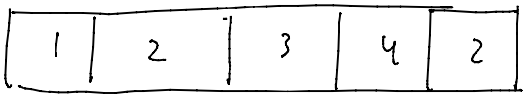
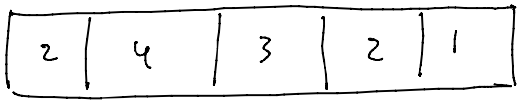
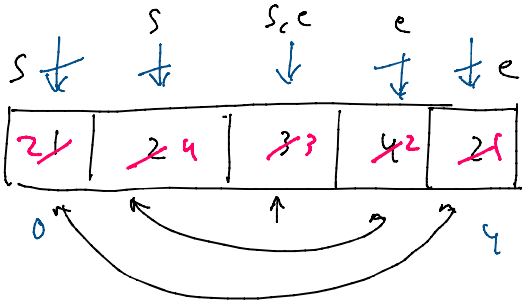
① Reverse an Array





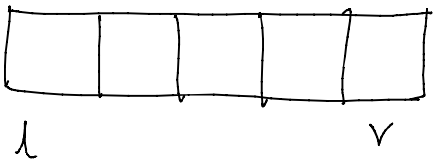
Naive Approach



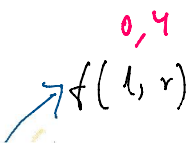
for loop/while

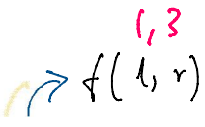
```
s = 0
e = n-1
swap (s, e)
s++
e--
```

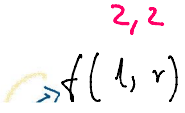
Recursion Using Two Pointer :

