

гл.ас. д-р. Нора Ангелова

Основни конструкции

Указатели

- Оператор &

&<променлива>

променлива – вече дефинирана променлива.

Семантика

Намира адреса на променливата

Указатели

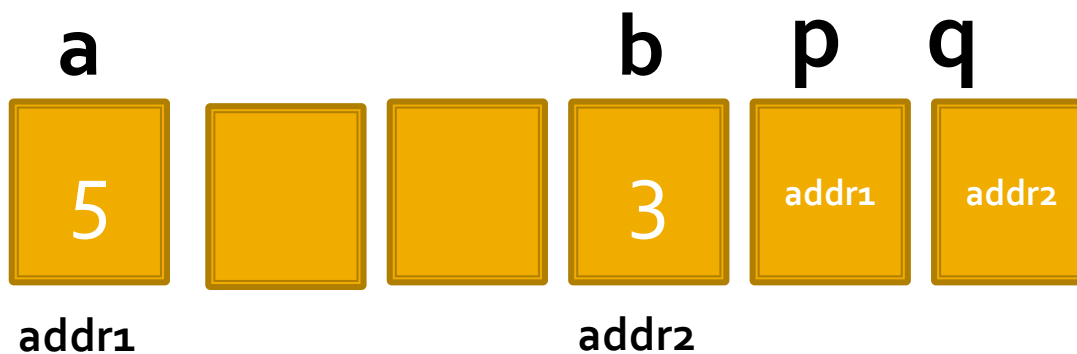
$T^* \text{ <променлива> [= <стойност>]_{\text{опц}}$

T^* – указател към променлива от тип T .

- T – име или дефиниция на тип.
- променлива – идентификатор.
- стойност – шестнадесетично число, представляващо адрес на данна от тип T или `nullptr` (NULL).

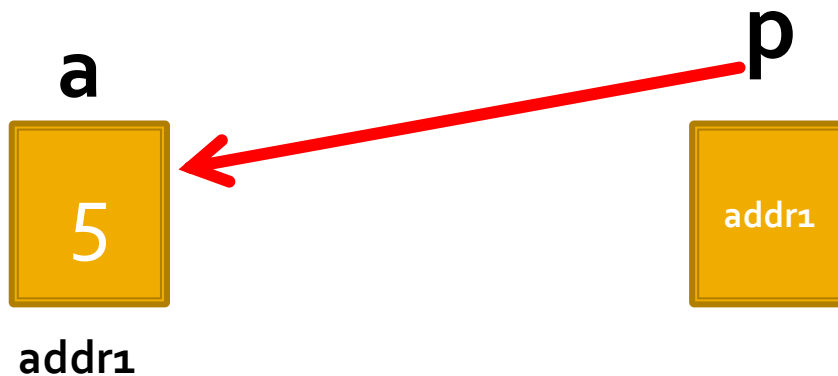
Указатели

```
int a = 5;  
int b = 3;  
int *p = &a;  
int *q = &q;
```



Указатели

```
int a = 5;  
int *p = &a;
```



$p == \text{addr1}$
 $*p == 5$

Указатели

```
int a = 5;  
int *p = &a;
```

`*p = 10; // присвоява нова стойност на a`



Указатели

Забележка:

Дефиницията: $T^* a, b;$

е еквивалентна на:

$T^* a;$

$T b;$

Дефиницията: $T^* a, ^*b;$

е еквивалентна на:

$T^* a;$

$T^* b;$

Аритметични и логически операции

■ +, -, ++, --, ==, !=, >, >=, <, <=

** Не е възможно въвеждане на данни от тип указател чрез оператора >>.*

Адресна аритметика

```
int *p;  
double *q;
```

■ $p = p + 1;$

премества указателя

$p = p + 1 * 4,$

4 е броят на байтовете, необходими за записване на данна от тип int.

■ $q = q + 1;$

премества указателя

$q = q + 1 * 8,$

8 е броят на байтовете, необходими за записване на данна от тип double.

