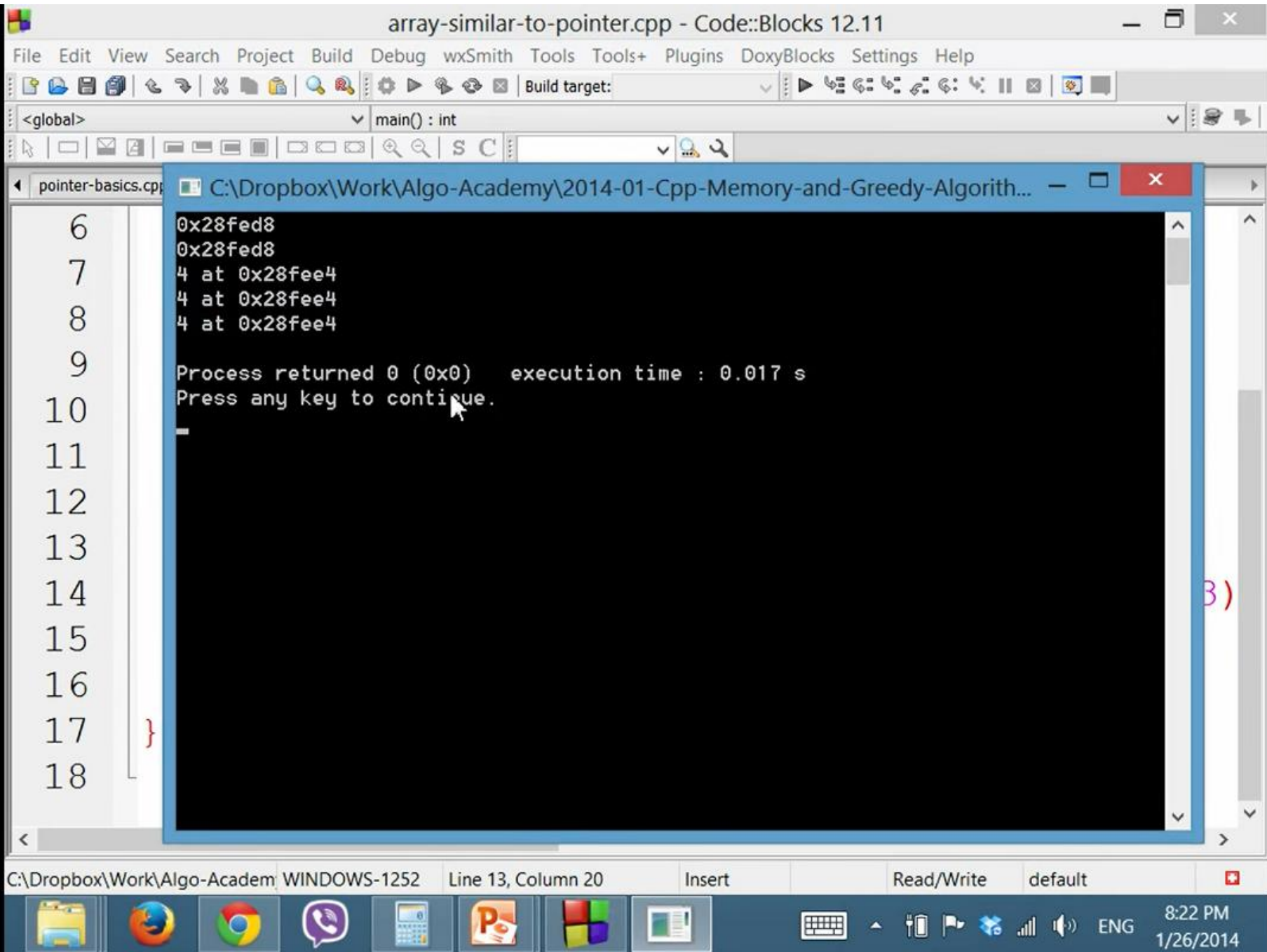


```
3
4  int main()
5  {
6      int arr[5] {1, 2, 3, 4, 5};
7
8      cout<<arr<<endl<<&(arr[0])<<endl;
9
10     int * arrPtr = arr;
11
12     cout<<arr[3]<<" at "<<&arrPtr[3]<<endl;
13     cout<<3[arr]<<" at "<<&3[arr]<<endl;
14     cout<< * (arrPtr + 3)<<" at "<<(arrPtr + 3)
15
16     return 0;
```







## Основни понятия

## Операция за изброяване

- $\langle \text{израз1} \rangle, \langle \text{израз2} \rangle$
- оценява и двата израза, но крайният резултат е оценката на втория израз
- $a, b, c, d \Leftrightarrow (a, (b, (c, d)))$
- **дясноасоциативна**
- използва се рядко
- Пример: `a = (cout << x, x);`

- Export PDF
- Create PDF
- Edit PDF
- Comment
- Combine Files
- Organize Pages
- Optimize PDF
- Fill & Sign
- Send for Signature
- Send & Track
- More Tools

