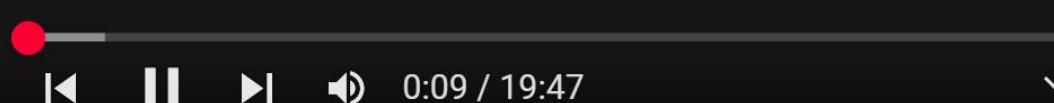


## Re 4. Problems on Functional Recursion | Strivers A2Z DSA Course



TUF



## Recursion

→ Reverse a word



TUF

## Recursion

→ Reverse an array

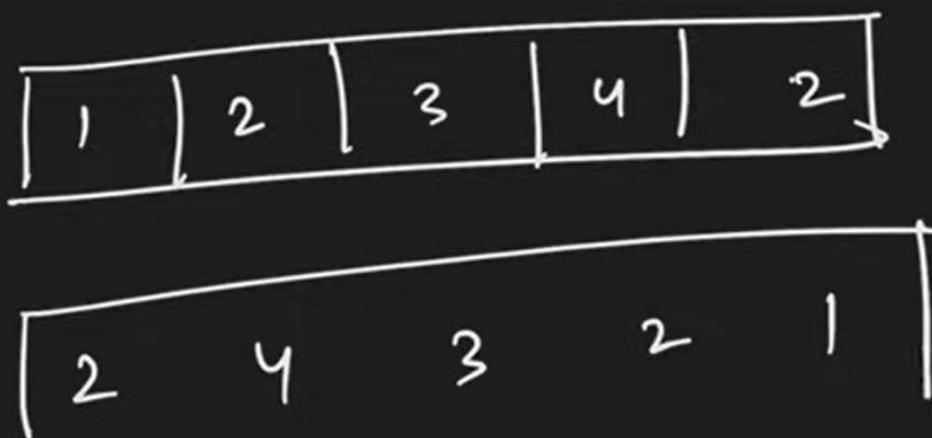
1    2



TUF

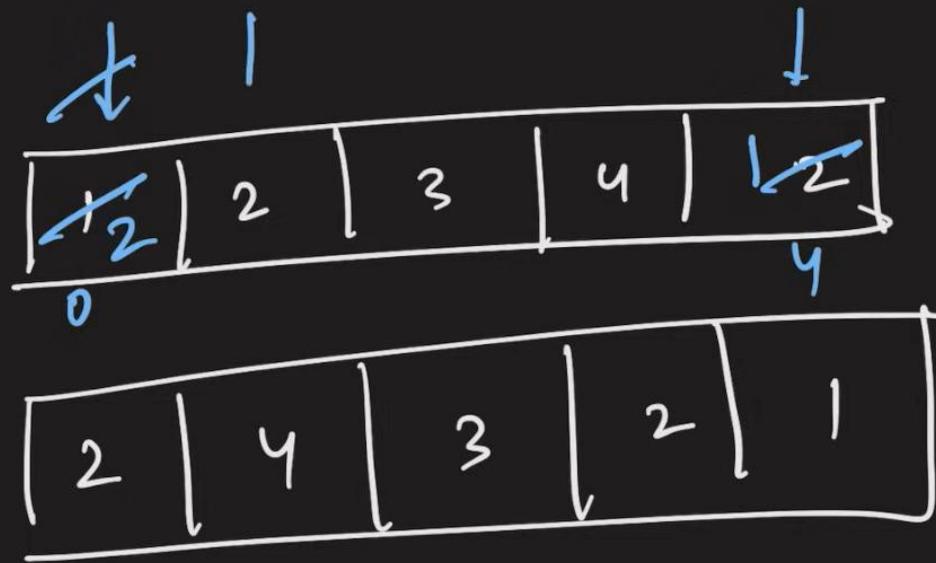
## Recursion

→ Reverse an array



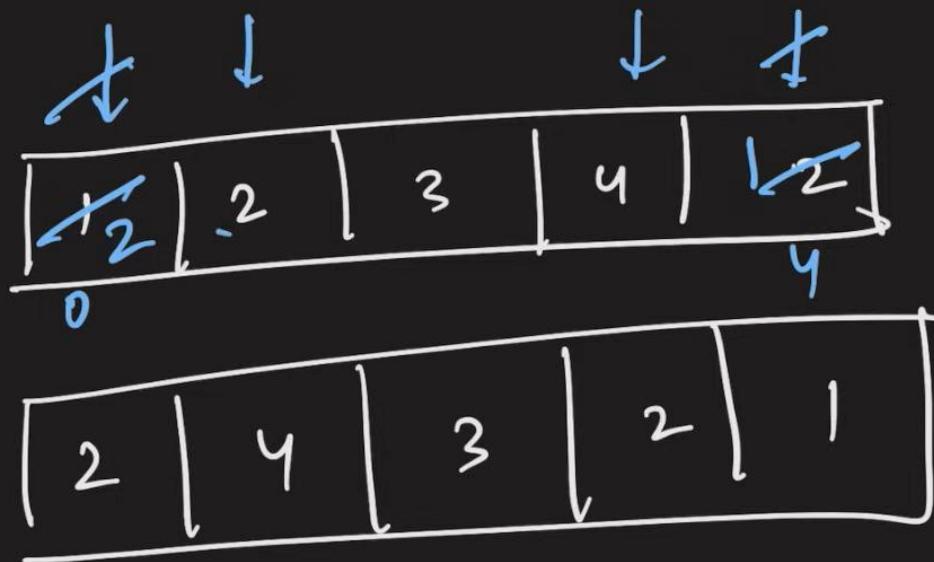
## Recursion

→ Reverse an array



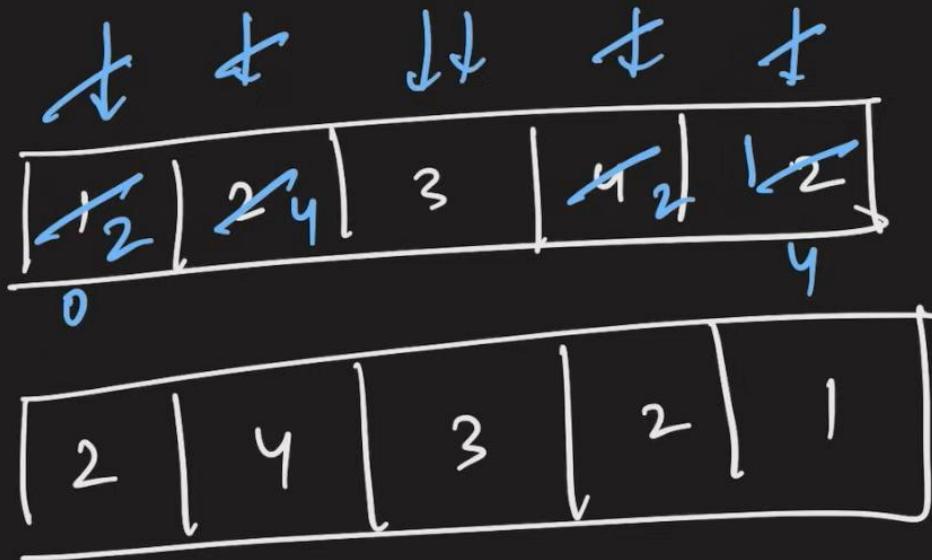
## Recursion

→ Reverse an array



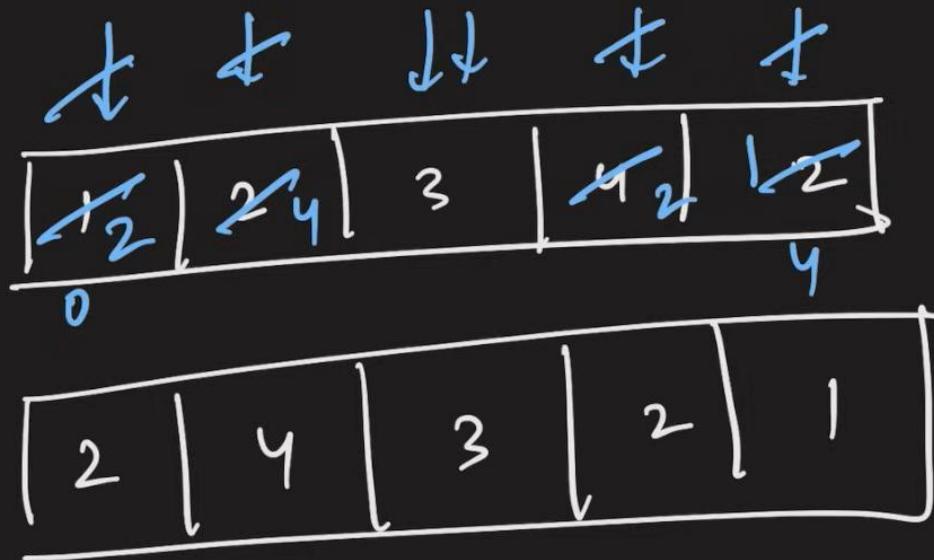
## Recursion

→ Reverse an array



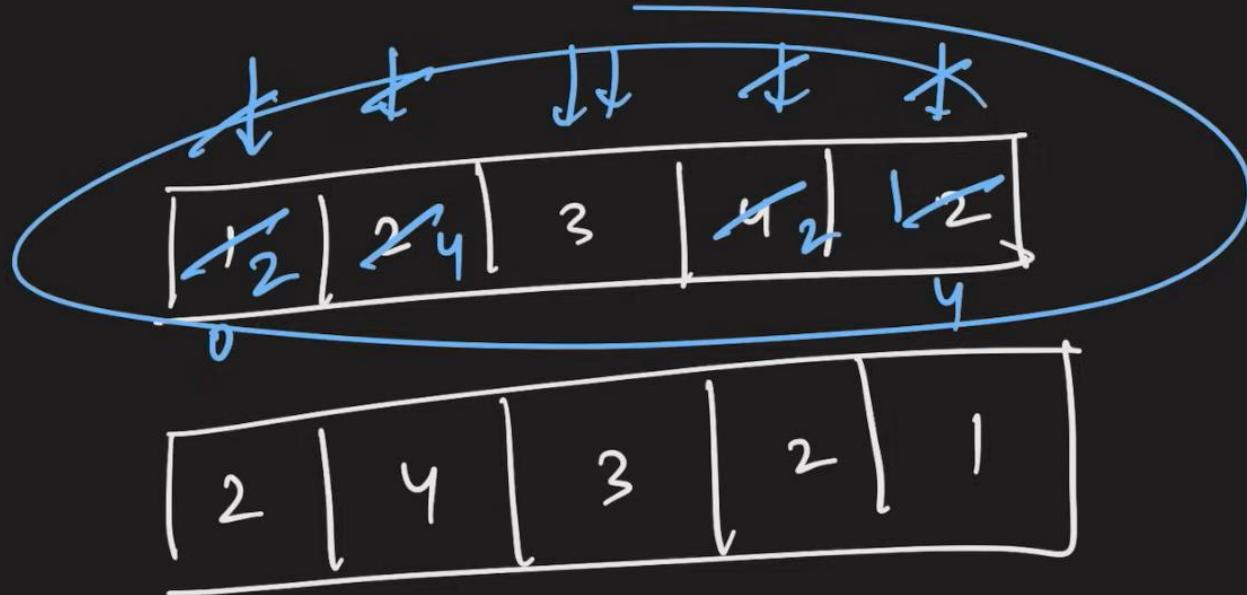
## Recursion

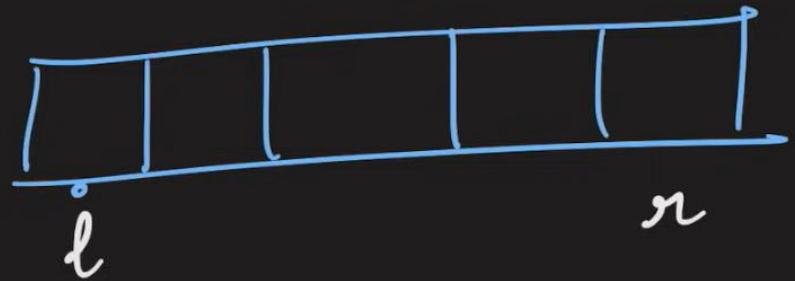
→ Reverse an array



## Recursion

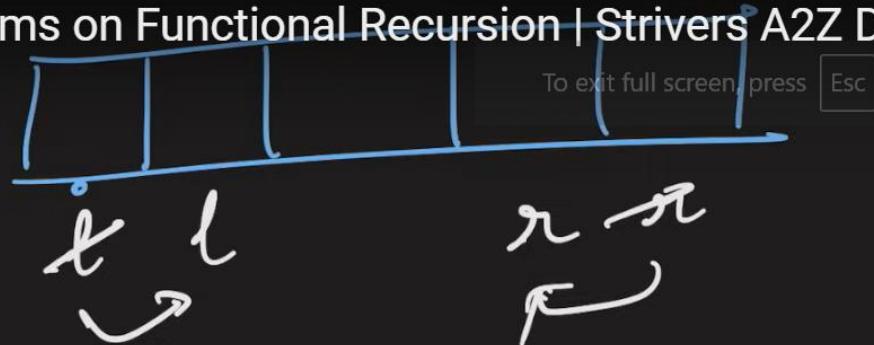