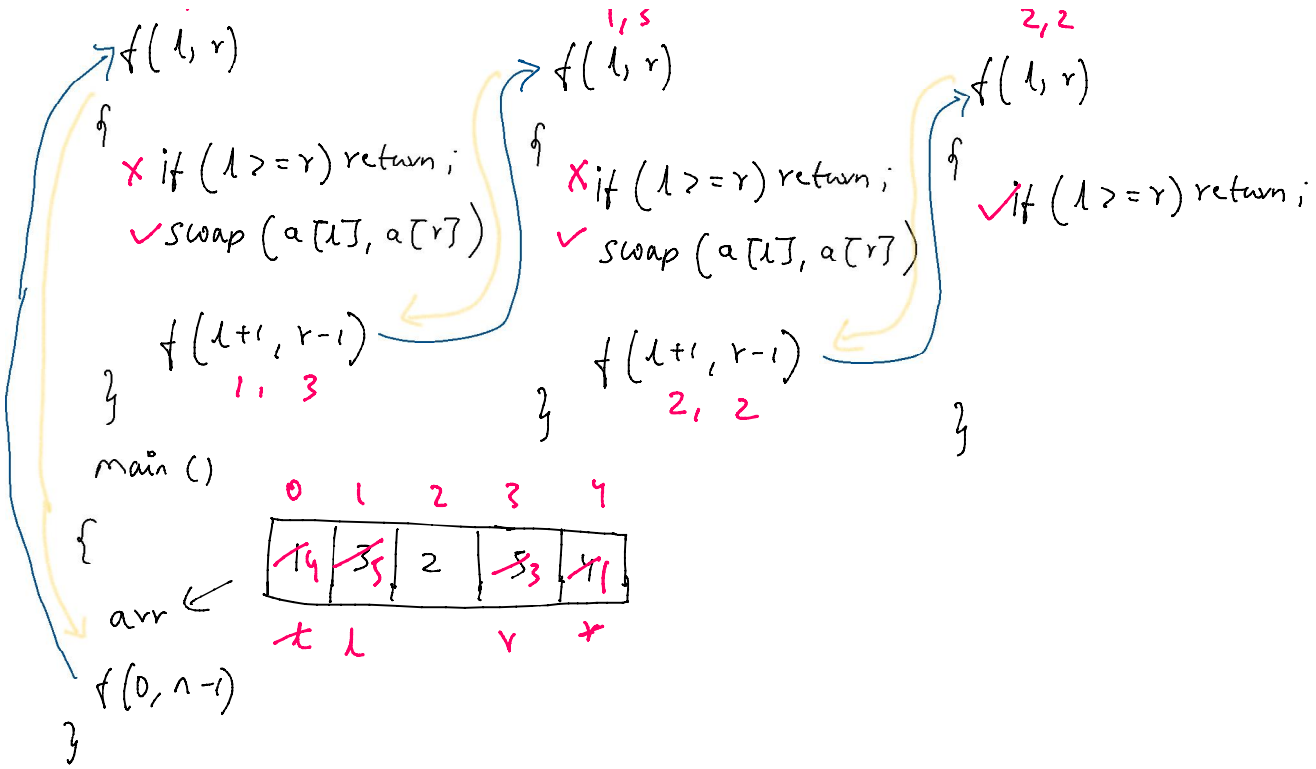


⇒ Recursion job is to swap · and l++ and r--

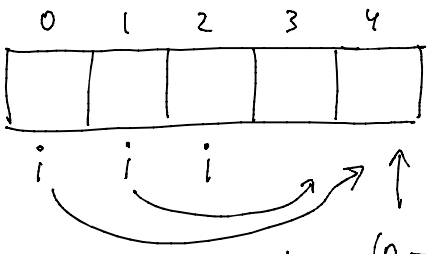








Can you solve this using single pointer?



n = 5

(n - i - 1)

(5 - 0 - 1) = 4

(n-i-1)

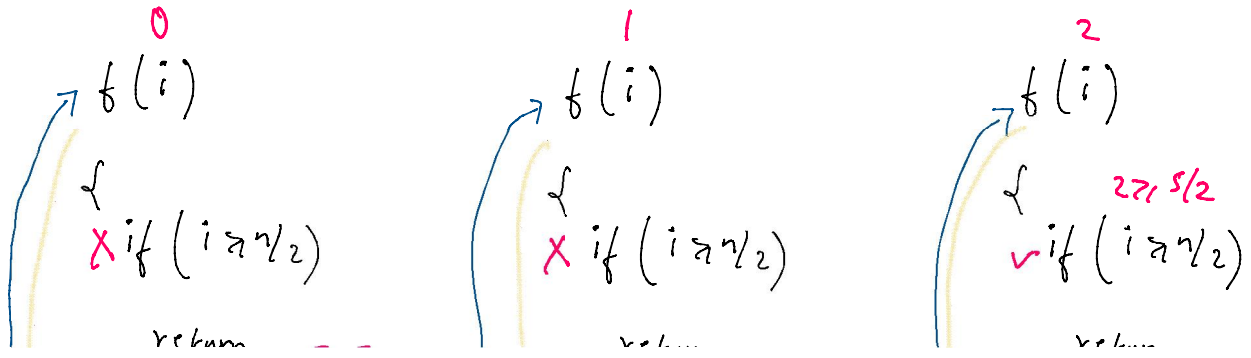
5 - 1 - 1

= 3

swap (i, n - i - 1)

till how long?

(i >= n/2) untill middle



if (i >= n/2) if (i >= n/2) if (i >= n/2)

return return return

a[0] , a[4] a[1] , a[3] X

swap(a[i], a[n-i-1]); swap(a[i], a[n-i-1]); swap(a[i], a[n-i-1]);

f(i+1); f(i+1); f(i+1);

} } }

main(

{

arr;

f(0);

}

```
#include<bits/stdc++.h>
using namespace std;
void reverse(int i, int arr[], int n){
    //base
    if(i >= n/2) return;
    swap(arr[i], arr[n-i-1]);
    reverse(i+1, arr, n);
}

int main(){
    int n;
    cin>>n;
    int arr[n];
    for(int i = 0; i<n; i++)
        cin >> arr[i];
    reverse(0, arr, n);
    for(int i =0; i<n; i++)
        cout << arr[i] << " ";
    return 0;
}
```

Input:

```
5
1 2 3 4 5
```

output.in:

```
5 4 3 2 1
```

② Palindrome :

Check if a string is palindrome or not

"MADAM" —reverse→ "MADAM" (true)

