factorial

```
1 int factorial(int n)
2 {
3     if(n<2)
4     return 1;
5
6     return n*factorial(n-1);
7 }</pre>
```

factorial with ternary operator

```
1 int factorial(int n)
2 {
3    return (n<2) ? 1 : n*factorial(n-1);
4 }</pre>
```

ternary ans = a & b : c

```
1 if (a)
2 ans=b;
3 else
4 ans=c;
```

factorial with tail recursion

```
int factorial_r(int n, int big_n)
     if (n < 2)
       return big_n;
5
6
     return factorial_r(n-1,n*big_n);
8
   int factorial(int n)
10
     return factorial_r(n,1);
12
```

binary search with recursion 1

```
1 int search(int a[], int n, int val)
2 {
3    return find_r(a,n,val,0,n-1);
4 }
```

binary search with recursion 2

```
int find_r(int a[], int n, int val, int low, int high)
6
7
8
      if ( high < low )</pre>
9
        return -1:
10
11
      int mid = (high + low)/2;
12
13
      if (a[mid]==val)
14
        return mid:
15
      if (val>a[mid])
16
        return find_r(a,n,val,mid+1,high);
17
18
      return find_r(a,n,val,low,mid-1);
19
20
```