→ Reverse an array





TUF

## ≡ Re 4. Problems on Functional Recursion | Strivers A2Z DSA Course

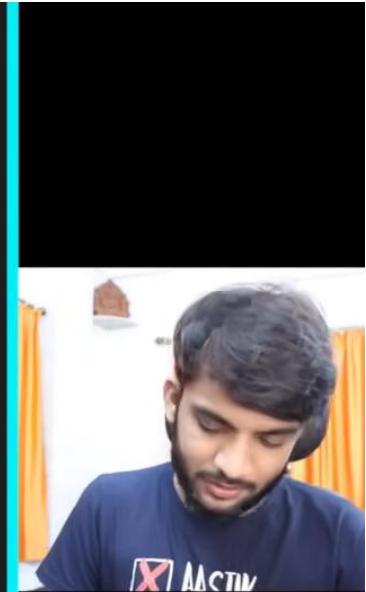


TUF



$f(l, r)$

`swap(a[l], a[r]):`



TUF



$f(l, r)$

`swap(a[l], a[r]);`

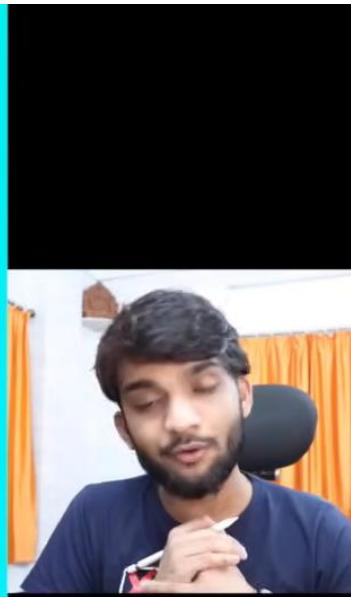
$f(l+1, r-1);$

}



↑      ↘

```
f(l, r)
{
    if (l >= r) return;
    swap(a[l], a[r]);
    f(l+1, r-1);
}
```



```
f(l, r)
{
    if (l >= r) return;
    swap(a[l], a[r]);
    f(l+1, r-1);
}
```



```
f(l, r)
{
    if (l >= r) return;
    swap(a[l], a[r]);
    f(l+1, r-1);
}

main()
{
    arr ↴
    f(0, n-1);
}
```



$f(l, r)$

{  
if ( $l \geq n$ ) return;

swap ( $a[l], a[n]$ );

$f(l+1, r-1)$ ;

}

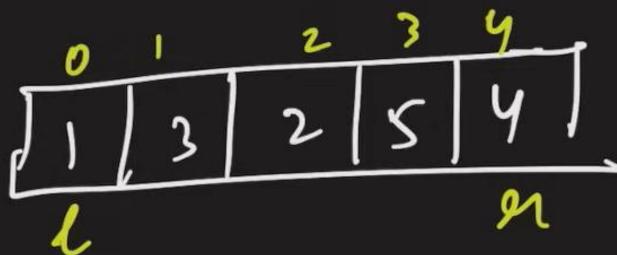
main()

