměrného pole

- bez inicializace (obecně):
  typ identifikátor[*rozmer1*]…[*rozmerN*];
- příklady:
```
int moje_matice[3][4];
float trojrozmerne[3][4][10];
```
- s inicializací:
  typ id[*r1*]…[*rN*]={*hodnoty v blocích*};
- příklady:
```
int matice[2][3]={{1,2,3},{4,5,6}};
int
m[2][3][4]={{{1,2,3,4},{5,6,7,8},{9,10,11
,12}},{{13,14,15,16},{17,18,19,20},{21,22
,23,24}}};
```

Přístup k prvkům pole

- pomocí operátorů indexu [ ]
- příklad:

```
int i, j;
int a[3][4];

for (i=0; i<3; i++)
   for (j=0; j<4; j++)
        a[i][j]=  1 + j + i * 4;

for (i=0; i<3; i++){
   for (j=0; j<4; j++)
        printf("%i\t", a[i][j]);
   printf("\n");
}
```

Dvourozměrné pole v paměti

- prvky pole `int x[2][3]` jsou v paměti uloženy v pořadí: `x[0][0]`, `x[0][1]`, `x[0][2]`, `x[1][0]`, `x[1][1]`, `x[1][2]`.
- na dvourozměrné pole se lze také dívat jako na jednorozměrné pole jednorozměrných polí daného typu (čili ukazatel na jednorozměrné pole daného typu).
- `x` je adresa dvourozměrné pole (typ `int[2][3]` nebo `int*[3]`)
- `x[0]` je adresa prvního řádku, `x[1]` je adresa druhého řádku (oba typ `int[3]` nebo `int*`)
- `x+1` a `x[0]+1` jsou tedy různé adresy

Pole jako parametr funkce

- jednorozměrné pole:

```
int maximum(int cisla[], int pocet){…}
…
int
cisla[10]={1,45,21,5,7,2,3,35,47,4};
max = maximum(cisla, 10);
```

- dvourozměrné pole:

```
int maxim(int cisla[][3], int
radku){…}
…
int cisla[2][3]={{1,45,21},{5,7,2}};
max = maxim(cisla, 2);
```

Úkol

- Napište v jazyku C funkci `int maximum(int prvky[][4], int radku)`, která vrátí hodnotu největšího čísla uloženého ve dvourozměrném poli prvky. První rozměr pole prvky lze určit pomocí parametru `radku`, druhý je pevně dán konstantou 4.

Příklad výstupu:

```
Vypis pole:
 10  2 15  -2
-52 41  0  12
 15  3  1  -8
Maximum je: 41
```

Úkol – řešení

```
int maximum(int prvky[][4], int radku)
{
        int max=prvky[0][0];
        for (int i=0;i<radku;i++)
                for (int j=0;j<4;j++)
                        if (prvky[i][j]>max) max=prvky[i][j];
return max;
}

void tisk(int prvky[][4], int radku)
{
        for (int i=0;i<radku;i++)
        printf("%5d %5d %5d %5d \n",
        *prvky[i],*(prvky[i]+1),*(prvky[i]+2),*(prvky[i]+3));
}
```

Bodovaný úkol

Napište v jazyku C funkci `int *suma_radku(int prvky[][4], int radku)`, která vypočítá součty na jednotlivých řádcích pole prvky a vrátí jednorozměrné pole obsahující tyto součty. První rozměr pole prvky lze určit pomocí parametru radku, druhý je pevně dán konstantou 4.