

Нельзя про типы

```
auto X;  
if ( )  
    x = f();  
else  
    x = g();
```

typeof-unique
(Можно же сказать typeof)
(но только для компиляции)

typeof(f()) X;
if ()
 x = f();
else:
 x = g();

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

```
typeof(p) z = &b; (полностью копирует тип)
```

```
typeof_unique(p) y; // int x (снимает модификаторы верхнего уровня)
```

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

```
typeof(*p) // const int (т.к. p - указатель)
```

⑬ В C++ есть decltype (там все не так просто)

```
int a[256], b[256], c[256]
```

```
for (size_t i=0, i < 256, i++)
```

```
{  
    c[i] = a[i] + b[i];  
}
```

размер массива размер элемента
sizeof(a) / sizeof(*a)



работает только с массивами

дугет работать доженко :)

(дугет работать по 4 байта и 64 битам)

```
void f(const int *a, const int *b, int *c)
```

```
for (size_t i=0, i < 256, i++)
```

```
{  
    c[i] = a[i] + b[i];  
}
```

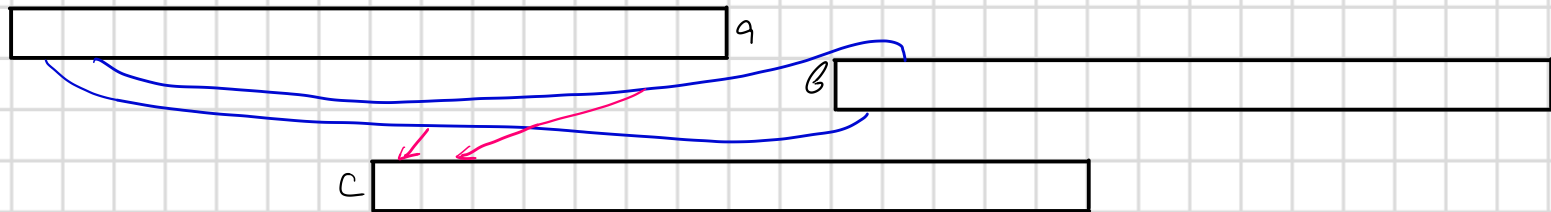
дугет нлеха :)

(Компилятор не поставит ошибки)

$\text{int } q[257], w[256]$

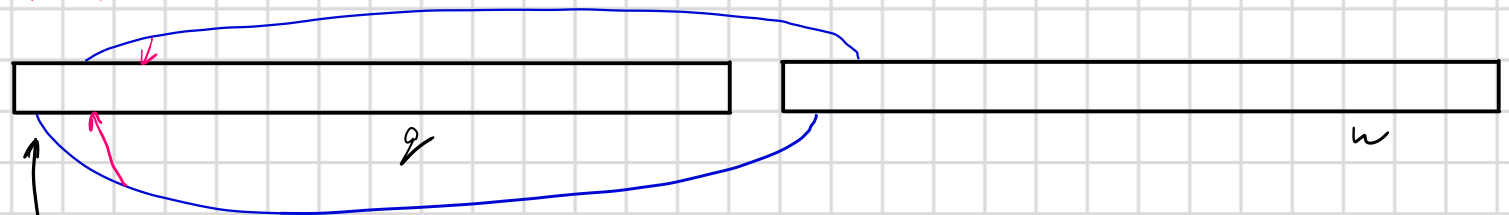
$f(w, q, q+1)$

Борзенько:



Все дез. проблем т.к. не пересекаются

Плехо:



не можем взять 4 чн. более ранее при выполнении

Как функция, зная то мы не пересекаем массивы?

`void f(const int *restrict a, const int *restrict b, int *restrict c)`

`restrict` - дает обещание компилятору и он оптимизирует (указатель не перекрывается с другими)

⑬ Если указатели пересекаются, то все равно можно `restrict`, но при условии что данные не изменятся

⑬ в C \rightarrow `restrict`

в C++ \rightarrow `__restrict`

Строки

char - тип с неизвестным знаком

Вместо можно `uint8_t` / `int8_t`

char `c = 0x30` или `'0'` ← символ
код символа '0'

явная длина

"ABC" \Leftrightarrow 'A', 'B', 'C', 0 ← хранение строки
значение (означает что строка закончилась (если же длины строки))

явная длина ↓

"ABC" \Leftrightarrow 'A', 'B', 'C' и длина строки (3) (неизвестно где хранится)

char * , но "ABC" → const char *

Как посчитать длину строки руками?

```
size_t strlen(const char *p)
{
    size_t
    for (x=0; p[x]; x++);
    return x;
}
```

1

или

```
size_t strlen(const char *p)
{
    const char *z=p
    while (*z)
        z++;
    return z-p
}
```

2

нахождение первого вхождения

```
const char *strchr(const char *p, char c)
{
    size_t
    for (x=0; p[x] != c; x++)
        if (p[x] == c: return x
    else: nullptr
}
```

Char *a...

