KATEDRA
INFORMATIKY
UNIVERZITA PALACKÉHO V OLOMOUCI

Vícerozměrná pole

Základy programování 2

Mgr. Markéta Trnečková, Ph.D.

Vícerozměrná pole

- pole, jejichž prvky jsou pole

```
typ jmeno[v1]...[vn]
```

Příklad

```
/* definice dvourozmerneho pole */
int pole[2][3];

/* definice ctyrrozmerneho pole */
int pole2[2][3][4][5];
```

Vícerozměrná pole

Inicializace

Příklad (vp_init.c)

```
/* jednorozmerne pole */
int pole1[4] = {1, 2, 3, 4};

/* dvourozmerne */
int pole2[2][3] = {{1, 2, 3}, {1, 2, 3}};

/* trojrozmerne */
int pole3[2][3][4] = {{{1, 2, 3, 4}, {1, 2, 3, 4}, {1, 2, 3, 4}},
                      {{5, 6, 7, 8}, {5, 6, 7, 8}, {5, 6, 7, 8}}};
```

Vícerozměrná pole

Uložení v paměti

`x[2][3]`

adresa	100	102	104	106	108	110	112
prvek	`x[0][0]`	`x[0][1]`	`x[0][2]`	`x[1][0]`	`x[1][1]`	`x[1][2]`	volno

Vícerozměrná pole

Příklad

Příklad

Napište funkci, která bude přijímat dvourozměrné pole a dva indexy `r` a `c` (případně další argumenty) a vypíše prvek na indexech `[r][c]`. Tuto funkci použijte ve funkci pro vypsání celého dvourozměrného pole.

Vícerozměrná pole

Příklad (pr4-vypis.c)

```
// m, n ... rozmery pole
void vypis_prvek(int** p, int r, int c)
{
    printf("%i ", p[r][c]);
}

int vypis(int** p, int m, int n)
{
    int i,j;
    for(i=0; i<m; i++){
        for(j=0; j<n; j++){
            vypis_prvek(p,i,j);
        }
        printf("\n");
    }
}
```

Proč tento kód nefunguje?

Vícerozměrná pole

Příklad (pr4-vypis2.c)

```c
// m, n ... rozmery pole
void vypis_prvek(int p[][10], int r, int c)
{
    printf("%i ", p[r][c]);
}

int vypis(int p[][10], int m, int n)
{
    int i,j;
    for(i=0; i<m; i++){
        for(j=0; j<n; j++){
            vypis_prvek(p,i,j);
        }
        printf("\n");
    }
}
```

Vícerozměrná pole

Uložení v paměti

`x[2][3]`

adresa	100	102	104	106	108	110	112
prvek	`x[0][0]`	`x[0][1]`	`x[0][2]`	`x[1][0]`	`x[1][1]`	`x[1][2]`	volno

Vícerozměrná pole

Uložení v paměti

`x[2][3]`

adresa	100	102	104	106	108	110	112
prvek	`x[0][0]`	`x[0][1]`	`x[0][2]`	`x[1][0]`	`x[1][1]`	`x[1][2]`	volno
	`x[0]`			`x[1]`			
	`x`						

Příklad

Jaký je výsledek:

```
x + 1
x[0] + 1
```

Vícerozměrná pole

Reprezentace jednorozměrným polem

Příklad

Je možné dvourozměrné pole `pole[m][n]` neprezentovat polem jednorozměrným? Jak?

Příklad

Napište funkci, která bude přijímat dvourozměrné pole, které je ale reprezentováno **jednorozměrným polem**, a dva indexy `r` a `c` (případně další argumenty) a vypíše prvek na indexech `[r][c]`. Tuto funkci použijte ve funkci pro vypsání celého dvourozměrného pole.

Reprezentace jednorozměrným polem

Dvourozměrné pole o rozměrech $m \times n \rightarrow$ jednorozměrné pole s $m \cdot n$ prvky

Příklad

```
/* deklarace, m a n jsou konstanty */
int pole1d[m*n];
int pole2d[m][n];

/* pristup k prvku na indexech i,j */
pole1d[i*n + j] = 12;
pole2d[i][j] = 12;
```

Vícerozměrná pole

Příklad (pr4-vypis3.c)

```
// m, n ... rozmery pole
void vypis_prvek(int p[], int r, int c, int m)
{
    printf("%i ", p[r*m + c]);
}

int vypis(int p[], int m, int n)
{
    int i,j;
    for(i=0; i<m; i++){
        for(j=0; j<n; j++){
            vypis_prvek(p,i,j,m);
        }
        printf("\n");
    }
}
```

Vícerozměrná pole

Alokace

1. Alokujeme pole pointerů o `m` prvcích.
2. Alokujeme `m` jednorozměrných polí o `n` prvcích, přičemž pointery uložíme do pole alokovaného v prvním kroku.

Příklad

```
/* 1. krok */
int **pole2d = malloc(m * sizeof(int *));

/* 2. krok */
for (i = 0; i < m; i++)
    pole2d[i] = malloc(n * sizeof(int));
```

Vícerozměrná pole
Dealokace

1. Dealokujeme `m` jednorozměrných polí.
2. Dealokujeme pole pointerů.

Příklad

```
/* 1. krok */
for (i = 0; i < m; i++){
    free(pole2d[i]);
    pole2d[i] = NULL;
}

/* 2. krok */
free(pole2d);
pole2d = NULL;
```

Vícerozměrná pole

Vyzkoušejte tento výpis v kombinaci s poli, která jsou alokována na haldě.

Příklad (pr4-vypis4.c)

```
// m, n ... rozmery pole
void vypis_prvek(int** p, int r, int c)
{
    printf("%i ", p[r][c]);
}

int vypis(int** p, int m, int n)
{
    int i,j;
    for(i=0; i<m; i++){
        for(j=0; j<n; j++){
            vypis_prvek(p,i,j);
        }
        printf("\n");
    }
}
```

Cvičení

1. Do kódu `pr4-vypis4.c` dopište kontrolu alokace a také správnou dealokaci.
2. Vytvořte funkci, beroucí celočíselné argumenty m a n, která alokuje a vrátí dvourozměrné pole o velikosti $m \times n$. Prvky pole budou reprezentovat násobky. Tedy prvek pole na indexu i, j bude roven $i \cdot j$. Vypište toto pole.
3. Přepište předchozí funkci s použitím reprezentace jednorozměrným polem.
4. Pomocí dvourozměrného pole lze reprezentovat hrací pole při piškorkách (prázdné políčko $= 0$, křížek $= 1$, kolečko $= 2$). Napište funkci, která prohledá toto dvourozměrné pole a vrátí nejdelší souvislou posloupnost křížků nebo koleček
   1. na řádku
   2. ve sloupci
   3. diagonálně
5. Naprogramujte hru piškvorky pro dva hráče.
   Dokud v poli nebude posloupnost 5 stejných znaků (využijte předchozí funkci) se budou hráči střídat a umisťovat svůj znak do pole (na střídačku budou hráči vyzváni, aby zadali 2 souřadnice, kam chtějí umístit znak).