Ukazatele na funkce

Jiří Zacpal

DEPARTMENT OF COMPUTER SCIENCE
PALACKÝ UNIVERSITY, OLOMOUC

KMI/ZP2 Základy programování 2

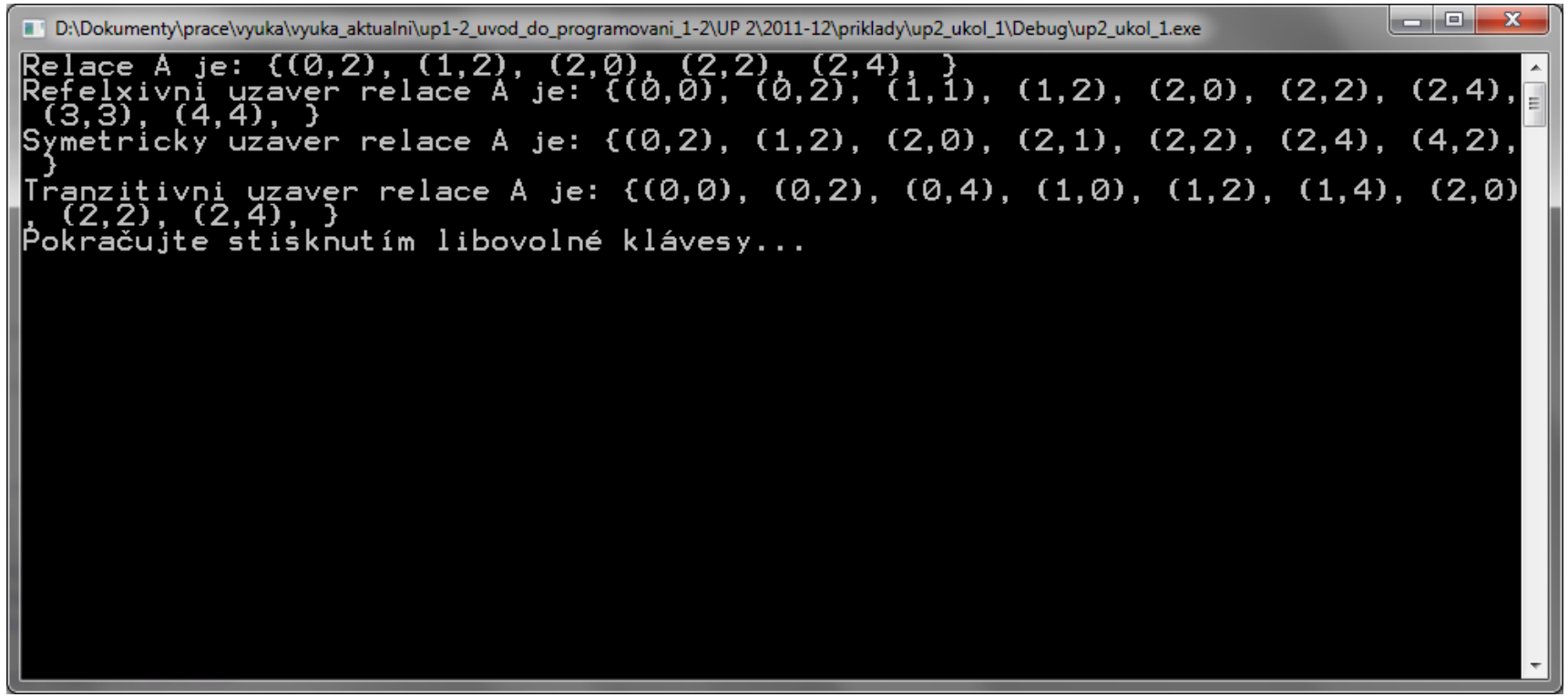
Zadání 1. příkladu

Napište funkce:

```
char **reflexivni(char **R, int n)
char **symetricky(char **R, int n)
char **tranzitivni(char **R, int n),
```

které pro zadanou binární relaci *R* na množině *M*={0,…,*n*-1} vytvoří reflexivní, symetrický a tranzitivní uzávěr relace R.

Zadání 1. příkladu



Zadání 1. příkladu

Nechť *R* je binární relace na množině *M* ($R \subseteq M \times M$).

- **Reflexivní uzávěr *R*** je relace $R \cup \{(x, x) \mid x \in M\}$
- **Symetrický uzávěr *R*** je relace $\{(x, y) \mid (x, y) \in R \text{ nebo } (y, x) \in R\}$
- **Tranzitivní uzávěr *R*** je relace $\bigcup_{i=1}^{\infty} \mathrm{T}^i(R)$ kde,
  - T^i je *i*-krát iterovaná aplikace funkce

$$\mathrm{T}(S) = S \cup \{(x, z) \mid \text{existuje } y \text{ takové, že } (x, y) \in S, (y, z) \in S\}$$

Ukazatel na funkci

- příklad deklarace:

```
double (*p_fd)(double);
```

- příklad deklarace s inicializací:

```
double polynomA(double x)
{
      return 3*x*x+4*x-10;
}
double (*p_fd)(double)=polynomA;
```

- typ ukazatele je někdy vhodné definovat pomocí konsturkce `typedef`:

```
typedef double (*P_FDD)(double);
P_FDD p_f = polynomA;
```

Práce s ukazatelem na funkci

- přiřazení adresy funkce do ukazatele:

```
p_f = polynomA;
```

- volání funkce pomocí ukazatele:

```
v = (*p_f)(-1);
```

nebo

```
v = p_f(-1);
```

- Výše uvedené možnosti volání funkce pomocí ukazatele na tuto funkci se nijak neliší; v praxi se používá ta níže uvedená.

Ukazatel na funkci jako parametr funkce

- definice funkce s parametrem typu ukazatel na funkci:

```
double *map(double (*fce)(double), double *vstup);
```

- volání funkce s parametrem typu ukazatel na funkci:

```
double na3(double x)
{
      return x*x*x;
}
...
pole_out=map(na3, pole);
```

- podobně při použití ukazatele na funkci:

```
double (*p_na3)(double) = na3;
pole_out=map(p_na3, pole);
```

Příklad 1

```
double na2(double x)
{return x*x;}

double na3(double x)
{return x*x*x;}

double *map(double (*fce)(double), double *vstup, int delka)
{
   int i;
   double *m;
   m=(double *)malloc(delka*sizeof(double));
   for(i=0;i<delka;i++)
       m[i]=fce(vstup[i]);
   return m;
}
```

Pole ukazatelů na funkce

- ukazatele (prvky pole) musí být stejného typu – funkce se stejným typem návratové hodnoty a stejnými typy parametrů
- deklarace pole ukazatelů na funkce:

```
double(*pole_fci[5])(double);
```

- deklarace s inicializací:

```
double(*pole_fci[])(double)={na3,na4,sin,cos,tan};
```

- s použitím dříve definovaného typu:

```
P_FDD pole_fci[] = {na3 , na4, sin, cos, tan};
```

- volání funkcí z pole:

```
vysl = pole_fci[1](1);
```