{

arr ↴

$f(0, n-1)$ ;

}



TUF



```

→ f(l, r)
{
    → if (l >= n) return ; x
    → swap(a[l], a[r]);
    → f(l+1, r-1);
}
main()
{
    arr ↴
    f(0, n-1);
}

```

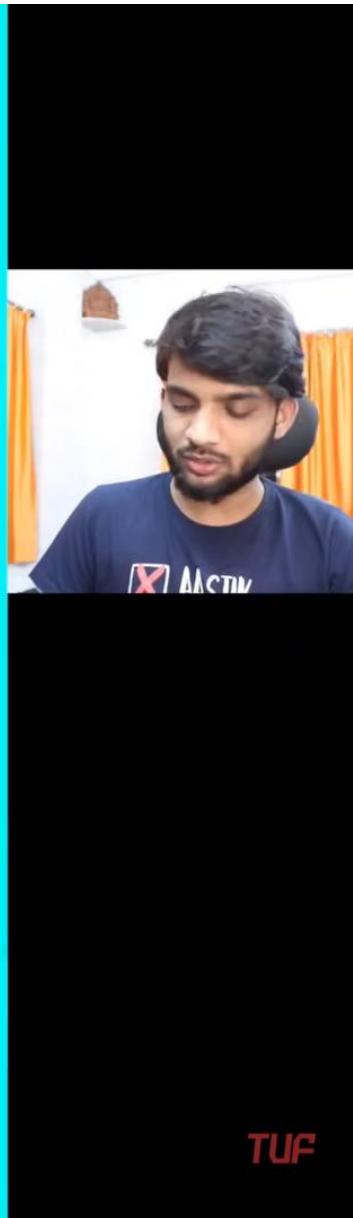
swap ✓

f(2, 2)

>

0	1	2	3	4
4	3	2	5	9

l      r ←

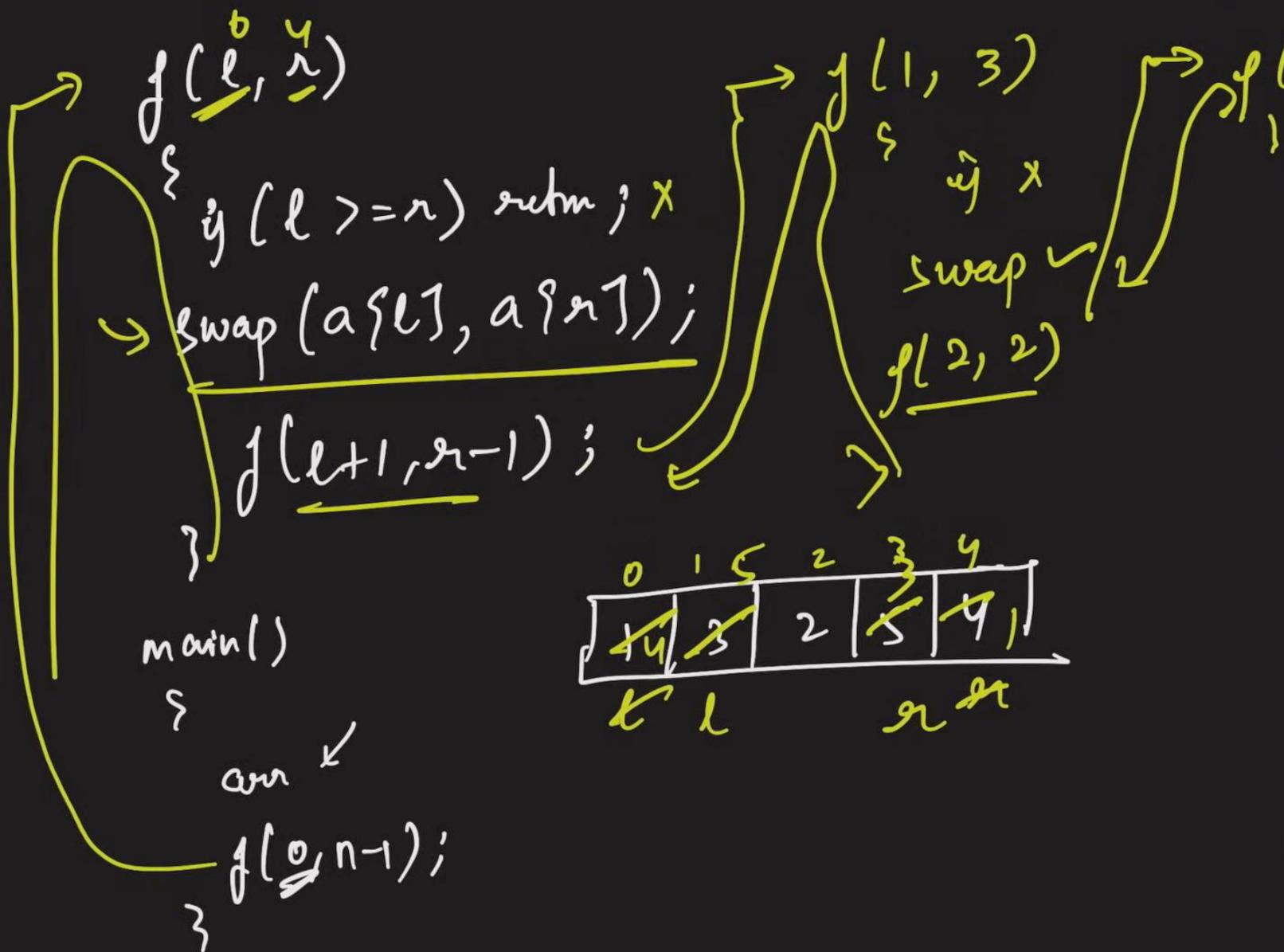


)  
 $\geq n)$  return  $; X$   
 $a[l], a[r]);$   
 $l+1, r-1);$   
 >  
 $\underline{g(l, 3)}$        $\xrightarrow{\text{if } x}$   $f(2, 2)$   
 $\underline{f(2, 2)}$        $\xrightarrow{\text{swap } \checkmark}$   $\hat{g}(2 \geq 2) \checkmark$   
 $\underline{g(l, 2)}$   
 ↘  

