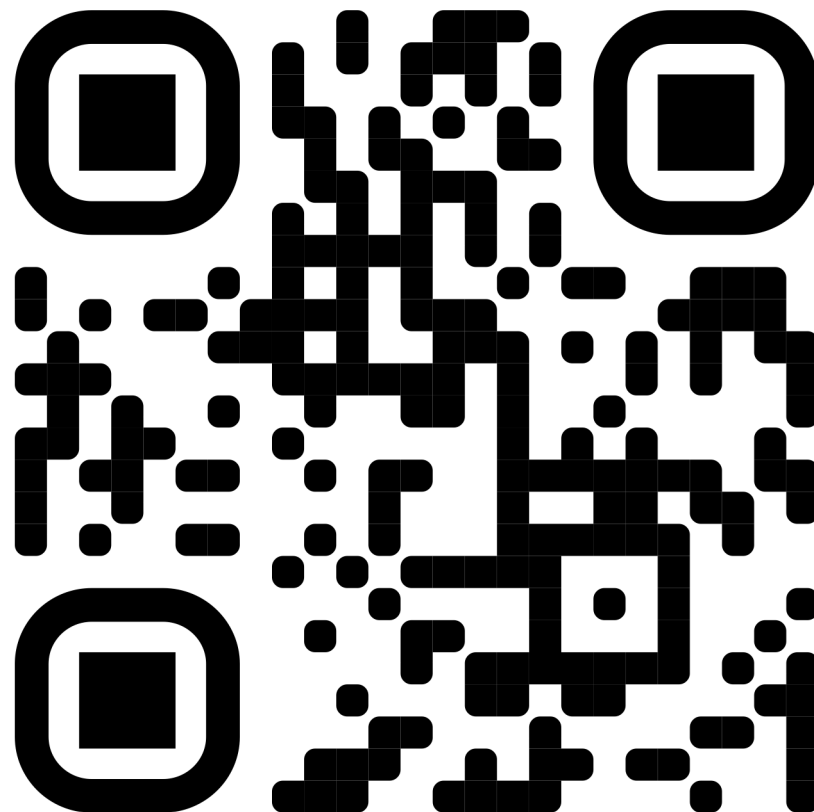


Указатели Функции

u.to/72D_Gg

Лекция 2, 12 февраля, 2021



Лектор:

Дмитрий Северов, кафедра информатики 608 КПМ

dseverov@mail.mipt.ru

Обратная связь : **u.to/7Wn7Gg**

УКАЗАТЕЛИ – косвенный доступ

- Переменные, содержащие адрес

- Описание и инициализация

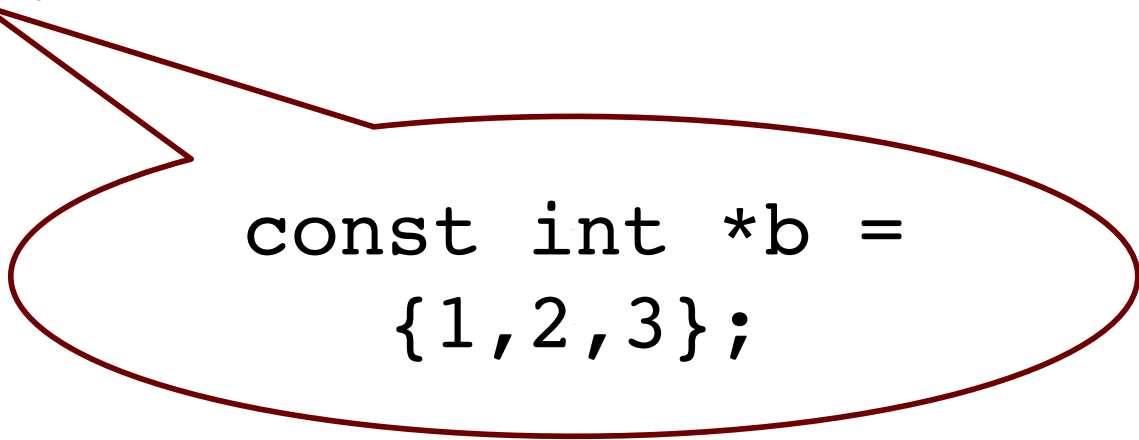
```
int *Pc, **Pd;
```

```
int a=5, b[]={1,2,3}, *Pa=&a, *Pb=b;
```

```
char *s="String";
```

- Присваивание значения

```
Pd=&Pb; Pc=&b[1];
```



```
const int *b =  
    {1,2,3};
```

Операции с указателями

```
int n, b, a[10], *pb=&a[0], *pe=&a[9], *pc;
```

■ Арифметические

```
n = pe - pb; // кол-во элементов (не байт!)
```

```
pb++; // &a[1] (адрес следующего  
// элемента но не байта!)
```

```
pc = pe - 1; // &a[8]
```

```
pb + pe; // ОШИБКА !!!
```