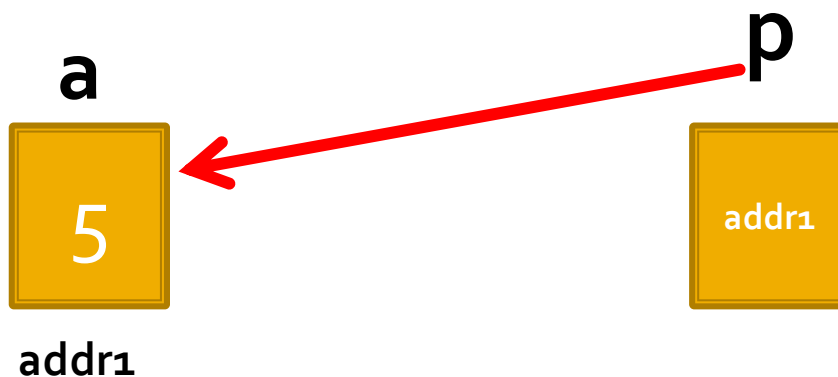
Указатели

- Изравняване на типове

```
int a = 5;  
int *p = &a;
```

a – int
&a – int*

p – int*
*p – int



Указатели и масиви

- Едномерни масиви
- Двумерни масиви

Задача:

Да се изведат стойностите на двумерен масив.

Да се използват указателна аритметика.

```
int matrix[5][4] = {  
    { 1,  2,  3,  4},  
    { 5,  6,  7,  8},  
    { 9, 10, 11, 12},  
    {13, 14, 15, 16},  
    {23, 24, 25, 26}  
};
```

```
for(int i=0; i < 5; i++) {  
    for(int j=0; j < 4; j++)  
        cout << (*(matrix + i)+j) << " ";  
    cout << endl;  
}
```

Указатели и масиви

```
int arr[40] = {1, 2, 3, 4};
```

arr – указател към първи елемент на масива.

arr – съдържа адрес на arr[0].

```
arr == &arr[0]
```



Какъв е типът на arr?

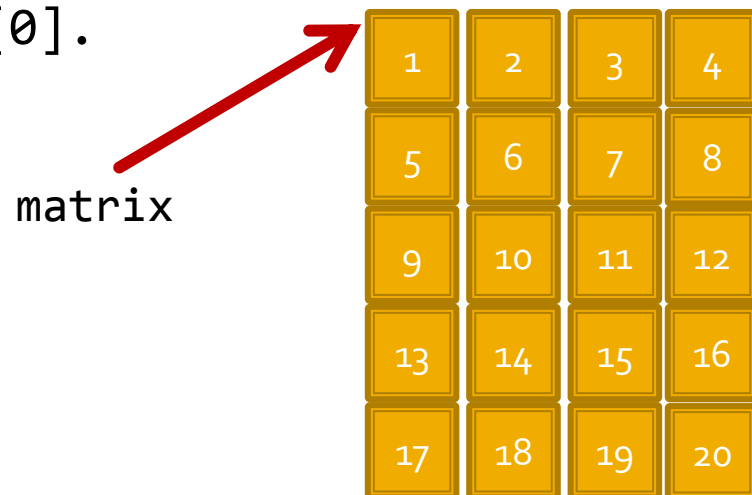
```
int *
```

Указатели и масиви

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

`matrix` – указател към първи елемент на масива, който също е масив.

`matrix` – съдържа адреса на `matrix[0]`.



