

Операции

$+$, $-$, $*$, $/$, $\%$

Вопросы Сложные
Типы для, не
правильно
до

Особенности

1) Только на **одинаковых**

Когда $\text{int} + \text{float} \rightarrow$ конверсия (если один вещественный, другой тоже в вещественный; еще сравнение по рангу и конверт в вещественный; при этом ранг unsigned выше; ранг меньше int переводится в int)

2) Операции с unsigned дает положительную арифметику; иначе если signed , то $\text{undefined behavior}$ (всегда код как у unsigned)

3) результат операции имеет тот же тип что и исходные операнды/операнды

4) Округление к нулю

5) операция на ноль \rightarrow $\text{undefined behavior}$



6) операция $\text{min signed type} / -1 \rightarrow 128 > \text{max signed type}$ ($\text{undefined behavior}$)

$==$; $!=$; $<$; $<=$; $>=$

Особенности

1) приведение к одному типу

Пр: $\text{uint } u = 3;$ $i < u \rightarrow \text{False}$

$\text{int } i = -1;$ i станет unsigned т.е. $2^{32} - 1$

2) результат сравнения \rightarrow int (0 или 1), но в C++ bool

$x \& \& y$; $x || y$, $!x$

③ **соединения:**

1) результат: 0 или 1

2) для $\&\&$ и $||$ bool. неопределенное поведение (иногда не угадает компилятор)

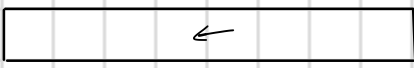
$\&$; $|$; $^$; \sim ; \ll ; \gg

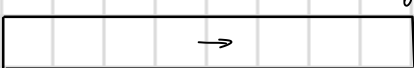
Пр:

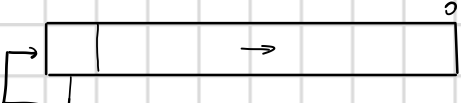
1) $3 \&\& 6 \rightarrow 1$

$3 \& 6 \rightarrow 2$

③ **Сдвиги:**


1) \leftarrow  $\leftarrow 0$ сдвиг влево

2) $0 \rightarrow$  \rightarrow логический сдвиг вправо

3)  \rightarrow арифметический сдвиг вправо

Пр:

uint $u = 3$

$u \ll$  $\#$ 32 бита не используются

③ еще одно определение

$\text{sizeof}(\text{int})$ ^{или значение} возвращает размер в байтах

$\text{sizeof}(\text{double}) \rightarrow 8$

$a = b$

$a = b = c = 0;$

$\text{if } (a = b) \leftarrow \text{бронте } \text{акотто}$

$f(3, a=4)$

$a += b \sim a = a + b$

$a++ ; ++a \leftarrow \text{всего } ++$

$a = 3$

$b = a++ = 3$

$c = ++a = 4$

$\text{if } (\dots)$

действие;

или:

\leftarrow действие 2;

$\text{if } (\dots)$

{

действие;

действие 2;

}

13) в C++ законно

$\text{if } (\text{int } x = f())$

{

$y = x$

}

и в else тоже может быть

$\text{while } (\text{ycn.})$

{

команда;

}

do

{

команда;

}

$\text{while } (\text{ycn.})$

break - выход из цикла (внутреннего)

continue -

goto метка (нужно ставить куда перейти)

```
while (усл)
  while ( )
    while ( )
      {
        if ( )
          goto метка (lend);
      }
```

метка (lend)

for (A; B; C) ~ A;

{ D; continue не пропускает C }

{ while (B) { D; C; } }

continue пропускает C

13

for (A; B; C); ← уже конец цикла

int x, y;

scanf ("%i %i", &x, &y); == 2;

Получает, и читает y, и еще только x (проверка сканера)

13 scanf возвращает кол-во прочитанных т.е. можно проверить

printf ("x+y = %i\n", x+y);

Работа с файлами:

^{конец}
FILE *f = fopen("имя", "способ");

способ = {"r", "w"} (13) где r - чтение, w - запись, b - бинарный модификатор

Пр:

FILE *fin = fopen("имя", "r");

FILE *fout = fopen("имя", "w");

^{чтение из}
fscanf(fin, "%i %i", &x, &y);

^{запись в}
fprintf(fout, "x+y = %i\n", x+y);

fclose(fin);

fclose(fout); ^{закрываю файл}

if (fin != ^{тоже файл} NULL) ← проверка на открытие

fclose(NULL) - вы не можете

Понятие файла main:

int main(int argc, char **argv)

my.exe 1 abc (argc = 3)

{
if (argc != 3) → error

fopen(argv[1], "r")

fopen(argv[2], "w")

}