■ Разыменования

```
b = *pc; // b = a[8];
```

**NB: операции с указателями являются
размерными!!!**

Операции с указателями

```
struct complex { double Re; double Im; } a, *p;  
p = &a;
```

■ Разыменования

```
a.Re = 1.;
```

```
p->Im = 0.;
```

**NB: операции с указателями являются
размерными!!!**

Универсальные указатели

- `void *P, *Q;`

- Запрещены операции

- Арифметические
- Разыменования

- Типичное применение

```
P= malloc(sizeof(double)); // P = new double;  
Q= calloc(N,sizeof(int)); // Q = new(int[N]);  
free(Q); // delete(Q);  
Q= realloc(Q,M);
```

- Универсальный указатель можно

- Инициализировать значением указателя любого типа
- Преобразовать в указатель любого типа

```
#include <iostream>
using namespace std;
```

```
int main() {
    const int n= 64;
    char str[]="Hello, World!\n";
    void *p, *ps= str;
    char **pps= (char**)&ps;
    int *a, *pb, *pe;
```

```
    cout << str << ps << " " << pps << ' ' << (*pps) << endl;
    p= malloc(4*n); pb= a= (int*)p; for (int i=0; i<n; i++) a[i] = i;
    cout<< pb << "=pb\t pb+1=" << pb+1 << "\n";
    cout<< pb+2 << ": a[2]= » << *(pb+2) << endl;
    free(a); pb=a=NULL;
    cout << pb << endl;
```

```
    pb=a=new int[n]; for(int i=0; i<n; i++) *(a+i)=10*i;
    cout << a << "\t" << *(pb+2) << endl;
    pe=&a[n-1];
    cout << pe-pb << endl;
    a=(int*)realloc(a, 2*n);
    cout<<"a="<< a << "\ta[0]="<<*a<<
        "\na[n-1]="<<a[n-1]<< " a[n]="<<a[n]<<endl;
```

```
}
```

Hello, World!

```
0x7fff5fbff559 0x7fff5fbff4e0 H
0x100519920=pb pb+1=0x100519924
0x100519928: a[2]= 2
0x0
0x100519920 20
63
a=0x100519920 a[0]=0
a[n-1]=630 a[n]=64
```

Ссылки^{*)}

- Другое имя переменной (псевдоним)
- Описание и инициализация

```
int a=5, &b=a;
```

```
// a и b — разные имена одной  
// и той же области памяти
```

^{*)} ТОЛЬКО В C++

Тип FILE*

```
typedef struct _iobuf FILE;
```

Или

```
#define FILE struct _iobuf
```

```
FILE *a;
```

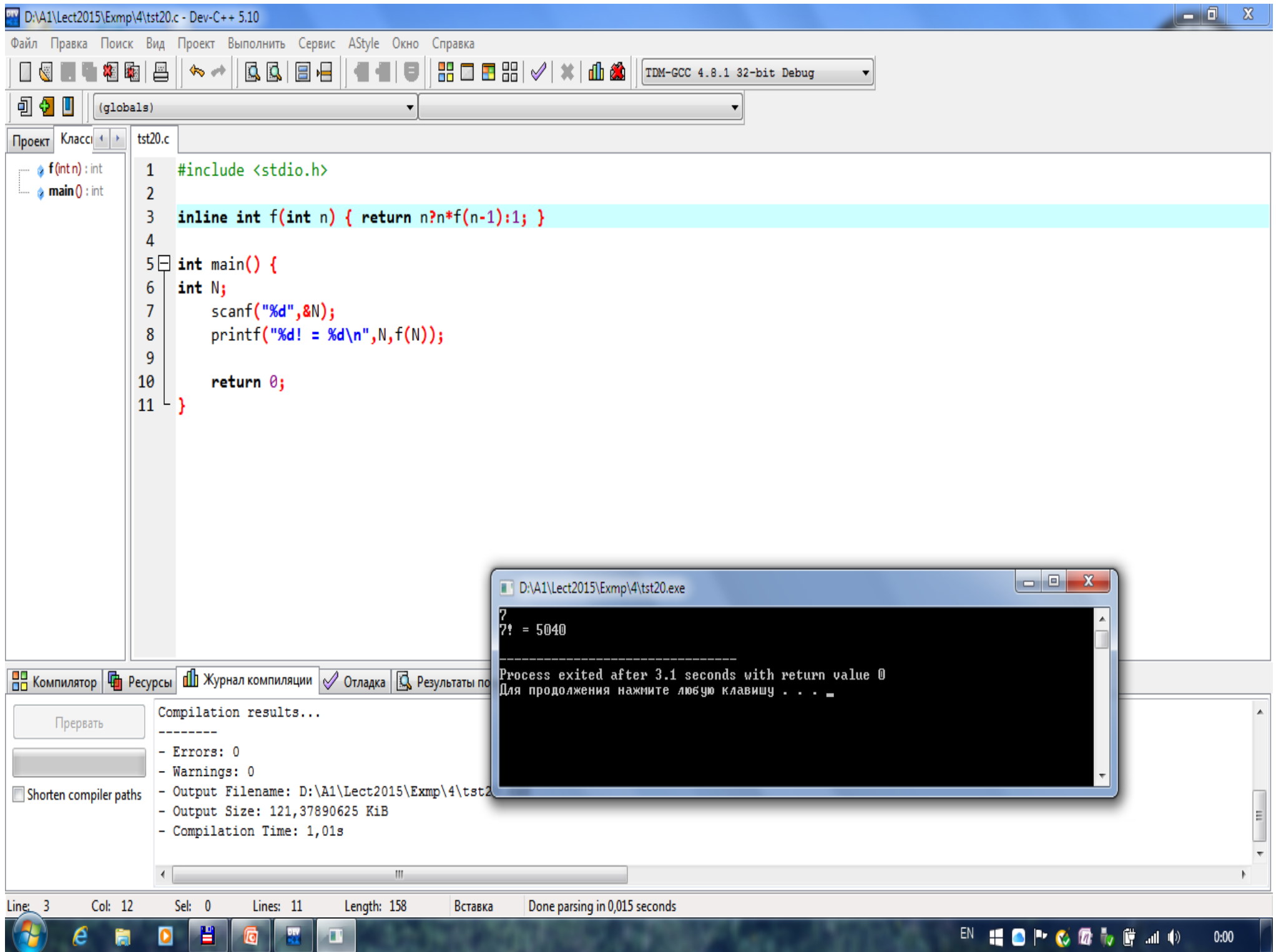
```
FILE* fopen(  
            const char* name,  
            const char* access  
            );
```

```
int fclose(FILE* a);
```

```
struct _iobuf {  
    char *_ptr;  
    int _cnt;  
    char *_base;  
    int _flag;  
    int _file;  
    int _charbuf;  
    int _bufsiz;  
    char  
*_tmpfname; };
```


