

Базовые типы данных

(fundamental)

Базовые типы

- Эквиваленты машинных типов
- Целые типы
 - char, short int, int , long int, long long int
 - Бывают знаковые и беззнаковые
 - signed unsigned
 - Дополнительно в C++
 - wchar_t, bool
- Вещественные
 - float, double, long double
- Указатели

Целочисленные типы

- Операции выполняются точно в заданном диапазоне

`#include climits`

Константы `INT_MIN` и `INT_MAX`

- Нет переполнения
- Операция деления / перегружена. Остаток от деления %

Поразрядные операции

&	Поразрядное И (and)
	Поразрядное ИЛИ (or)
^	Поразрядное исключающее ИЛИ (xor)
>>	Поразрядный сдвиг вправо
<<	Поразрядный сдвиг влево
~	Поразрядное отрицание НЕ (not)

Вывод двоичного представления

```
void dispBinary (unsigned char u) {  
    unsigned char t;  
    for (t=128; t>0; t=t>>1)  
        if ( u & t )  
            cout <<"1";  
        else  
            cout <<"0";  
    cout<<"\n";  
}
```

t – маска

1	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

u – число

1	1	1	0	0	1	0	0
---	---	---	---	---	---	---	---

u & t

1	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Умножение и деление на 2

```
int mul_2(unsigned int x){  
    return x<<1;  
}
```

```
int div_2(unsigned int x){  
    return x>>1;  
}
```

Проверка на нечетность

```
bool odd (unsigned int x){  
    return x&1;  
}
```

Симметричное кодирование

```
int code (int origin, int key) {  
    return origin^key;  
}
```

```
int main(){  
    int key = 7777;  
    int val = 23456;  
    val = code(val,key);//кодирование  
    val= code(val,key); //раскодирование  
    return 0;  
}
```

Вещественные типы

- Представление в нормализованной форме в виде порядка и мантиссы (экспоненциальной представление)
- Конечное множество представителей для бесконечного множества значений
- Неравномерная плотность представителей на числовой оси
- Вычисления выполняются приближенно
- Возможно переполнение и исчезновение порядка

Пример

- Вычислить сумму

$$\sum_{i=1}^n \frac{1}{i}$$

```
double sum_1(int n) {  
    int k=n+1;  
    double s=0;  
    for (int i=1; i<k; ++i)  
        s+= 1.0/i;  
    return s;  
}
```

```
double sum_2(int n) {  
    double s=0;  
    for (int i=n; i; --i)  
        s+= 1.0/i;  
    return s;  
}
```