Store and share files in the Document Cloud

[Learn More](#)



## Съкратени оператори за присвояване

- $a = a + 2 \Leftrightarrow a += 2$
- $--, *=, /=, %=$
- $a = a + 1 \Leftrightarrow ++a$
- $a = a - 1 \Leftrightarrow --a$
- $a++$  увеличава  $a$  с 1, но връща предишната стойност на  $a$ 
  - $a++ \Leftrightarrow (a = (tmp = a) + 1, tmp)$
- $a--$  действа аналогично
- $a++$  връща  $a$ , което е  $\langle lvalue \rangle$ 
  - Пример:  $++a += 5;$
- $a--$  връща предишната стойност на  $a$ , което е  $\langle rvalue \rangle$ 
  - Пример:  $x = a++ * b; a++ += 5;$

- Export PDF
- Create PDF
- Edit PDF
- Comment
- Combine Files
- Organize Pages
- Optimize PDF
- Fill & Sign
- Send for Signature
- Send & Track
- More Tools

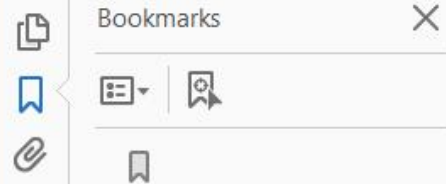
Store and share files in the Document Cloud

[Learn More](#)

## Циклични структури

## Оператор for — примери

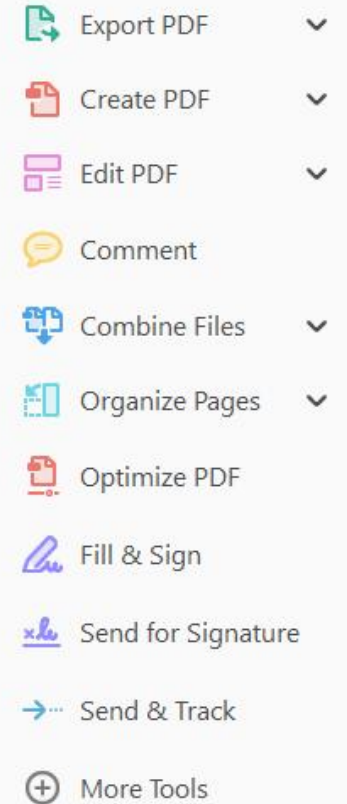
```
double sum = 0, x;  
int n;  
cout << "Въведете брой числа: "; cin >> n;  
for(int i = 1; i <= n; i++) {  
    cout << "Въведете число: ";  
    cin >> x;  
    sum += x;  
}  
cout << "Средно аритметично: " << sum / n << endl;  
  
for(int i = 1, x = 0, y = 1; i < 5; i++) {  
    x += i;  
    y *= x;  
}
```



## Основни понятия

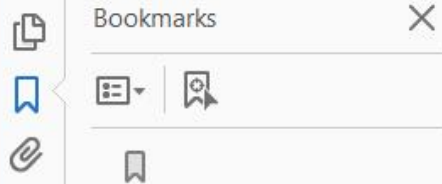
## Дефиниране на функция

- $\langle \text{сигнатура} \rangle \{ \langle \text{тяло} \rangle \}$
- $\langle \text{сигнатура} \rangle ::= [ \langle \text{тип\_резултат} \rangle \mid \text{void} ]$   
 $\langle \text{идентификатор} \rangle ( \langle \text{формални\_параметри} \rangle )$ 
  - `void` = празен тип, не връща резултат
  - ако типът на резултата се пропусне, подразбира се `int`
- $\langle \text{формални\_параметри} \rangle ::=$   
 $\langle \text{празно} \rangle \mid \text{void} \mid \langle \text{параметър} \rangle \{ , \langle \text{параметър} \rangle \}$
- $\langle \text{параметър} \rangle ::= \langle \text{тип} \rangle [ \langle \text{идентификатор} \rangle ]$ 
  - ако  $\langle \text{идентификатор} \rangle$  се пропусне, параметърът няма име и не се използва
  - Пример:  $f(x, y) = x + 5$



Store and share files in the Document Cloud

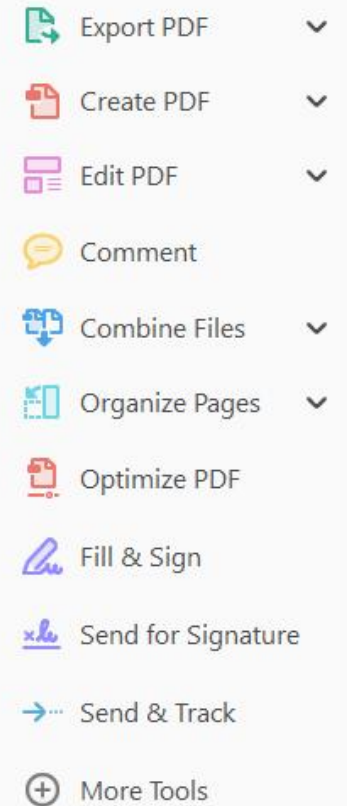
[Learn More](#)



