

write function by raising 2 to the power n

- 1) using a loop
- 2) using recursion

2^n

```
power_loop(number)//number=5
{
    //2^5=32
}
```

```
power_recursion(number)//number=5
{
    //2^5=32
}
```

```
void loopPower(int number) {
    int powerNum = 1;
    for (int i = 1; i <= number; i++) {
        powerNum = powerNum * 2;
    }
    std::cout << powerNum << std::endl;
}
```

break
continue
switch
for



```
int recursePower(int number) {
    if (number >= 1) {
        return 2 * recursePower(number - 1);
    }
    else {
        return 1;
    }
}
```

recursePower (5)=2*recursePower (4)=2*2*recursePower (3)=
=2*2*2*recursePower (2)=2*2*2*2*recursePower (1)
=2*2*2*2*2*recursePower (0)=2*2*2*2*2*1

```
example(int number)
{
    if(number==2)
    for()
    {
        for()
    }
    if(number==3)
    for()
    {
        for()
        {
            for()
        }
    }
}
```

MFTI

MIPT

$((34*5!)^{-5}+6)/8$