

Jednorozměrné pole

Deklarace:

```
int A[10]; /* deseti prvkové pole typu int, prvky mají indexy od 0 do 9 */
float B[15]; /* patnácti prvkové pole typu float, prvky mají indexy od 0 do 14 */
```

použití symbolické konstanty:

```
#define MX 5
long X[MX], Y[MX]; /* dve pole o delce MX, tj. 5 */
```

použití symbolické konstanty a typedef:

```
#define VEL 10 /* definice konstanty */
typedef int VEC[VEL]; /* definice noveho datoveho typu */
VEC A, B, C; /* deklarace tri stejných poli, tj. deseti prvkových poli typu int */
```

Dvojitřozměrné pole

Deklarace:

```
int A[3][4]; /* dvojrozmerne pole typu int, 3 radky s indexy 0..2 a 4 sloupce s indexy 0..3 */
float B[5][5]; /* ctvercove dvojrozmerne pole typu float, cisla radku i sloupce jsou v intervalu 0..4 */
```

použití symbolické konstanty:

```
#define RX 2
#define SX 10
#define VEL 8
double X[RX][SX]; /* pole o 2 radcích (0..1) a 10 sloupcích (0..9) */
char Q[VEL][VEL]; /* ctvercove dvojrozmerne pole */
```

použití symbolické konstanty a typedef:

```
#define RX 4
#define SX 10
typedef int RAD[SX]; /* definice noveho datoveho typu, radek matice */
RAD MAT[RX]; /* deklarace dvojrozmerneho pole s nazvem MAT, ma 4 radky a 10 sloupcu */
```

za předpokladu deklarace

```
#define DELKA 100
int A[DELKA];
int i,j;
A[i]=5; /* ulozi 5 do i-teho prvku od zacatku pole, prvni prvek ma index 0 */
A[i+1]=A[i]+4; /* ulozi do i+1 prvku i-ty prvek zvetseny o 4 */
```

```
A[DELKA-i-1]=10; /* ulozi do i-teho prvku od konce 10, i muze nabyvat hodnot od 0 do 99 */
```

```
A[DELKA-i-1]=A[DELKA-i-2];
/* ulozi do i-teho prvku od konce predchozi prvek, i muze nabyvat hodnot od 0 do 98 */
```

```
printf("%d", A[i]); /* vypis i-teho prvku pole */
scanf("%d", &A[i]); /* nacteni i-teho prvku pole */
scanf("%d", &A[i]); /* totez, nacteni i-teho prvku pole */
scanf("%d", &A[0]); /* nacteni 0-teho prvku pole */
scanf("%d", A); /* totez, nacteni 0-teho prvku pole */
```

Algoritmy na jednorozměrné pole

pole1.c - načtení pole, výpis pole, součet prvků pole, hledání maxima
pole2.c - porovnání obsahu dvou polí (shoda/neshoda, počet rozdílů, výpis rozdílů)
pole3.c - kopírování pole, kopírování pole pozpátku
pole4.c - hledání maxima, hledání pozice maxima, hledání prvků s určitou vlastností
j_posl.c - zapsání posloupnosti do pole

Algoritmy na dvojrozměrné pole

dpole1.c - naplnění náhodnými čísly, součet hlavní a vedlejší diagonály, součet prvků pod diagonálou
dpole2.c - součet a součin čtvercových matic
dpole3.c - řešení soustavy rovnic
dpole4.c - záměna řádků, transpozice
d_posl.c - zapsání posloupnosti do pole