**Utility functions**

**head()**

function retHead(arr)

{

    return arr[0];

}

**tail()**

let arr=[1,2,3,4];

function retTail(arr,beg=0,retArr=[])

{

    if(beg==0)

        return retTail(arr,1,retArr);

    else if(beg+1>arr.length)

        return retArr;

    else

    {

        retArr[beg-1]=arr[beg];

        return retTail(arr,beg+1,retArr);

    }

}

console.log(retTail(arr));

**map()**

let arr=[1,2,3];

function map(arr,func,i=0,ret=[])

{

    if(arr[i]==undefined)

        return ret;

    else

    {

        ret[i]=func(arr[i],i,arr);

        return map(arr,func,i+1,ret);

    }

}

console.log(map(arr,a=>a\*a));

**filter()**

let arr=[1,2,3]

function filter(arr,func,i=0,ret=[])

{

    if(arr[i]==undefined)

        return ret;

    else

    {

        if(func(arr[i],i,arr))

            ret[ret.length]=arr[i];

        return filter(arr,func,i+1,ret);

    }

}

//console.log(arr);

console.log(filter(arr,a=>a%2==0));

**reduce()**

let arr=[1,2,3,4]

function reduce(arr,func,init=0,i=0,ret=0)

{

    if(init!=0 && i==0)

        ret=init;

    if(arr[i]==undefined)

        return ret;

    else

    {

        ret=func(ret,arr[i],i,arr);

        return reduce(arr,func,init,i+1,ret);

    }

}

console.log(reduce(arr,(a,b)=>a+b));

**max()**

arr=[4,1,8,6,3]

function calMax(arr,i=0,max=Number.MIN\_VALUE)

{

    if(i==arr.length)

        return max;

    else

    {

        if(arr[i]>max)

            max=arr[i];

        return calMax(arr,i+1,max);

    }

}

console.log(calMax(arr));

**min()**

arr=[4,1,8,6,3,-3]

function calMin(arr,i=0,min=Number.MAX\_VALUE)

{

    if(i==arr.length)

        return min;

    else

    {

        if(arr[i]<min)

            min=arr[i];

        return calMin(arr,i+1,min);

    }

}

console.log(calMin(arr));

**reverse()**

let arr=[1,3,6,5,2]

function reverse(arr,i=0,rev=[])

{

    if(i==arr.length)

        return rev;

    else

    {

        rev=reverse(arr,i+1,rev);

        rev[rev.length]=arr[i];

        return rev;

    }

}

console.log(reverse(arr));

**last()**

let arr=[1,2,3,5];

function retLast(arr,i=0)

{

    if(arr[i+1]==undefined)

        return arr[i];

    else

        return retLast(arr,i+1);

}

console.log(retLast(arr));

**init()**

let arr=[1,2,3,4,5];

function retInit(arr,i=0,ret=[])

{

    if(arr[i+1]==undefined)

        return ret;

    else

    {

        ret[i]=arr[i];

        return retInit(arr,i+1,ret);

    }

}

console.log(retInit(arr));

**length()**

let arr=[1,2,3,4,5,6]

function calLength(arr,i=0)

{

    if(arr[i]==undefined)

        return 0;

    else

        return 1+calLength(arr,i+1);

}

console.log(calLength(arr));

**take()**

let arr=[1,2,3,4,5];

function take(arr,limit,i=0,ret=[])

{

    if(i==limit)

        return ret;

    else

    {

        ret[i]=arr[i];

        return take(arr,limit,i+1,ret);

    }

}

console.log(take(arr,5));

**drop()**

let arr=[1,2,3,4,5,6];

function drop(arr,limit,i=0,ret=[])

{

    if(arr[i]==undefined)

        return ret;

    else if(i+1<=limit)

        return drop(arr,limit,i+1,ret);

    else

    {

        ret[i-limit]=arr[i];

        return drop(arr,limit,i+1,ret);

    }

}

console.log(drop(arr,6));

**contains()**

let arr=[1,5,3,7]

function contains(arr,element,i=0)

{

    if(arr[i]==undefined)

        return false;

    else if(arr[i]==element)

        return true;

    else

        return contains(arr,element,i+1);

}

console.log(contains(arr,2));

**cycle()**

function cycle(arr)

{

    return {

        i:0,

        next: function()

        {

            return arr[(this.i++)%arr.length];

        }

    }

}

let x=cycle([1,2,3,4]);

console.log(x.next());

console.log(x.next());

console.log(x.next());

console.log(x.next());

console.log(x.next());

console.log(x.next());

**range()**

function range(start,end,skip=1,ret=[],i=0)

{

    if(skip>0)

    {

        if(i==0 && start<end)

        {

            ret[i]=start;

            return range(start,end,skip,ret,i+1);

        }

        else if(ret[i-1]+skip>=end || start>end)

            return ret;

        else

        {

            ret[i]=ret[i-1]+skip;

            return range(start,end,skip,ret,i+1);

        }

    }

    else

    {

        if(i==0 && start>end)

        {

            ret[i]=start;

            return range(start,end,skip,ret,i+1);

        }

        else if(ret[i-1]+skip<=end || start<end)

            return ret;

        else

        {

            ret[i]=ret[i-1]+skip;

            return range(start,end,skip,ret,i+1);

        }

    }

}

console.log(range(20,10,-1));

**sum()**

function sum(arr)

{

    return arr.reduce((a,b)=>a+b,0);

}

console.log(sum(range(1,4)));

**sumOfSquares()**

function sumOfSquares(arr)

{

    return arr.reduce((a,b)=>a+b\*b,0);

}

console.log(sumOfSquares(range(0,4)));