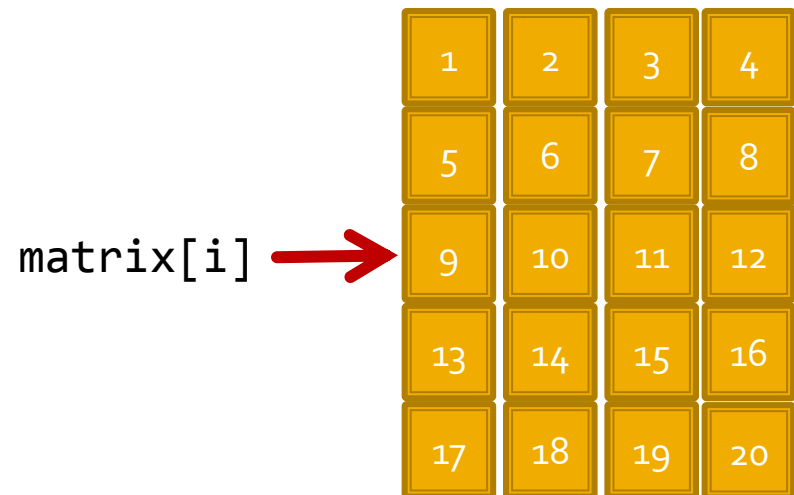
Указатели и масиви

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

`matrix[i]` – масив от цели числа.

`matrix[i]` – съдържа адреса на `matrix[i][0]`.

Какъв е типът на `matrix`?
<първи_елемент_на_масива>*
<масив_от_цели_числа>*
`int**`



Указатели и масиви

```
int arr[40] = {1, 2, 3, 4};  
int matrix[5][4] = {  
    { 1,  2,  3,  4},  
    { 5,  6,  7,  8},  
    { 9, 10, 11, 12},  
    {13, 14, 15, 16},  
    {23, 24, 25, 26}  
};
```

Да се изведе на стандартния изход:

```
cout << arr;  
cout << arr + 39;  
cout << *(arr + 4) << arr[4];  
cout << matrix;  
cout << matrix + 2;  
cout << *(matrix+3) + 1;  
cout << (*(matrix+4) + 3)  
      << matrix[4][3];
```

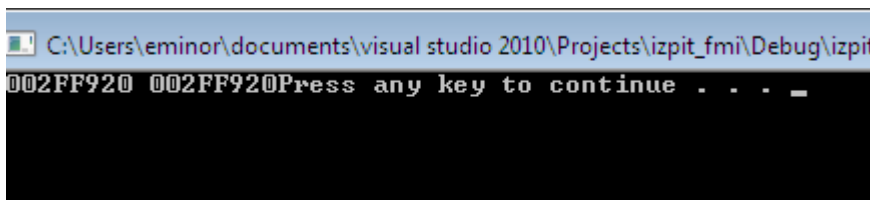
1. Адресът на първия елемент на arr;
2. Адресът на последния елемент на arr;
3. Петият елемент на arr;
4. Адресът на matrix;
5. Адресът на третия ред на матрицата matrix;
6. Адресът на елемента на четвърти ред и втори стълб на matrix;
7. Последният елемент на matrix;

Указатели и масиви

```
int arr[40] = {1, 2, 3, 4};  
int matrix[5][4] = {  
    { 1,  2,  3,  4},  
    { 5,  6,  7,  8},  
    { 9, 10, 11, 12},  
    {13, 14, 15, 16},  
    {23, 24, 25, 26}  
};
```

1. Адресът на третия ред на матрицата matrix;

```
cout << matrix + 2;           cout << *(matrix + 2) + 0;
```



Указатели и масиви

```
int arr[40] = {1, 2, 3, 4};  
int matrix[5][4] = {  
    { 1,  2,  3,  4},  
    { 5,  6,  7,  8},  
    { 9, 10, 11, 12},  
    {13, 14, 15, 16},  
    {23, 24, 25, 26}  
};
```

1. Адресът на третия ред на матрицата matrix;

```
cout << matrix + 8;          ERROR
```

