

PODATEK TIPa TABELA – ARRAY

Imenovanje podatkovnega tipa:

- Tabela, vektor (1D)
- Polje (2D)
- Matrika (3D)

1.) Predstavitev

Podatek tipa ARRAY ima 3 možne prdstavitve

a) ENODIMENZIONALNA TABELA

a

0	
1	
2	X
3	

0 .. 3

Če želimo poklicati tretjo celico naredimo takole: **a[2]**

b) DVOZIMENZIONALNA TABELA

a

	0	1	
			0
			1
		X	2
			3

0..3,0..1

Če želimo poklicati tretjo vrstico v drugem stolpcu naredimo takole: **a[2][1]**

c) TRIDIMENZIONALNA TABELA ali PROSTORSKA TABELA

a

0	0	1	2	3	0
					1
0..1,0..3					
1	0	1	2	3	0
		X			1
0..1,0..3					
2	0	1	2	3	0
					1
0..1,0..3					

Če želimo poklicati v prvi vrstici drugi stolpec druge tabele naredimo takole: `a[0][1][1]`

Vsebine posameznih segmentov:

- vsi primitivni podatkovni tipi (int, char, float,...)
- ...

Vsi segmenti v tabeli so enakega podatkovnega tipa.

Indeksi tabele so lahko:

- cela števila

Deklaracija

Tudi podatek tipa ARRAY je intervalni podatkovni tip.

int a[20]; - enodimenzionalna tabela celih števil

- Deklaracijsko pravilo v Cju

Zgledi deklaracij:

int polje[256];

float tabela[5];

char array[24];

int znak['z'];

int t[3][6];

char w[2][8][3];

DEKLARACIJA TABELE IN INICIALIZACIJA POLJ

int m[12] = {31,28,31,30,31,30,31,31,30,31,30,31 };

char d[7][20] = {"Ponedeljek","Torek","Sreda",
"Cetrtek","Petek","Sobota","Nedelja"};

m[1] => 28

d[1] => "Torek"

Prvo polje nam predstavi stolpec, drugo polje nam predstavi vrstico in tretje polje nam predstavi dimenzijo.

```
int x[2][4][3] = {{{0,1,2},
                  {3,4,5},
                  {6,7,8},
                  {9,10,11}},
                  {{12,13,14},
                  {15,16,17},
                  {18,19,20},
                  {21,22,23}}};
```

	<i>[0]</i>	<i>[1]</i>	<i>[2]</i>
x[0][0]	0	1	2
x[0][1]	3	4	5
x[0][2]	6	7	8
x[0][3]	9	10	11
x[1][0]	12	13	14
x[1][1]	15	16	17
x[1][2]	18	19	20
x[1][3]	21	22	23

	0	0	1	2	3
0	<i>0</i>	<i>3</i>	<i>6</i>	<i>9</i>	
1	12	15	18	21	

	1	0	1	2	3
0	<i>1</i>	<i>4</i>	<i>7</i>	<i>10</i>	
1	13	16	19	22	

	0	1	2	3
0	<i>2</i>	<i>5</i>	<i>8</i>	<i>11</i>
1	14	17	20	23

x[1][1][1] => 16
 x[1][0][2] => 14

VSTOP, IZSTOP V TABELO (pisanje, branje)

V tabelarični tip najlažje vstavljamo podatke s pomočjo FOR zanke.

A

--	--	--	--

0.3 - POSAMEZNI INDEKSI SO TIPA INTEGER, ZATO LAHKO UPORABIMO FOR ZANKO.

Klicanje:

a[0] – kličemo prvi segment

a[1] – kličemo drugi segment

a[2] – tretji segment

a[3] – četrti segment

Če je tabela enodimenzionalna potrebujemo eno for zanko.

Za dvodimenzionalne tabele potrebujemo dve vgnezdjeni for zanki.

Za trodimenzionalne tabele potrebujemo tri vgnezdene for zanke.

FOR ZANKE SE NE SMEJO SEKATI.

Vnos v tabelo

```
for (i=0;i<=4;i++)  
{  
    printf("Vnesi stevilo: "); scanf("%d",&t[i]);  
}
```

Izpis iz tabele

```
for (i=0;i<=4;i++)  
{  
    printf("Stevilo: %d\n",t[i]);  
}
```

Uporaba podprogramov za vnos in izpis podatkov iz eno dimensionalne tabele.

```
void vnos(int *t,int st) // ali void vnos(int t[],int st)
{
    int i,p;
    for (i=0;i<st;i++)
    {
        printf("Vnesi stevilo:");
        scanf("%d",&p); // ali scanf("%d",&t[i]);
        t[i] = p;
    }
}

void izpis(int *t,int st) // ali void izpis(int t[],int st)
{
    int i;
    for (i=0;i<st;i++)
    {
        printf("Stevilo %d\n",t[i]);
    }
}
```

Pokličemo vnos vrednosti in izpis vrednosti

```
int main(void)
{
    int t[5];
    vnos(&t[0],5); // ali pa tudi vnos(t,5);
    izpis(&t[0],5); // ali pa tudi izpis(t,5);
    return 0;
}
```

Uporaba podprogramov za vnos in izpis podatkov iz dvo dimensionalne tabele.

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

void beri(int t[10][10],int x,int y)
{
    int i,j;
    for (i=0;i<x;i++)
        for (j=0;j<y;j++)
        {
            printf("Vnesi vrednost: "); scanf("%d",&t[i][j]);
        }
}

void izpis(int t[10][10],int x,int y) // void izpis(int *t,int
x,int y)
{
    int i,j; // int i,j,c=0;
    for (i=0;i<x;i++)
    {
        for (j=0;j<y;j++)
            printf("%2d",t[i][j]); // printf("%2d ",t[c++]);
        printf("\n");
    }
}
```

```

int main(void)
{
    int p[10][10];
    beri(p,10,10);
    izpis(p,10,10); // izpis(p[0],10,10);
    return 0;
}

```

Naloge

- Izdelaj podprogram, ki prebere vrednosti enodimenzionalne tabele 10ih celih števil in podprogram, ki te vrednosti tudi izpiše
- Izdelaj podprogram, ki vrne vsoto števil v enodimenzionalni tabeli
- Izdelaj podprogram, ki poišče največje število v enodimenzionalni tabeli
- Izdelaj podprogram, ki poišče lokacijo, na kateri se nahaja največje število v enodimenzionalni tabeli
- Izdelaj podprogram, ki vrne vsoto števil na lihih mestih v enodimenzionalni tabeli
- Izdelaj podprogram, ki v dvozimenzionalno tabelo vstavi znak 'X' kot kaže slika

X				X
	X		X	
		X		
	X		X	
X				X

- Izdelaj podprogram, ki v dvozimenzionalno tabelo vstavi znak 'X' kot kaže slika

X	X	X	X	X
				X
X	X	X	X	X
X				
X	X	X	X	X