0	1	2	3	4
4	3	2	5	9, 1
$\leftarrow l$		$r \rightarrow$		

  
 ↙  
 $n-1);$

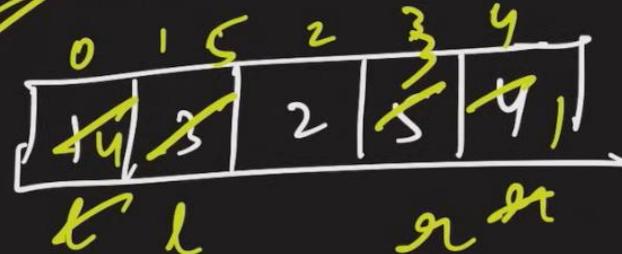


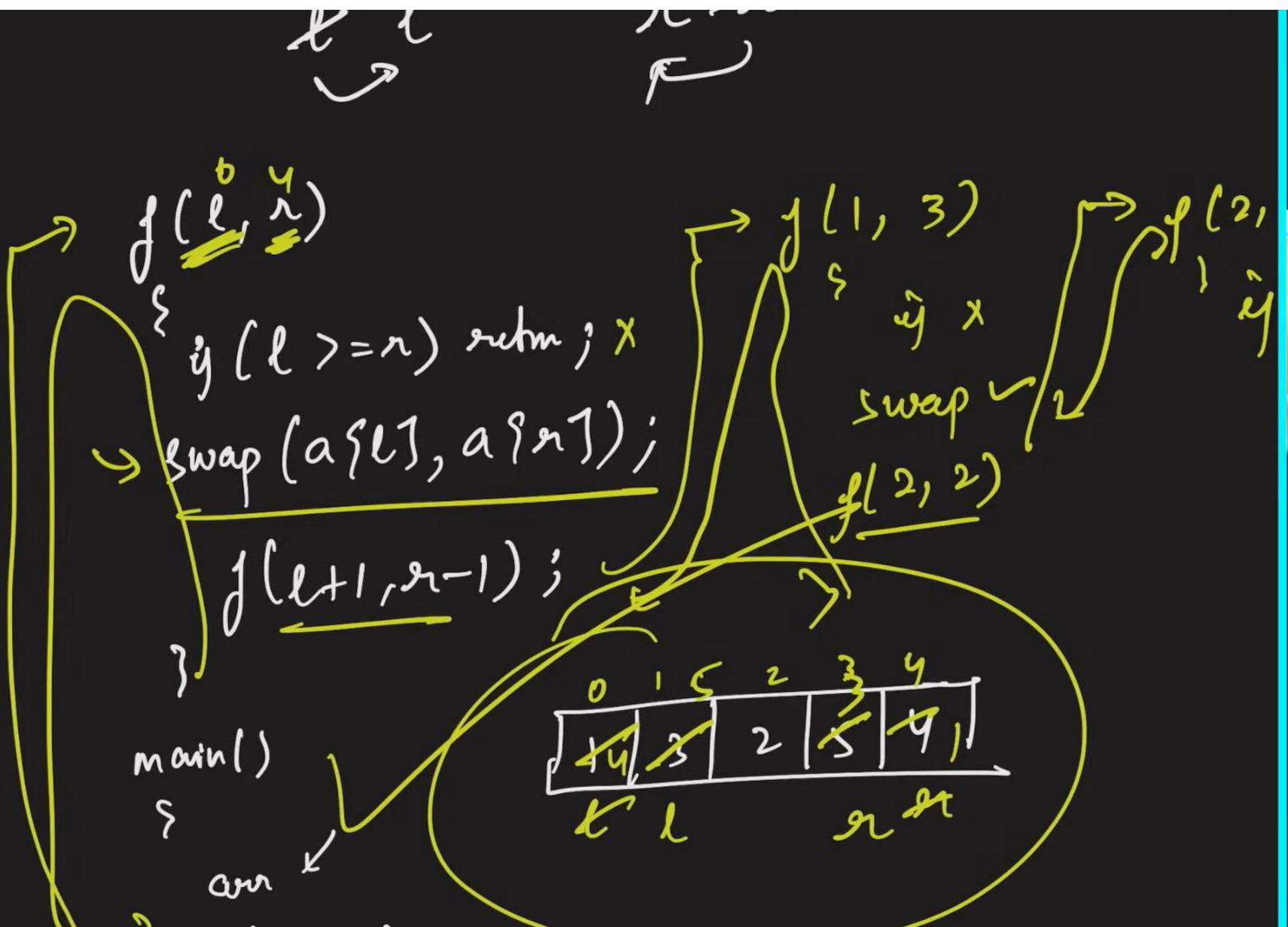


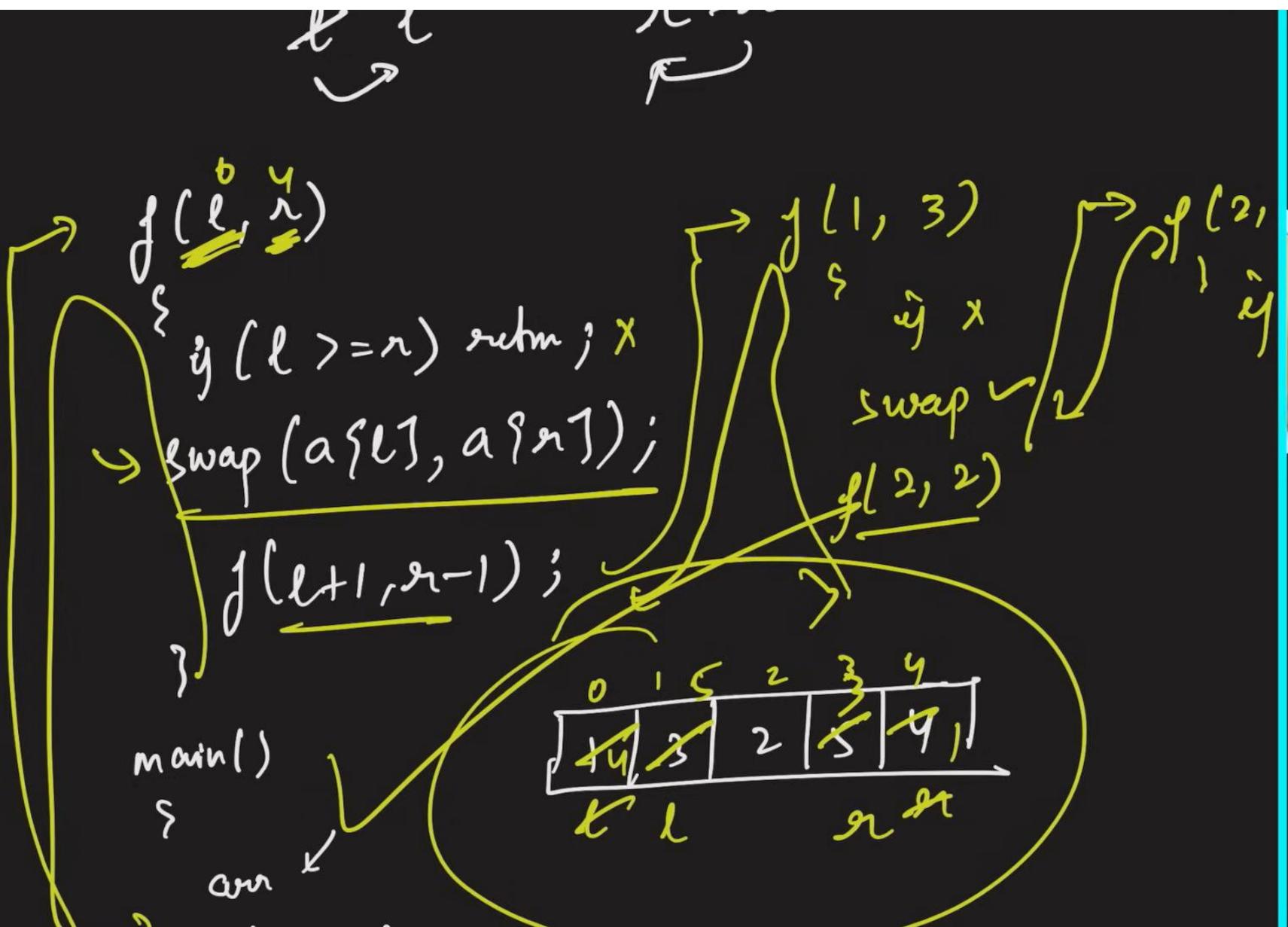
```

    } i ( l >= n ) return ;
} swap ( a [l] , a [n] );
}
j ( l + 1 , r - 1 );
    } swap v / \ f ( 2 , 2 )
    } swap v / \ f ( 2 , 2 )
main ( ) {
    } arr
    } f ( 0 , n - 1 );
}

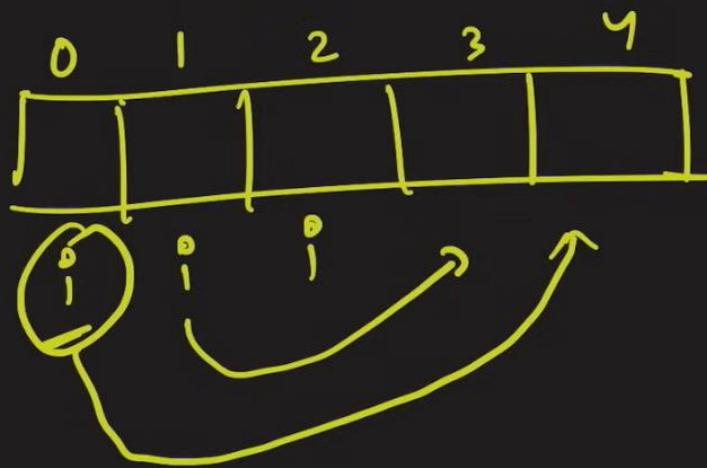
```







