

16 сс. $0x f. 8 p 1$ $\xrightarrow{2} \xrightarrow{dec}$ $12 \text{ et} \rightarrow 12 \text{ знаке после ,}$
 2 сс. $0001.1000 \times 2^1 = 1.1 \times 2 = 11_2 = 3_{10}$

$0x 6. 8 p -2$
 \downarrow
 $110.1000 \times$
 $\% a \leftrightarrow \% f$
 \uparrow

можно $.6 \leftarrow 6 \text{ цифр после точки}$
 $e \leftarrow \text{различность}$

Ударь, бам вопрос
 не является актуальным
 оставь его при себе
 Kill your self 😊

Массивы

`int q[10];` $\leftarrow \text{создание (там! массив)}$

`q[0] ... q[9]` $\leftarrow \text{обращение}$

`q[10]` $\leftarrow \text{до 3 знаке по джер}$

`int q[10] = {1, 2, 3};` $\leftarrow \text{инициализация (первые 3: 1, 2, 3; остальные нули)}$

$\times q = \{1, 2, 3\}$ $\leftarrow \text{не компируется}$

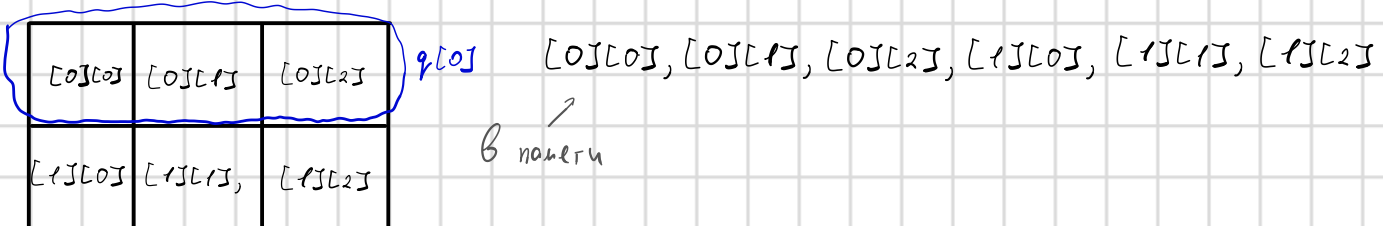
`int q[i];` $\leftarrow 1 \text{ элемент}$

$\text{sizeof}(q) / \text{sizeof}(q[0]) \leftarrow \# \text{ элементов}$

`int q[10] = { [2]=3, [4]=4 }` $\xrightarrow{\text{idx ee}} \rightarrow \{0, 4, 3, 0, \dots, 0\}$

`int q[2][3];`

! \rightarrow первый массив - непрерывная область памяти



^{конста}
int g[x][3], x-функция для константы в комп.

int w[0];
int main()
{
int g[2][3];
static int w[x];
}

VLA

Создается на стеке => Если g объявлено, то y тоже

! Статические и подобные переменные всегда инициализируются

Указатели

int a=2, b;

int *p; p-указатель, p-указывает на int

p = &a; ^{укажи адрес} (взятие адреса)

^{const} &5; ^{const} &(a+b); [✓] &a

int g[2]

p = &g[1] ✓

float f;

^{float}
p = &f;

p = &a;

*p = 1; ^{значение через указатель}

↓
a = 1

[✗]
a = p

const int c = 4;

int *p;

*p = 1 ← *должен работать* (изменил значение что)

p = &c ← *не сконвертирует тк const*

const int *p;

p = &c; ✓

*p = 1; ✗

p = &a; ✓

const int *p = &a;

int *const p = &a;

p = &b; ✗

то *p = 1; ✓

int *p, x; ⇔ int *p, x;

x - int; p - указатель

int *p;

int **z = &p;

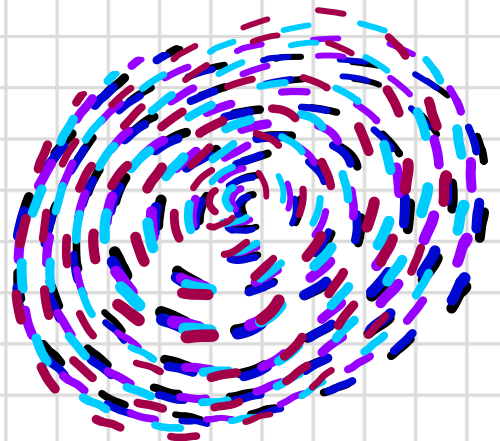
*z = &a; // p = &a

**z = 3; // a = 3

void *v;

V = p или V = z или V = fp (float *fp;)

V = &c; ✗



В C ^{зачто} $p=v$ / $z=v$ / $fp=v$

$\text{int } q[2];$

$\text{int } *p;$

$*q = 1$ // $q[0] = 1$

указатель на начало

$p = q;$ ✓ то $q = p;$ ✗

$p[0] = 2;$

$p[3] = 4;$

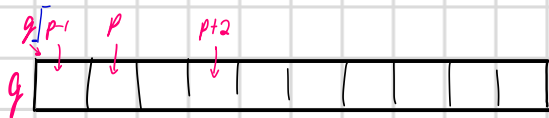
$p = \&q[1];$

$p[0] = 2;$ // $q[1] = 2$

$p[3] = 4;$ // $q[4] = 4$

$p[-1] = 2;$ // $q[0] = 2$

$p+2$ (сместить на 2 эл. вперёд)



$*(p+2) \equiv p[2]$ ✓

$p+2 \equiv 2[p] \Rightarrow 2[q]$ ✓

$p = \&*(q+1)$

$p = \&q[1];$ // $p = q+1$

$p = q$ // 1 (расстояние между указателями)

$q = p$ // -1

$\text{int } *p;$

$p = \text{malloc}(\overset{\text{даты}}{\text{sizeof(int)}} * 100);$ выделение памяти

