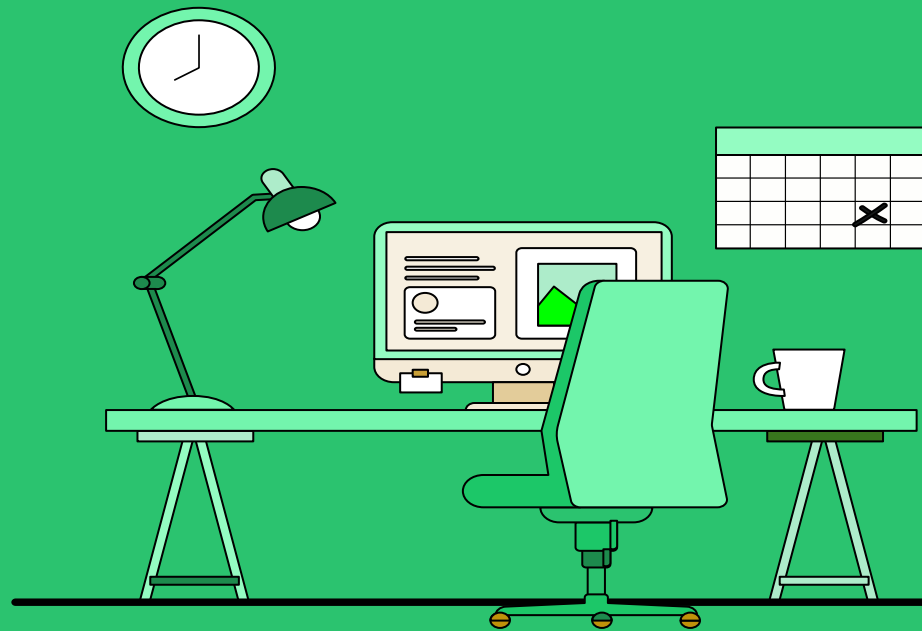


# БІБЛІОТЕКИ ВИПАДКОВІСТЬ

## C++



# Бібліотеки

LIBRARY

iostream  
cmath  
iomanip



# Підключаємо бібліотеки

```
#include <cmath>
```

```
#include <math.h>
```

## Базові функції:

abs - модуль

sin, cos, tg, ctg - тригонометричні функції

sqrt – квадратний корінь

log – логарифм

pow(x, y) –  $x^y$ - степінь

## Приклад використання математичних функцій

```
#include <iostream>
#include<cmath>
using namespace std;
int main()
{double a=1.11121314;
cout<<pow(a,3)<<" "<<sqrt(16)<<" "<<sin(a);
return 0;
}
```

### функції округлення

перекривати() округлення до більшого  
floor() округлення до меншого  
FMOD() обчислення залишку від ділення

### функції мінімуму, максимуму і різниці

fdim() позитивна різниця – C ++  
11  
Fmax() максимальне значення –  
C ++ 11  
Fmin() мінімальне значення – C  
++ 11

### Інші математичні функції бібліотеки <CMATH>

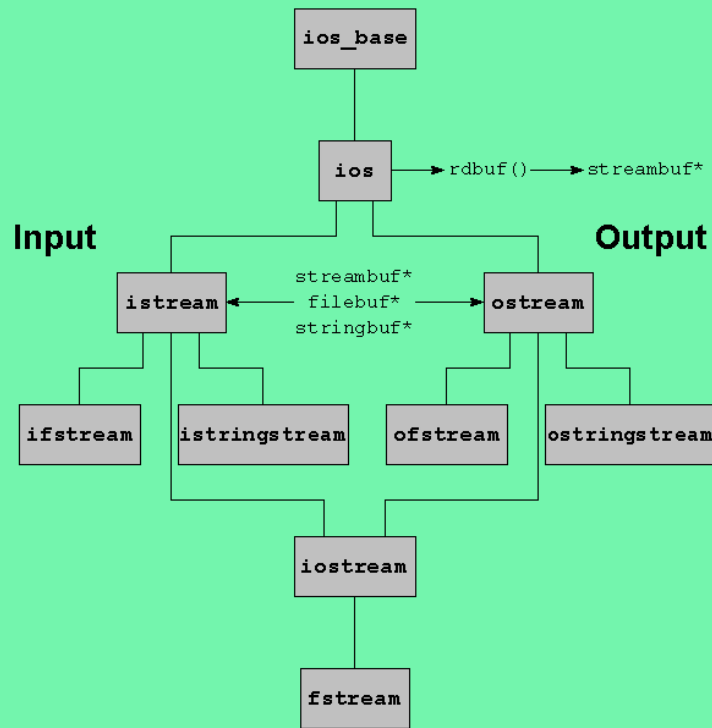
fabs() обчислення абсолютного  
значення  
abs() обчислення абсолютного  
значення  
fma() множення і додавання –  
C ++ 11

# Як написані бібліотеки?

```
00000348      bne      0x350
0000034c      b       __sqrt
00000350      fsqrt    f1,f1
00000354      blr
```