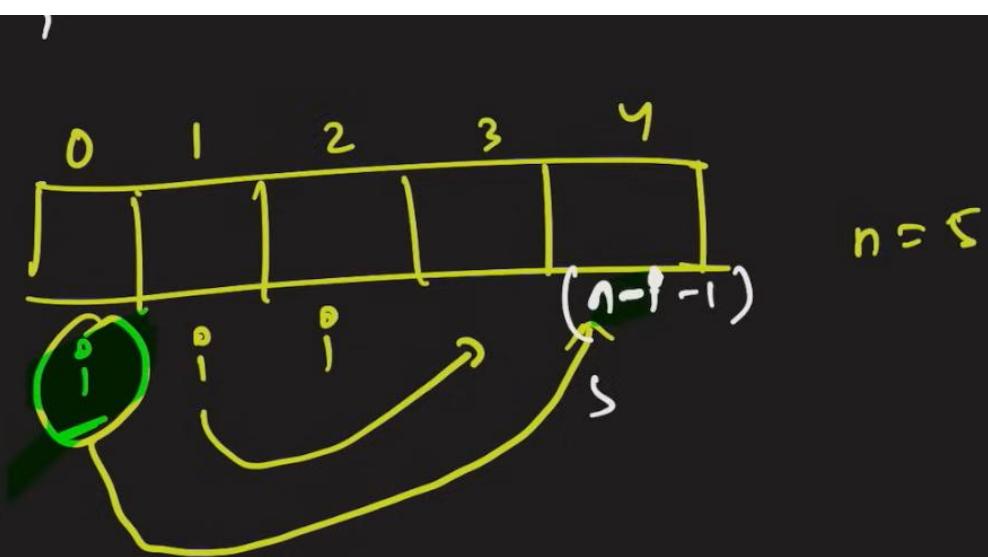
TUF



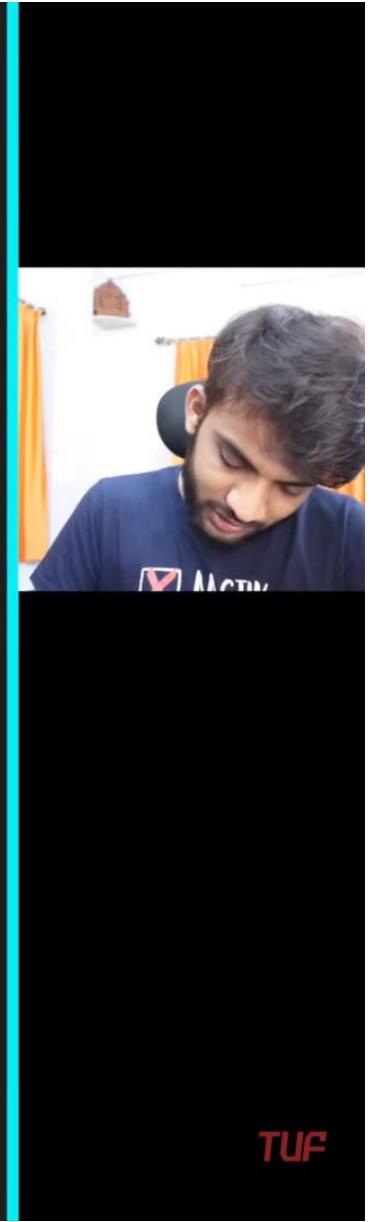
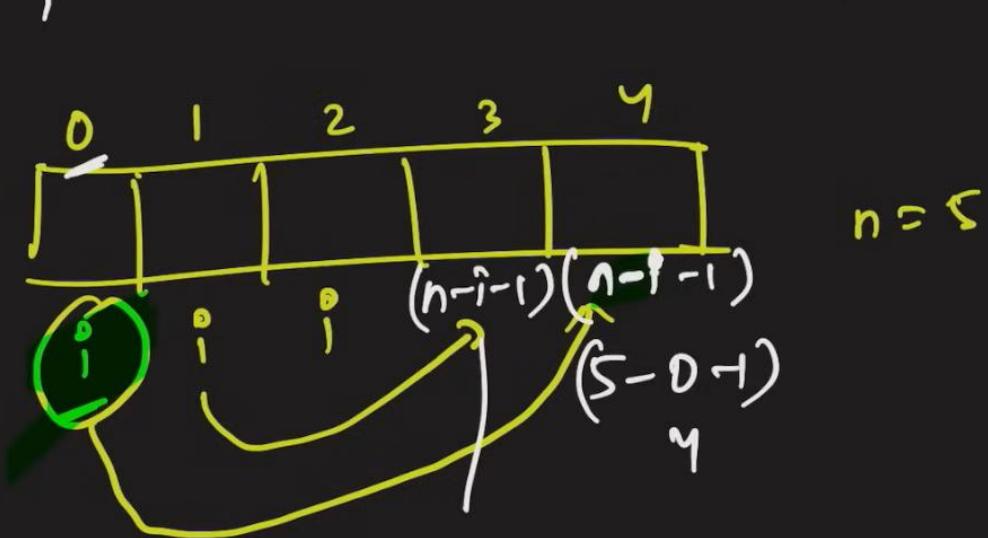
$n = 5$