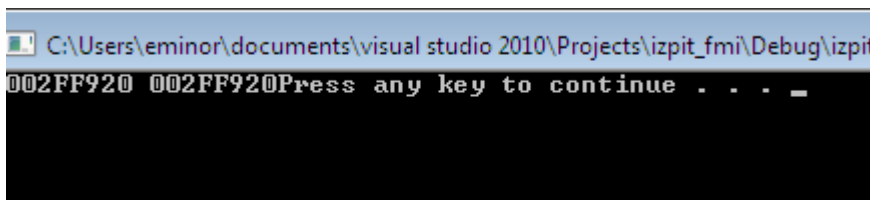
```
matrix + 8*sizeof(int*) (not int)
```

Указатели и масиви

```
int arr[40] = {1, 2, 3, 4};  
int matrix[5][4] = {  
    { 1,  2,  3,  4},  
    { 5,  6,  7,  8},  
    { 9, 10, 11, 12},  
    {13, 14, 15, 16},  
    {23, 24, 25, 26}  
};
```

1. Адресът на третия ред на матрицата matrix;

```
cout << *matrix + 8;
```



Празнични задачи

гл.ас. д-р. Нора Ангелова

Размяна на стойностите

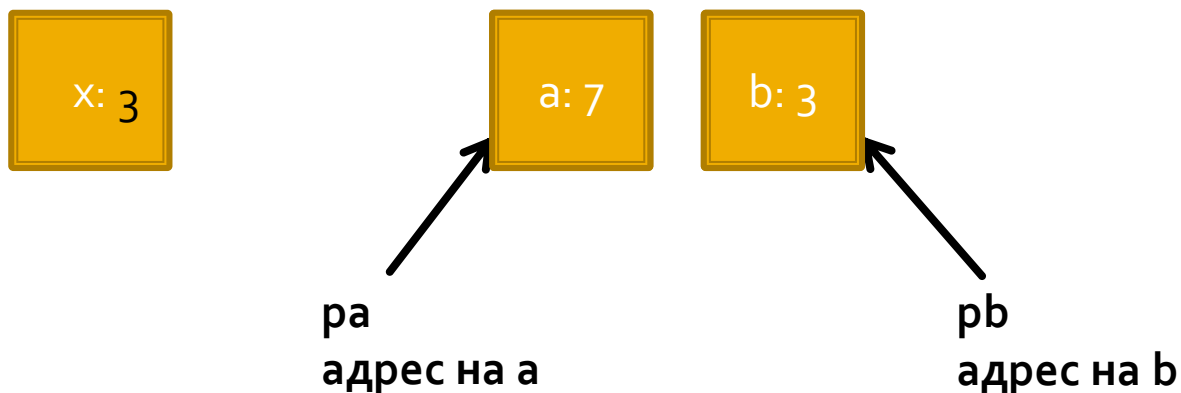
```
int a = 4, b = 7;  
int *pa = &a, *pb = &b;  
cout << *pa << " " << *pb << endl;
```

3 7

```
int x = *pa;  
*pa = *pb;  
*pb = x;
```

7 3

```
cout << *pa << " " << *pb << endl;
```



Извеждане

- Да се напише програма, която извежда на екрана – Halloween !

Почерпка или номер

- Да се напише програма, която въвежда число от потребител и казва: „Почерпка или номер“. Ако потребителят откаже да даде почерпка, програмата връща числото повдигнато на степен 7.
*Не използвайте вградени функции.

Минута е много

- Какво празнуваме на 1-ви ноември?

Денят на народните будители

е общобългарски празник, ознаменуващ делото на българските просветители, книжовници, революционери и свети будители на възраждащия се национален дух, стремеж към образование и книжовност.

Минута е много

- Кой са народните будители?

Кирил и Методий, Паисий Хилендарски, Петър Берон, братя Миладинови, Петко Славейков, Захари Стоянов, Васил Левски, Христо Ботев, Иван Вазов и още много българи

Народни будители

- Да се напише програма „Минута е много“. Програмата пита за име на народен будител и проверява дали въведеното име е вярно.

Вярно е всяко име, което е част от списъка. За име използвайте масив и елементи от тип `char`.

Край