Функции – конструкции действий

1. Имеют список аргументов (параметров)
2. Возвращают значение
3. Объявляются указанием своего имени и типа возвращаемого значения
4. Определяются перечислением операторов



Пример 1: решето Эратосфена

| 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |

Создание динамического массива

Заполнение признаком простоты



tst3.cpp

```
#include <stdio.h>

int* R(int);

int main() {
    int N, *a;
    scanf("%d", &N); a=R(N);
    for(int i=2; i<N; i++) if(a[i]) printf("%d ", i); printf("\n");
    delete(a);

    return 0;
}

int* R(int N) {
    int *a = new int[N], k;
    for(int i=2; i<N; i++) a[i]=1;
    for(int i=2; i<N; i++) if(a[i]) for(int j=i; (k=i*j)<N; j++) a[k]=0;
    return a;
}
```



tst3.cpp

#include <stdio.h>

int* R(int);

int main() {

int N, *a;

scanf("%d", &N); a=R(N);

for(int i=2; i<N; i++) if(a[i]) printf("%d ", i); printf("\n");

delete(a);

return 0;

}

int* R(int N)

int *a = new int[N];

for(int i=2; i<N; i++)

for(int j=i; (k=i*j)<N; j++) a[k]=0;

return a;

}

D:\A1\Lect2013\Exmp\4\tst3.exe

100

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

D:\A1\Lect2013\Exmp\4\tst3.exe

1000

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 2 | 3 | 5 | 7 | 11 | 13 | 17 | 19 | 23 | 29 | 31 | 37 | 41 | 43 | 47 | 53 | 59 | 61 | 67 | 71 | 73 | 79 | 83 | 89 | 97 | 101 | 103 | 107 | 109 | 113 | 127 | 131 | 137 | 139 | 149 | 151 | 157 | 163 | 167 | 173 | 179 | 181 | 191 | 193 | 197 | 199 | 211 | 223 | 227 | 229 | 233 | 239 | 241 | 251 | 257 | 263 | 269 | 271 | 277 | 281 | 283 | 293 | 307 | 311 | 313 | 317 | 331 | 337 | 347 | 349 | 353 | 359 | 367 | 373 | 379 | 383 | 389 | 397 | 401 | 409 | 419 | 421 | 431 | 433 | 439 | 443 | 449 | 457 | 461 | 463 | 467 | 479 | 487 | 491 | 499 | 503 | 509 | 521 | 523 | 541 | 547 | 557 | 563 | 569 | 571 | 577 | 587 | 593 | 599 | 601 | 607 | 613 | 617 | 619 | 631 | 641 | 643 | 647 | 653 | 659 | 661 | 673 | 677 | 683 | 691 | 701 | 709 | 719 | 727 | 733 | 739 | 743 | 751 | 757 | 761 | 769 | 773 | 787 | 797 | 809 | 811 | 821 | 823 | 827 | 829 | 839 | 853 | 857 | 859 | 863 | 877 | 881 | 883 | 887 | 907 | 911 | 919 | 929 | 937 | 941 | 947 | 953 | 967 | 971 | 977 | 983 | 991 | 997 |
|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|