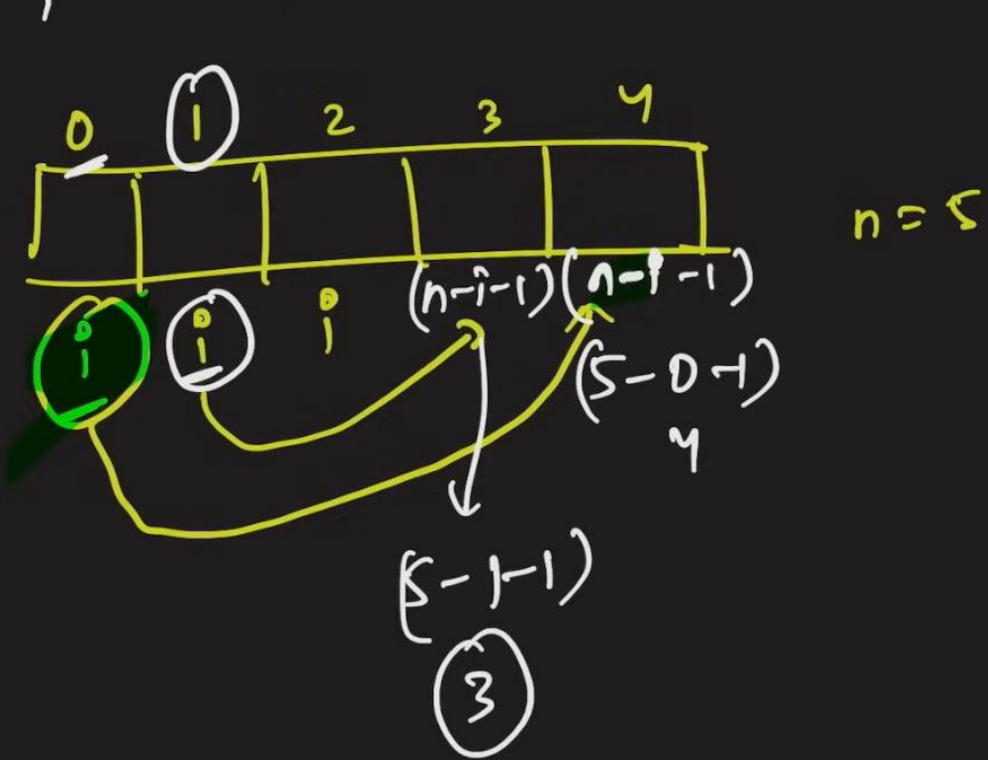
TUF



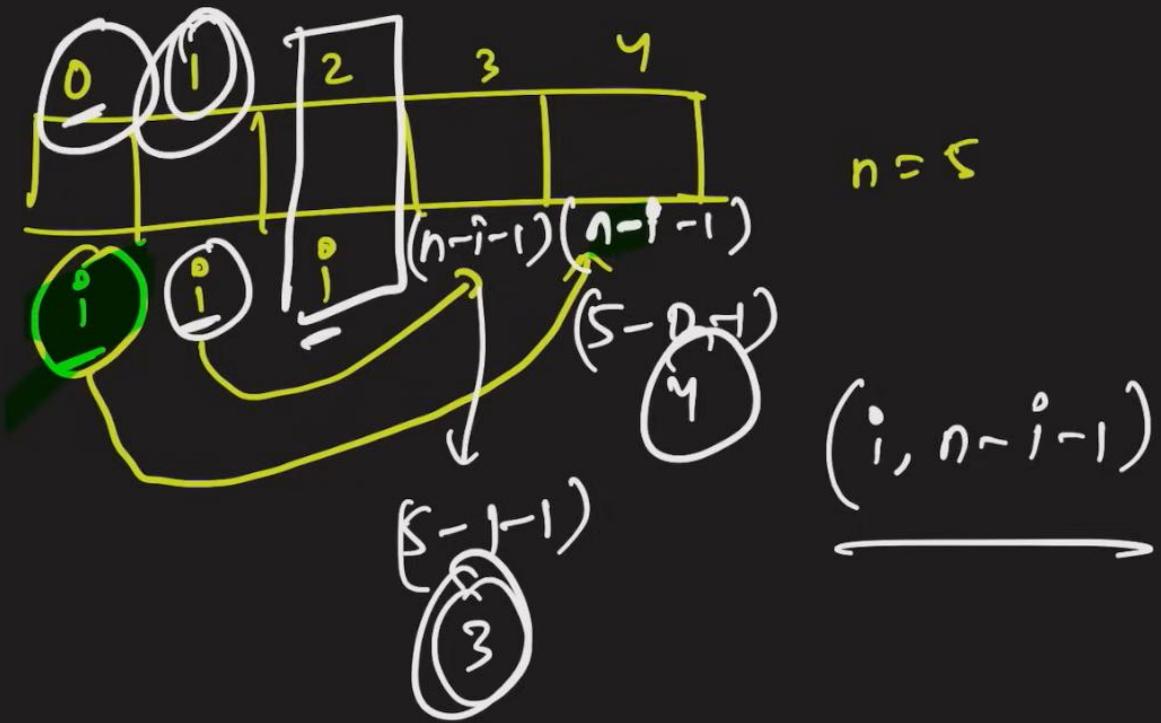
TUF

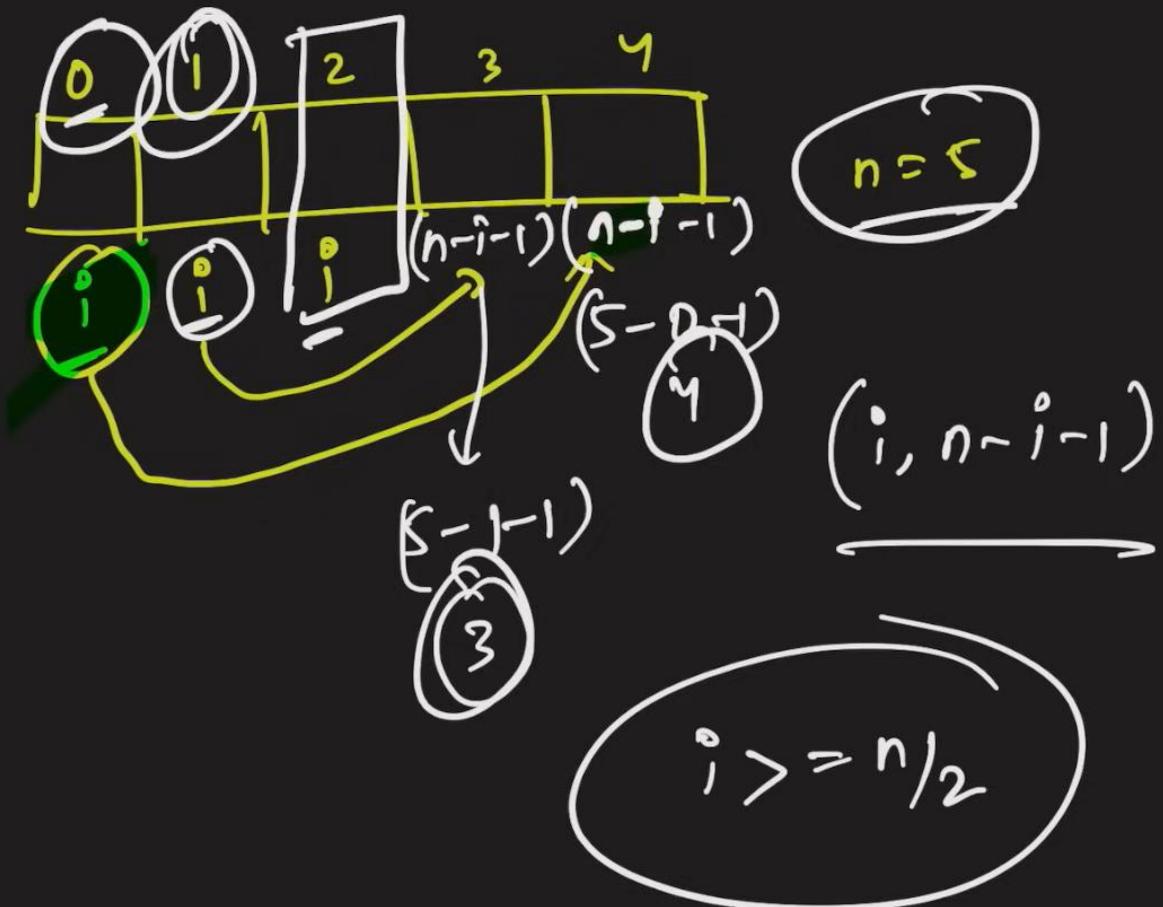


TUF



TUF





$f(i)$

{  
    if ( $i >= n/2$ )

        return ;

    swap ( $a[i], a[n-i-1]$ );



TUF

```
i <= n/2  
    {
```

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

```
    i++;  
}
```

```
m.
```



```
{  
    if (i >= n/2)  
        cout ;  
    swap(a[i], a[n-i-1]);  
    j++;  
}  
  
main()  
{  
    arr ↓  
    f(0)  
}
```



TUF

11 12

```
f(i)
{
    if (i >= n/2)
        return ;
    swap(a[i], a[n-i-1]);
    f(i+1);
}

main()
{ }
```

