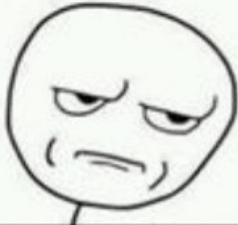


# Семинар 4 по УП

Реални числа. Математически методи. Функции

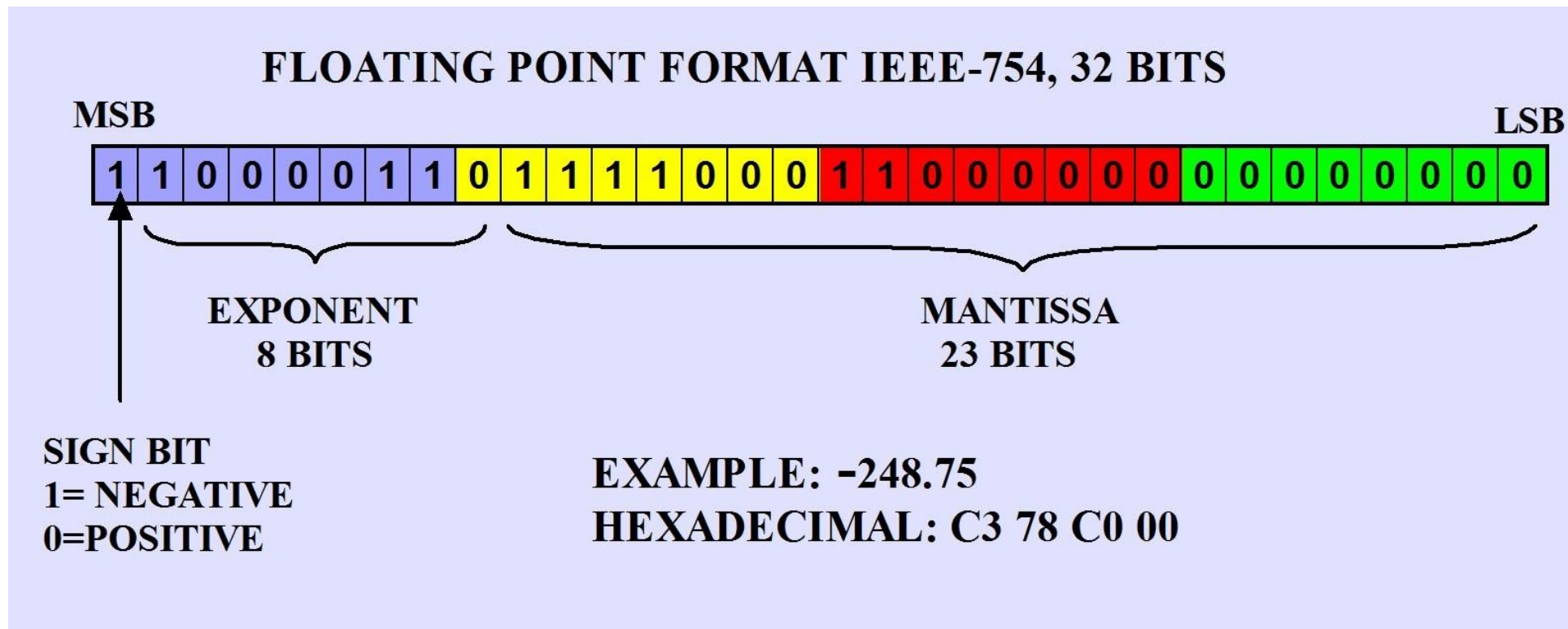
Math. The only place  
where people buy 60  
watermelons and no  
one wonders why.



# Малко за реалните числа

- Неизброимо безкрайно множество
- Представяне в паметта
- Паметта, освен че в общия случай е крайна, във всеки случай е изброима.
- Ами сега?

# Представяне на реалните числа в паметта



# Да приближения!

- Реалните числа, които нямат представяне, се закръглят до такова, което има.
- До какво води това?

