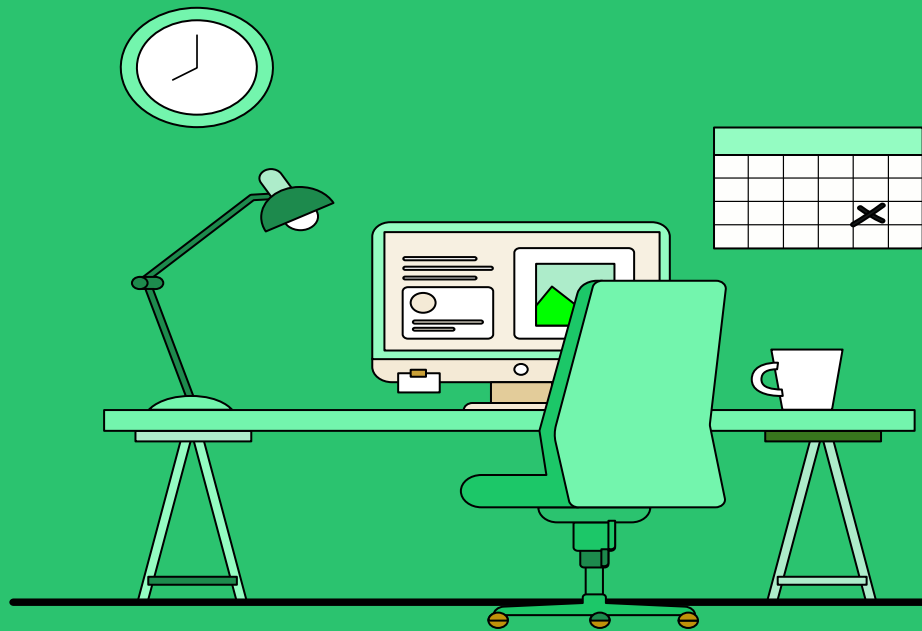
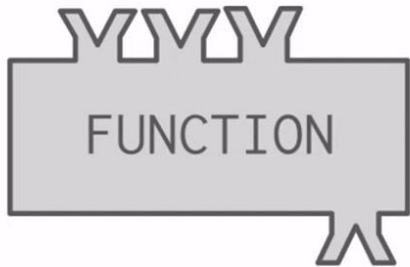


# функції

C++



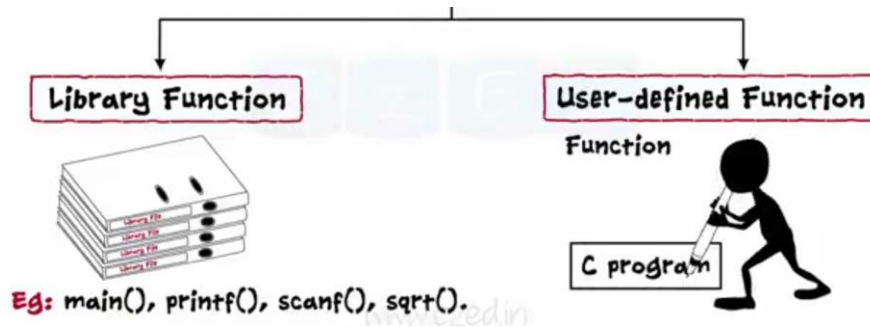
# Функції



Функції - це блоки коду, що виконують певні операції, мають власну назву

# Функції в C++

Функції дозволяють зробити програму модульною, тобто розділити програму на кілька маленьких підпрограм (функцій), які виконують частини завдання



- Функції поділяють на
- Бібліотечні
  - Користувацькі

# Функції в C++

```
#include <iostream>
using namespace std;
```

```
int main() {
    int a, b, c;
    cin >> a >> b;
    c=a+b;
    cout << c;
    return 0;
}
```

```
#include <iostream>
using namespace std;
int fout(int c)
{cout <<c;
return 0;
}
```

```
int main() {
    int a, b, c;
    cin >> a >> b;
    c=a+b;
    fout(c);
    return 0;
}
```

# ЧАС ДЛЯ КОДУ

---

- Протестуйте код з попереднього слайду
- Дopiшіть виклик функції, яка виводить “The end”

# Виклик функцій за умовою

```
#include <iostream>
using namespace std;

int cats1()
{cout<<"Чорний котик";
return 0;
}

int main(){
    int x;

    cin >> x;
    if (x<5) cats1();
    else cats2();
    return 0;
}
```

## Умови

- Коротка (if)
- Повна (if - else)
- Складена (декілька умов)

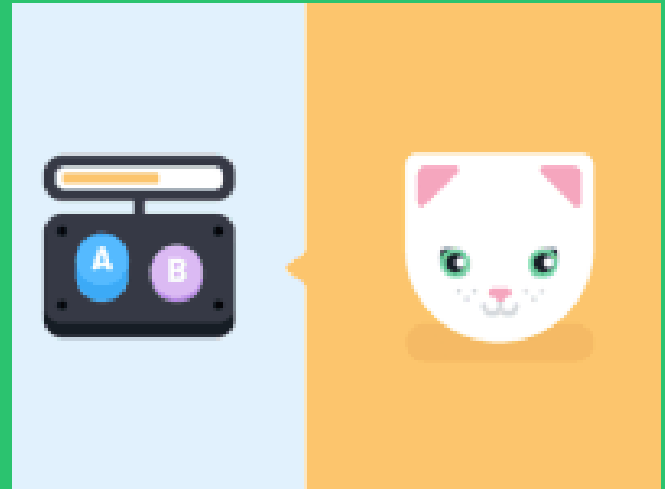
```
if(умова) дія ;
if(умова) дія 1; дія 2;
```