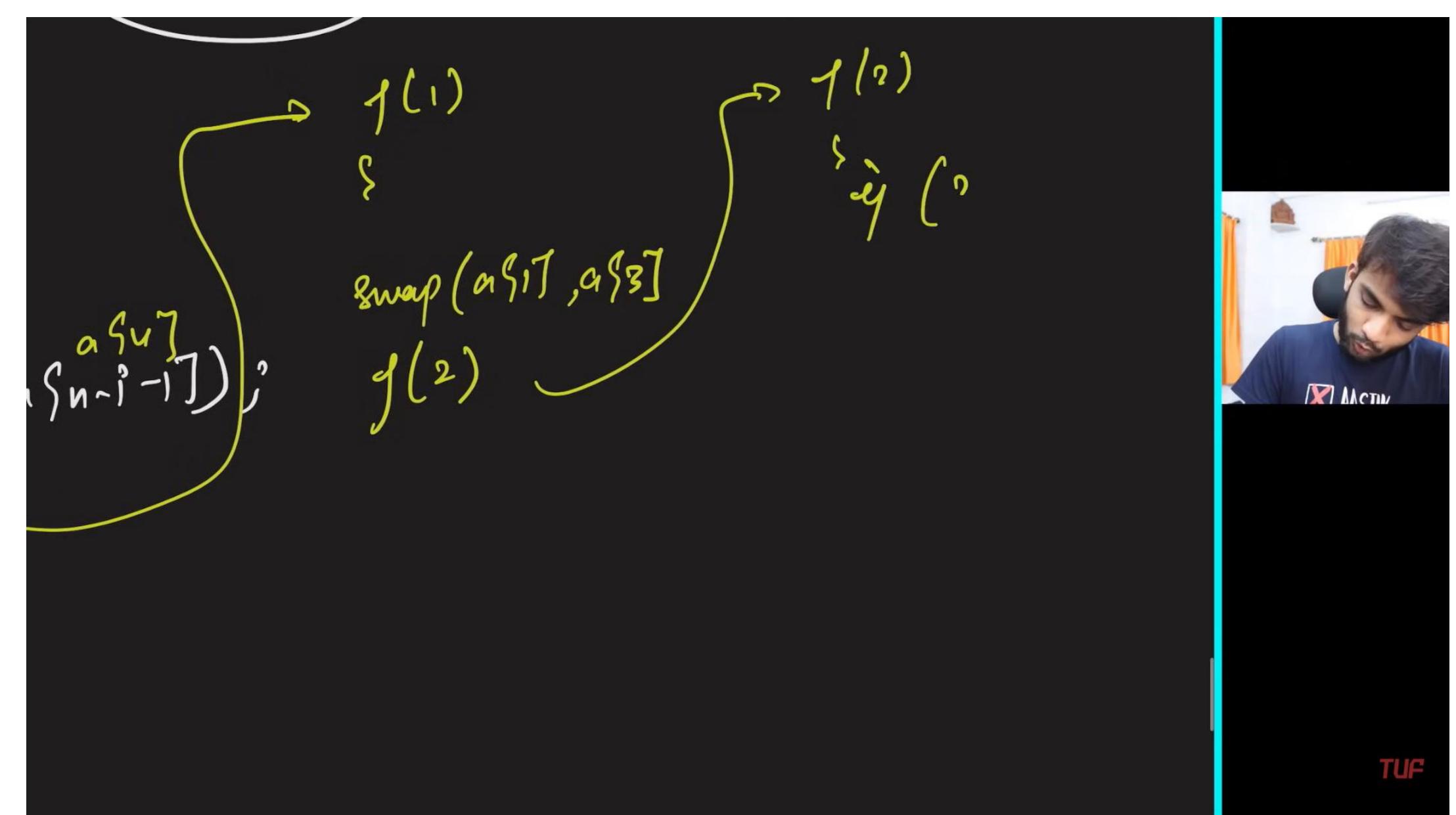


TUF

11 12

```
if (i >= n/2)
    swap(a[i], a[n-i-1]);
j(i+1);
}
main()
```





## Re 4. Problems on Functional Recursion | Strivers A2Z DSA Course



$\{ \{ \dots \{ a[5] \} \dots \} \}_{n-i-1} \} ;$

$f(1)$

$\{$

$swap(a[1], a[3])$

$f(2)$

A handwritten diagram on a blackboard. It shows a large curly brace on the left side of the screen, spanning from the bottom left towards the center. Inside this brace, there is a sequence of nested brackets and commas, representing a list or array. Above this sequence, the text "f(1)" is written in yellow. Below it, a curly brace symbol is shown. To the right of the main sequence, the text "swap(a[1], a[3])" is written in yellow. Further to the right, another curly brace symbol is shown above the text "f(2)" in yellow. Arrows point from the "f(1)" label to the first two elements of the sequence, and from the "f(2)" label to the last two elements.



TUF

```
    j(i)
    {
        if (i >= n/2)
            return ;
        swap(a[i], a[n-i-1]);
        j(i+1);
    }
main()
{
```





TUF

```
    f(i)
    {
        if (i >= n/2)
            return ;
        swap(a[i], a[n-i-1]);
        f(i+1);
    }
main()
{
    arr
```

A screenshot of a video conference interface. On the right side, a young man with dark hair and a beard is visible, wearing a dark blue t-shirt with a white graphic that includes a checkmark and the word "ACTIVE". He is holding a white stylus pen. On the left side, there is a terminal window with a dark background and light-colored text. The terminal shows code in code.cpp, which includes logic for reading from input.txt and writing to output.txt. It also shows the output of the program, which is "1 1" in the input.txt file and "1 0" in the output.txt file. At the bottom of the terminal window, the text "Finished in 0.8s]" is displayed. In the bottom right corner of the terminal window, the letters "TUF" are visible.

```
code.cpp stdc++.h
1 #include<bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     #ifndef ONLINE_JUDGE
6     freopen("input.txt", "r", stdin);
7     freopen("output.txt", "w", stdout);
8     #endif
9
10
11     return 0;
12 }
```

input.txt

1 1

output.txt

1 0

Finished in 0.8s]

TUF

A screenshot of a video conferencing interface. On the right side, a young man with dark hair and a beard, wearing a blue t-shirt, is visible from the chest up. He appears to be speaking or explaining something. To his left is a terminal window with a dark background and light-colored text. The terminal shows the following code in a file named 'code.cpp':

```
code.cpp stdc++.h
1 #include<bits/stdc++.h>
2 using namespace std;
3 void f
4 int r f
5 #for          For Loop
6 forv      Vector For Loop <stdin>;
7 fprintf      fprintf ...
8 #fopen
9 ...
10 cin >> n;
11 int arr[n];
12 for(int i = 0;i<n;i++) cin >> arr[i];
13 f(0, arr, n);
14
15 return 0;
16 }
```

The terminal has two tabs: 'input.txt' and 'output.txt'. The 'input.txt' tab shows the input '1 1'. The 'output.txt' tab shows the output '1 0'. At the bottom of the terminal window, the text 'Finished in 0.8s]' is displayed. In the bottom right corner of the terminal window, there is a small red logo with the letters 'TUF'.

A screenshot of a terminal window displaying a C++ program and its execution results. The terminal has three tabs: 'code.cpp', 'stdc++.h', and 'input.txt'. The 'code.cpp' tab shows the source code:

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 void f(int i, int arr[], int n) {
4     if(i >= n.)|
5 }
6 int main() {
7     #ifndef ONLINE_JUDGE
8     freopen("input.txt", "r", stdin);
9     freopen("output.txt", "w", stdout);
10    #endif
11    int n;
12    cin >> n;
13    int arr[n];
14    for(int i = 0;i<n;i++) cin >> arr[i];
15    f(0, arr, n);
16
17    return 0;
18 }
```

The 'input.txt' tab shows the input file content:

```
1 1
```

The 'output.txt' tab shows the output file content:

```
1 0
```

At the bottom of the terminal window, the message 'Finished in 0.8s]' is displayed. To the right of the terminal, there is a video feed of a person with dark hair and a beard, wearing a blue t-shirt. The TUF logo is visible in the bottom right corner of the video feed.

A screenshot of a video conference interface. On the left, a code editor window titled "code.cpp" shows a C++ program. The code defines a function `f` that swaps elements in an array if their index is greater than or equal to half the size. The main function reads input from "input.txt", calls `f`, and writes output to "output.txt". The developer's video feed is visible on the right, showing them wearing a blue t-shirt and headphones. The video feed has a timestamp of "1 1" at the top. The terminal window below shows the output "1 0". The bottom status bar indicates the program finished execution.