# Сами видяхте, че `==` издиша.

- Приемете на доверие, че и `!=` издиша.
- Ам ся как ще сравняваме `double`.
- Най-често когато се работи с реални числа се задава точност

# Ето малко код!

```
#include <iostream>
#include <cmath>

int main()
{
    double a = 0.3333333333333333;
    double b = 1.0 / 3.0;
    const double epsilon = 0.0000000001;

    std::cout << std::boolalpha << (a == b) << std::endl;

    if(a < b + epsilon && a > b - epsilon)
    {
        std::cout << "Ravni sa";
    }
    else
    {
        std::cout << "Ne sa ravni!";
    }
    std::cout << std::endl;
}
```

# Математически методи

- ceil, floor, abs, fabs, sqrt
- Никакъв pow() ама никакъв
- sin, cos, tan, asin, acos, atan

# Задачи

- По даден правоъгълен триъгълник намерете ъглите му.
- Намерете лице на триъгълник по 3 страни



# Функции

- Аз като ви казах, че програмистите сме мързеливи, наистина го мислех.
- Ако трябва да се прави едно и също действие многократно, но с различни стойности, то просто може да си извадим действието във функция и да не пишем едно и също 100 пъти.

# Типове функции

- Можем да направим функция която връща каквато искаме стойност.
- Може и да не връща нищо.
- Може да приема аргументи, но може и не.
- Идея как да бъдете анатемосани от математик. Покажете му 3 функции: Една, която не приема аргументи, една която взима, но не връща нищо и най-голямото престъпление – такава която нито приема аргументи, нито връща стойност.

# Примери, примери ...

```
int sumOfTwo(int a, int b)
{
    return a + b;
}
```

```
double average(double a, double b)
{
    return (a + b) / 2;
}
```

```
void OutputMenu()
{
    std::cout << "Hello. Welcome to my app! Choose option: " << std::endl
               << "1 - Something" << std::endl
               << "2 - Other stuff" << std::endl
               << "3 - Hot stuff" << std::endl
               << "4 - The hottest girls are in 4th group" << std::endl;
}
```

# Какво виждаме

- Всяка функция си има тип. Този тип оказва типа стойност, която функцията връща. В скобите има списък с аргументи и техните типове.
- Малко повече. Когато дефинирате параметри в тялото на функцията. При подаване на стойности при викането се създават копия на тези променливи, които са в local scope на функцията.

# Каквооо?

```
#include <iostream>

void change(int a)
{
    a = 5;
    std::cout << "A in function is: " << a << std::endl;
}

int main()
{
    int a = 10;
    change(a);
    std::cout << "A in main is: " << a << std::endl;
}
```

# Защо вади това?

```
martin@martinLaptop:~/Desktop/kn_2023-2024_IP_Seminar/Week 4 - Real numbers, Math Functions, Functions$ ./a.out
A in function is: 5
A in main is: 10
martin@martinLaptop:~/Desktop/kn_2023-2024_IP_Seminar/Week 4 - Real numbers, Math Functions, Functions$
```

- Как да направим така че да се смени и в главната функция. Нарича се подаване по референция. Звучи сложно, но не е. Просто в дефиницията преди името се слага &.

```
void change(int &a)
```

# Глобални променливи

- Лоша практика, но съществува.
- Видима за всички функции
- Дефинира се извън тялото на функциите

# Ето така!

```
#include <iostream>

int a;
void change()
{
    a = 5;
    std::cout << "A in function is: " << a << std::endl;
}
int main()
{
    change();
    std::cout << "A in main is: " << a << std::endl;
}
```



# Задачи

- Напишете функция, която проверява дали едно число е палиндром.
- Напишете функция, която приема масив от числа и проверява дали всичките са четни
- Напишете функция, която по дадени 2 точки и два радиуса определя, дали 2 окръжности се пресичат.
- Напишете функция, която по дадени 3 точки определя дали лежат на 1 права.
- Напишете функция, която приема масив и връща дали е сортиран
- Напишете функция, която проверява дали два вектора в  $\mathbb{N}^3$  са линейно зависими
- Напишете функция, която приема едно число  $n$  и връща  $n$  тото число на Фибоначи