Хотим: char *b = (char *) strchr(a, 'i');

генер. каст
возвращает const

```
int strcmp(const char *a, const char *b)
```

сравнение строк

0: равны, < 0: если лексикограф. a > b, > 0: b > a

$a \leq b \Leftrightarrow \text{strcmp}(a, b) \leq 0$

%c

Char c = 0x30

printf("%c", c); → 0

printf("%i", c); → 48 (т.к. char - unsigned int)

%s (напечатать целиком строку) (если до ноль-терминатора и печатает %c)

char *p;


printf("%s", p) и scanf("%s", p)

нет &
читает до пробельного символа

Кодировки



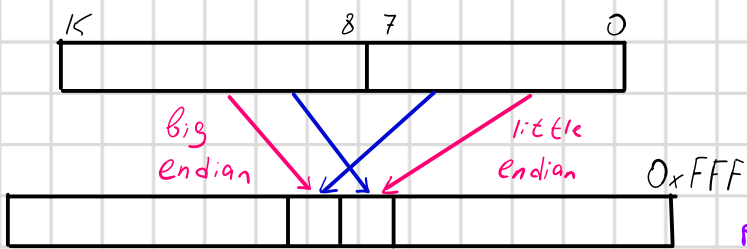
ASCII 0...127 128..255 (for drawing characters)

Code page (cp866, cp1251, KOI-8) 

DBCS (double byte character set) ^(see document 2, table) Купується с наявності 2-х місцевості

Unicode (символы кодируются 16 битами)

Unicode 2 (2 carbons) (еще 16 лет для carbons меньше, а еще 16 лет для водорода и 16 лет для кислорода)



BOY OxFEFF ❌

↪ UTF-16

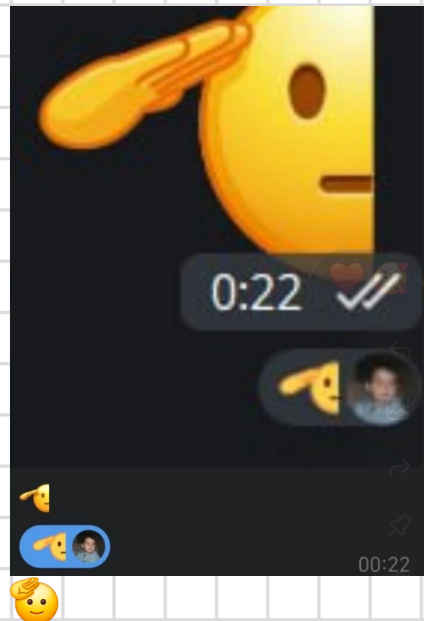
13) Sonbunhwaon meza ucanizyer little endian (unare ky xho hape cyabnon daituky)

VCS-2 ← ydoro :)

UTF-8 ← допустимо 👍

- совпадает кодами с ASCII
- на символ 1-4 байта (теперь байты а не биты)
- кодируем все языки нормально

Использовать UTF-8 и не страдать



Сборка программы

Файл a.c

```
int f(int);  
  
int main(void)  
{  
    return f(3);  
}
```

Файл b.c

```
int f(int x)  
{  
    return x+2;  
}
```

a.c → a.obj
b.c → b.obj) → my.exe

```
int main(void)  
{  
    printf("%i\n", f(3));  
    return 0;  
}
```

Линкер должен соединить printf.

Это учтено в .lib (стандартные системные ф-ии)

a.c → a.obj
b.c → b.obj
.lib) → my.exe

ссылка

gdi32.dll → MessageBox (используем системные библиотеки)

Как сделать динамическую библиотеку?

расширение: .dll (Win) .so (Unix)

можно скомпилировать {...}.c в {...}.dll

Компиляция

Компилятор:

`clang -c main.c -o main.obj -m64 -O2 -std=c23`
compile указываем имя битность сборки стандартные оптимизации стандарт

Вызов линкера:

`ld -link_/subsystem:console_/defaultlib:libcmt_main.obj_/out:main.exe_/MACHINE:x64`
битность
разряд: 32 или 64

13) Если не указать `-c` при компиляции, то линкер не найдет `main.exe` + нужно `main.exe`

! нужно изучить компилятор полностью