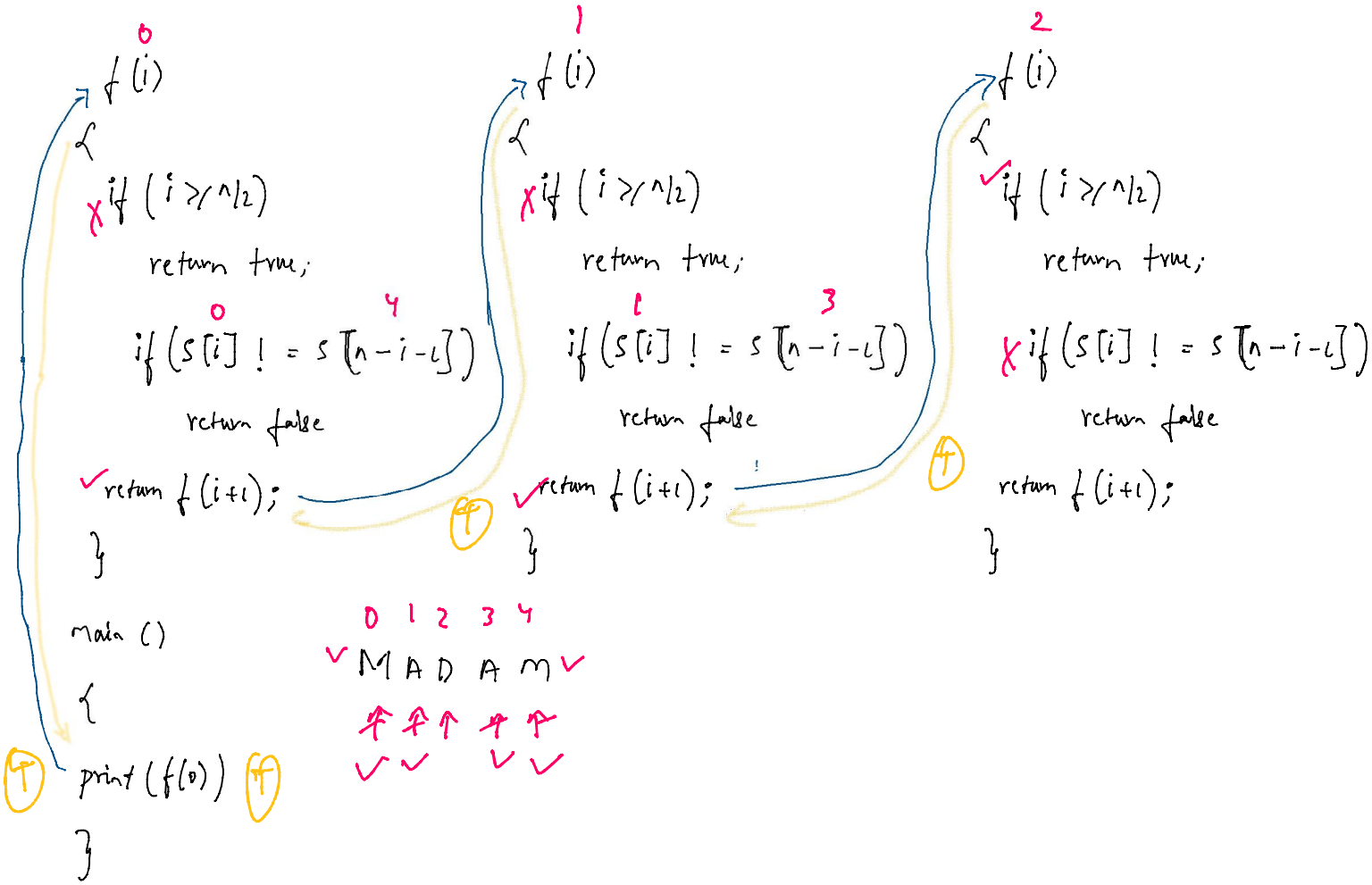
"11211" —reverse→ "11211" (true)

"123" —reverse→ "321" (false)

* It's same like reversing, the left half and the right half

Should be same.

M A D A M




```cpp
#include<bits/stdc++.h>
using namespace std;
bool f(int i, string &s) {
    if(i >= s.size() / 2) return true;
    if(s[i] != s[s.size() - i - 1]) return false;
    return f(i+1, s);
}
int main() {
    #ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
    #endif
    string s = "madam";
    cout << f(0, s);
    return 0;
}
```

T.C ⇒ O(n/2) (Till half you are going

S.C ⇒ O(n/2)