# ЧАС ДЛЯ КОДУ

---

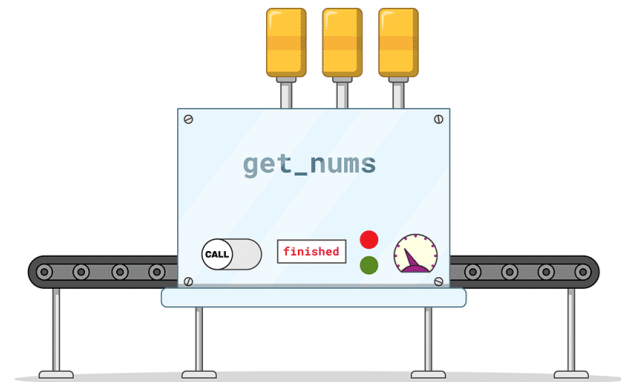
Напишіть код з двома функціями, які викликаються за певною умовою



# Виклик функцій в циклі

```
#include <iostream>
using namespace std;
int get_nums()
{cout<<"Я працюю";
return 0;
}
```

```
int main() {
    int x;
    cin >> x;
    while (x<5) {
        get_nums();
        x++; }
    return 0;
```



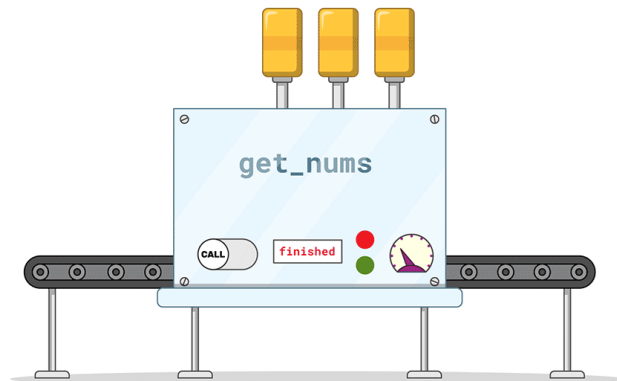


# Передаємо дані в функції

```
#include <iostream>
using namespace std;

int get_nums(int x)
{cout<<"Я працюю"<<x<<endl;
  return 0;
}

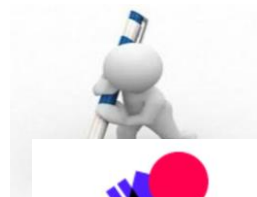
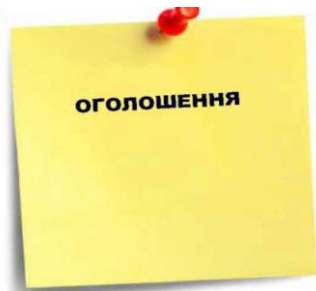
int main(){
    int x;
    cin >> x;
    while (x<5) {
        get_nums(x);
        x++; }
    return 0;
}
```



# Оголошення, опис та виклик функції

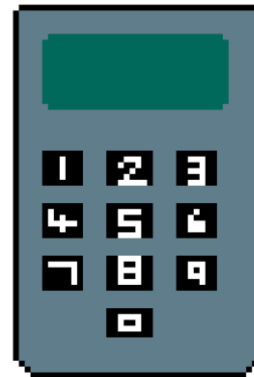
```
#include <iostream>
using namespace std;
int get_nums(int x)
{cout<<"Я працюю"<<x<<endl;
return 0;
}
```

```
int main(){
    int x;
    cin >> x;
    while (x<5) {
        get_nums(x);
        x++; }
    return 0;
}
```

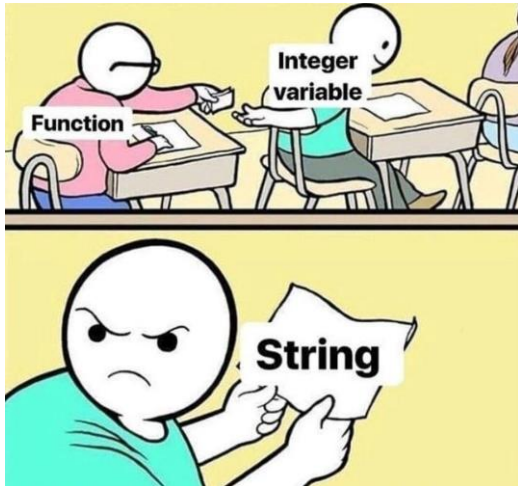


# Калькулятор

```
int  sum( int a, int b) {  
    cout<<a+b);  
return 0;  
}  
int main ()  
{int a,b;  
cin>>a>>b;  
sum(a,b);  
return 0;}
```



# Параметри функції



```
#include <iostream>
using namespace std;
    int sum(int a, int b)
    {
        return a+b;
    }
```

```
int main(){
    int x, y;
    cin >> x>>y;
    cout<<sum(x,y) ;
    return 0;
}
```

# ЧАС ДЛЯ КОДУ

---

Перевірте передачу в функції  
даних різного типу



## Прототип функції

a

b

```
#include <iostream>
using namespace std;
```

```
int sum(int, int);
```

```
int main() {
```

```
    int x, y;
```

```
    cin >> x >> y;
```

```
    cout << sum(x, y);
```

```
    return 0;
```

```
}
```

```
int sum(int a, int b)
```

```
{
```

```
    return a+b;
```

```
}
```

# Що таке return?

Вихід з  
функції

Повернення  
значення



```
#include <iostream>
using namespace std;

int mul(int a, int b)
{
    return a*b;
}

int main()
{
    int x=10, y=20, z;
    z = mul(x,y);
    cout<< z << endl;
    return 0;
}
```

# ЧАС ДЛЯ КОДУ

---

Напишемо код з  
прототипованою функцією та  
без прототипу