Perhaps gcc can someday be coaxed to eliminate the  
Matt suggests "use -freorder-blocks", we'll pick  
\*/

```
double sqrt(double x)
{
    if ((_get_cpu_capabilities() & kHasFsqrt))
        return __fsqrt( x );
    else return __sqrt( x );
}
```

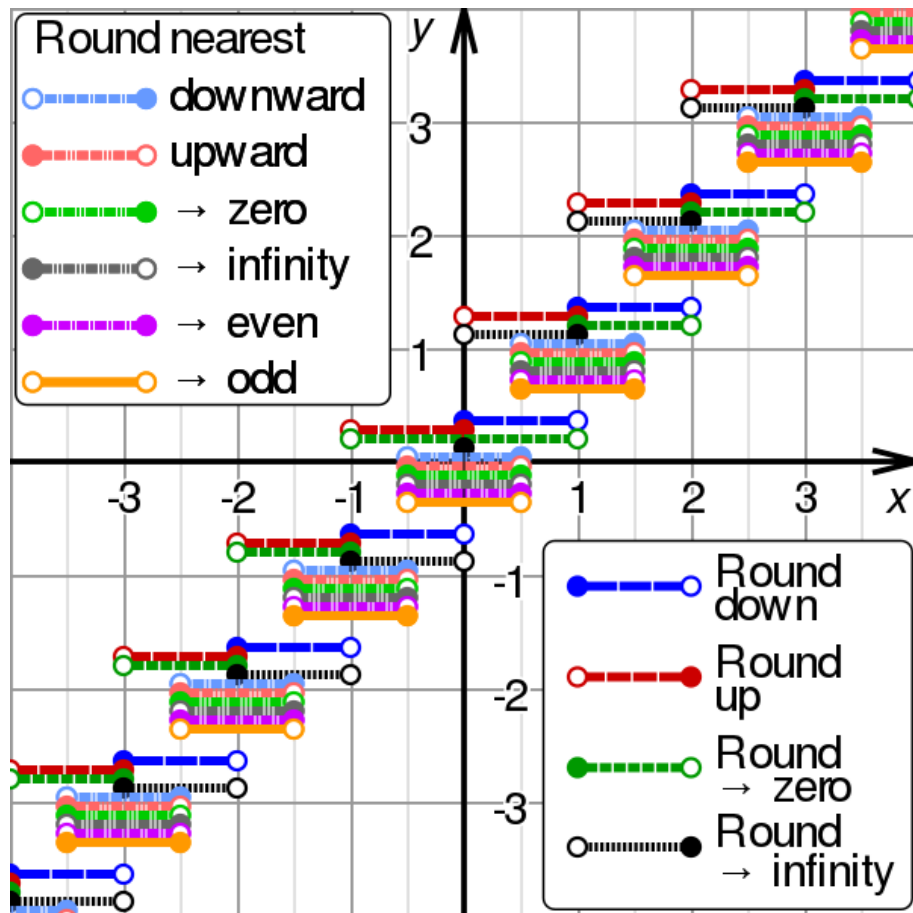


iostream

# Приклади функцій

Function	Description	Type	Example	Value
sqrt (x)	squareroot $\sqrt{x}$	double	sqrt(4.0)	2.0
pow (x, y)	power $x^y$	double	pow (2.0,3.0)	8.0
fabs (x)	absolute value for double	double	fabs(-3.5)	3.5
			fabs (3.5)	3.5
ceil (x)	ceiling (round up)	double	ceil (3.1)	4.0
			ceil (3.8)	4.0
floor (x)	floor (round down)	double	floor (3.1)	3.0
			floor (3.8)	3.0

# Таке різне округлення



## Directed rounding to an integer

- **Rounding** down.
- **Rounding** up.
- **Rounding** towards zero.
- **Rounding** away from zero.
- **Round** half up.
- **Round** half down.
- **Round** half towards zero.
- **Round** half away from zero.

# ЧАС ДЛЯ КОДУ

---

- Перевірте, як працюють функції округлення `round` та `floor`
- Знайдіть інформацію про різні види округлення та аналог таких функцій в `c++`



# Випадковість?



```
#include <iostream>
using namespace std;
int main()
{int a;
```

```
a=rand()%100;
cout<<a<<" ";
```

```
return 0;
}
```

# ЧАС ДЛЯ КОДУ

---

Завдання:

Напишіть програму, яка генерує випадкові цілі числа в діапазонах:

- 0..10
- 10..20
- -5..5

# Вгадаємо число?

*//фрагмент коду*

```
int a,b;  
srand();  
cin>>a;  
b=rand()%10;  
while(a!=b)  
{  
    if(a<b) cout<<"Enter more";  
    if(a>b) cout<<"Enter less";  
    cin>>a;  
}  
  
if(a==b) cout<<"Yes!!!";
```

## Випадковість + Цикли



# ЧАС ДЛЯ КОДУ

---

Завдання:

Напишіть програму, яка генерує випадкове ціле число від 1 до 100, а задача користувача - вгадати це число.

В код додайте порівняння введеного числа з випадковим та підказку для користувача