## Основни понятия

## Деклариране на функция

- `<декларация_на_функция> ::= <сигнатура>;`
- Декларацията е “обещание” за дефиниция на функция
- Декларацията не е задължителна
- Една функция може да бъде декларирана няколко пъти...
- ...но може да бъде дефинирана **само веднъж**
- Неизпълнените обещания водят до проблеми...
  - ...освен когато никой не разчита на тях



Store and share files in the Document Cloud

[Learn More](#)



## Ограничени операции

- `strncpy(<буфер>, <низ>, n)`
  - копира първите  $n$  символа на  $\langle \text{низ} \rangle$  в  $\langle \text{буфер} \rangle$
  - винаги записва точно  $n$  символа в  $\langle \text{буфер} \rangle$ , допълвайки с  $\backslash 0$  при нужда
  - ако  $\langle \text{низ} \rangle$  има повече от  $n$  символа, не записва  $\backslash 0$ !
  - връща  $\langle \text{буфер} \rangle$
- `strncat(<низ1>, <низ2>, n)`
  - конкатенира първите  $n$  символа на  $\langle \text{низ}_2 \rangle$  след  $\langle \text{низ}_1 \rangle$
  - винаги поставя  $\backslash 0$  на края
  - все още е отговорност на програмиста да осигури достатъчно място в  $\langle \text{низ}_1 \rangle$ !
  - връща  $\langle \text{низ}_1 \rangle$
- `strncmp(<низ1>, <низ2>, n)`
  - сравнява първите  $n$  символа на  $\langle \text{низ}_1 \rangle$  и  $\langle \text{низ}_2 \rangle$
  - връща  $-1$ ,  $0$  или  $1$ , също като `strcmp`

- Export PDF
- Create PDF
- Edit PDF
- Comment
- Combine Files
- Organize Pages
- Redact
- Protect
- Optimize PDF
- Fill & Sign
- Send for Signature

Store and share files in the Document Cloud

[Learn More](#)







## Bookmarks



## Указатели

## Операции с указатели

- рефериране (<value>)
- дерефериране (\*<указател>)
  - унарна операция!
- сравнение (==, !=, <, >, <=, >=)
- указателна аритметика (+, -, +=, -=, ++, --)
- извеждане (<<)
- няма въвеждане! (>>)

- Export PDF
- Create PDF
- Edit PDF
- Comment
- Combine Files
- Organize Pages
- Redact
- Protect
- Optimize PDF
- Fill & Sign
- Send for Signature

Store and share files in the  
Document Cloud

[Learn More](#)





Bookmarks

- Bookmark 1
- Bookmark 2
- Bookmark 3
- Bookmark 4

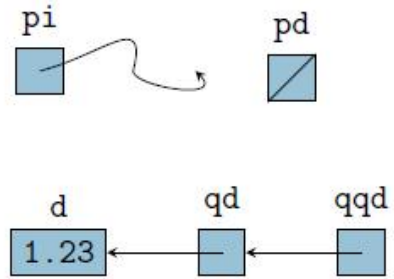
## Указатели

### Дефиниране на указателни променливи

```
<тип> *<име> [ = <израз> ] { , *<име> [ = <израз> ] };
```

#### Примери:

- `int *pi;`
- `double *pd = NULL;`
- `double d = 1.23;`
- `double *qd = &d;`
- `double **qqd = &qd;`



- Export PDF
- Create PDF
- Edit PDF
- Comment
- Combine Files
- Organize Pages
- Redact
- Protect
- Optimize PDF
- Fill & Sign
- Send for Signature

Store and share files in the Document Cloud  
[Learn More](#)



## Bookmarks



## Указатели и константи

## Указател към неизвестен тип

- **Проблем:** Не можем да насочваме един и същ указател към променливи от различен тип!
- **Решение:** `void*` — указател към неизвестен тип
- ✓ Преобразуваме автоматично от `T*` към `void*`
  - `int x, *p; void *q = p;`
  - `void *r = &x, *pr = &r, *s = &r;`
- × Няма автоматично преобразуване от `void*` към `T*`
  - `int* p; void* q = p;`
  - ~~`int* r = q;`~~
  - `int* s = (int*)q;`
- × Няма дереферирание (`void` е празният тип)
  - `int x; void* p = &x;`
  - ~~`*p = 2; void y = *p;`~~

- Export PDF
- Create PDF
- Edit PDF
- Comment
- Combine Files
- Organize Pages
- Redact
- Protect
- Optimize PDF
- Fill & Sign
- Send for Signature

Store and share files in the Document Cloud

[Learn More](#)



ENG

12:33 AM  
7/9/2018