```
code.cpp
1 #include<bits/stdc++.h>
2 using namespace std;
3 void f(int i, int arr[], int n) {
4     if(i >= n/2) return;
5     swap(arr[i], arr[n-i-1]);
6 }
7
8 int main() {
9     #ifndef ONLINE_JUDGE
10    freopen("input.txt", "r", stdin);
11    freopen("output.txt", "w", stdout);
12 #endif
13    int n;
14    cin >> n;
15    int arr[n];
16    for(int i = 0;i<n;i++) cin >> arr[i];
17    f(0, arr, n);
18
19    return 0;
20 }
```

Finished in 0.8s]

1 1

1 0

TUF

```
code.cpp stdc++.h
1 #include<bits/stdc++.h>
2 using namespace std;
3 void f(int i, int arr[], int n){
4     if(i >= n/2) return;
5     swap(arr[i], arr[n-i-1]);
6     f(i+1, arr, n);
7 }
8 int main() {
9     #ifndef ONLINE_JUDGE
10    freopen("input.txt", "r", stdin);
11    freopen("output.txt", "w", stdout);
12    #endif
13    int n;
14    cin >> n;
15    int arr[n];
16    for(int i = 0;i<n;i++) cin >> arr[i];
17    f(0, arr, n);
18
19    return 0;
20 }
```

input.txt

1 1

output.txt

1 0

Finished in 0.8s]

TUF



A screenshot of a video conference interface. On the right side, a participant wearing a dark blue t-shirt with a white 'TUF' logo is visible. On the left, a code editor window titled 'code.cpp' displays C++ code. A completion dropdown menu is open over the word 'f', listing several options: 'for' (For Loop), 'forv' (Vector For Loop), 'fprintf' (fprintf ...), and 'freopen'. The code itself includes file I/O redirection and a recursive function 'f'. Below the code editor, a message 'Finished in 0.8s]' is displayed. In the top right corner of the code editor, there are two tabs: 'input.txt' and 'output.txt', each showing the values '1' and '1' respectively.

```
code.cpp
1 #include<bits/stdc++.h>
2 using namespace std;
3 void f(int i, int arr[], int n) {
4     if(i >= n/2) return;
5     swap(arr[i], arr[n-i-1]);
6     f(i+1, arr, n);
7 }
8 int main() {
9     #ifndef ONLINE_JUDGE
10    freopen("input.txt", "r", stdin);
11    freopen("output.txt", "w", stdout);
12    #endif
13    int n;
14    cin >> n;
15    int arr[n];
16    for(int i = 0;i<n;i++) cin >> arr[i];
17    f(0, arr, n);
18
19 f
20 for      For Loop
forv      Vector For Loop
fprintf      fprintf ...
freopen
```

```
code.cpp stdc++.h
1 #include<bits/stdc++.h>
2 using namespace std;
3 void f(int i, int arr[], int n) {
4     if(i >= n/2) return;
5     swap(arr[i], arr[n-i-1]);
6     f(i+1, arr, n);
7 }
8 int main() {
9     #ifndef ONLINE_JUDGE
10    freopen("input.txt", "r", stdin);
11    freopen("output.txt", "w", stdout);
12    #endif
13    int n;
14    cin >> n;
15    int arr[n];
16    for(int i = 0;i<n;i++) cin >> arr[i];
17    f(0, arr, n);
18    for(int i = 0;i<n;i++) cout << arr[i] << " ";
19    return 0;
20 }
```

Finished in 0.8s]

input.txt

1

output.txt

1 0

TUF

A screenshot of a video conference interface. On the left, a code editor shows a C++ file named 'code.cpp' with the following content:

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 void f(int i, int arr[], int n) {
4     if(i >= n/2) return;
5     swap(arr[i], arr[n-i-1]);
6     f(i+1, arr, n);
7 }
8 int main() {
9     #ifndef ONLINE_JUDGE
10    freopen("input.txt", "r", stdin);
11    freopen("output.txt", "w", stdout);
12 #endif
13    int n;
14    cin >> n;
15    int arr[n];
16    for(int i = 0;i<n;i++) cin >> arr[i];
17    f(0, arr, n);
18    for(int i = 0;i<n;i++) cout << arr[i] << " ";
19    return 0;
20 }
```

On the right, there is a terminal window titled 'input.txt' containing the following input data:

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

Below it is another terminal window titled 'output.txt' containing the following output data:

```
1 0
```

In the bottom right corner of the video frame, there is a small image of a young man with a beard, wearing a dark t-shirt, looking towards the camera.

TUF

A screenshot of a video call interface. On the left, a code editor window shows a C++ file named 'code.cpp'. The code contains a recursive function 'f' that swaps elements in an array if the index 'i' is greater than or equal to n/2. It also includes code to handle input from 'input.txt' and output to 'output.txt'. The main function reads an integer 'n' from standard input, initializes an array 'arr' of size 'n', and prints the array elements to standard output. The code editor has tabs for 'code.cpp' and 'stdc++.h'. Below the code editor is a terminal window showing the output: 'Finished in 1.5s]'. To the right of the code editor is another terminal window titled 'input.txt' containing the input array [1, 5, 1, 2, 3, 4, 5]. Below it is another terminal window titled 'output.txt' containing the reversed array [1, 5, 4, 3, 2, 1]. On the far right, a video feed of a young man with a beard, wearing a dark t-shirt with a logo, is speaking and gesturing with his hands. The background shows orange curtains.

```
code.cpp
1 #include<bits/stdc++.h>
2 using namespace std;
3 void f(int i, int arr[], int n) {
4     if(i >= n/2) return;
5     swap(arr[i], arr[n-i-1]);
6     f(i+1, arr, n);
7 }
8 int main() {
9     #ifndef ONLINE_JUDGE
10    freopen("input.txt", "r", stdin);
11    freopen("output.txt", "w", stdout);
12    #endif
13    int n;
14    cin >> n;
15    int arr[n];
16    for(int i = 0;i<n;i++) cin >> arr[i];
17    f(0, arr, n);
18    for(int i = 0;i<n;i++) cout << arr[i] << " ";
19    return 0;
20 }
```

Finished in 1.5s]

input.txt

output.txt

TUF

Check if a string is palindrome.



TUF

Q Check if a string is palindrome.  
→ a string on reversal

area



TUF

≡ Re 4. Problems on Functional Recursion | Strivers A2Z DSA Course

~~↳ Checks if a String is palindrome~~  
→ a string on reversal  
reads the same



TUF

◀ ▶ ▷ 🔍 11:28 / 19:47

▼

CC HD

Q Check if a string is palindrome.  
→ a string on reversal  
reads the same

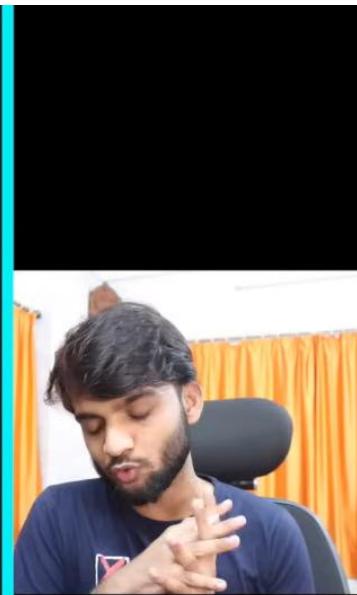
"MADAM"  $\xrightarrow{\text{reverse}}$



TUF

Q Check if a string is palindrome.  
→ a string on reversal  
reads the same

"MADAM" → reverse "M A D A M"  
"11211" → "11211"



M A D A M



TUF

✓  
M A D A M  
— — — — —



TUF

M A D A M

— — — — —



TUF

M A D A M



TUF

## ≡ Re 4. Problems on Functional Recursion | Strivers A2Z DSA Course



M A D A M



TUF



13:19 / 19:47



M A D A M

```
f(i)
if (i >= n/2) return true;
if ([i:i] == [n-i-1:n-i])
    return f(i+1);
}
return f(i+1);
```



M A D A M

$f(i)$   
if ( $i >= n/2$ ) return true;  
if ( $\lfloor s_i \rfloor = \lfloor s_{n-i-1} \rfloor$ )  
return  $f$  else;  
return  $f(i+1);$

}



$f(i)$

if ( $i \geq n/2$ ) return true;

if ( $\lfloor \log i \rfloor = \lfloor \log_{n-i-1} \rfloor$ )  
return  $f$  else;

return  $f(i+1);$

}

main()

{

print( $f(0)$ )



TUF

M A D A M

$f(i)$

$i \geq n/2$  return true;

$i < n/2$   $f(n-i) = f(n-i-1)$   
return false;

return  ~~$f(i+1)$~~

}

main()

{



TUF

## Re 4. Problems on Functional Recursion | Strivers A2Z DSA Course



0

$f(i)$

$y()$  x

$e(s[i]) = s[i]$

exit

$i >= n/2$  return;

$i < s[i] \neq s[n-i-1]$  return  $f$  else;

return  $f(i+1)$ ;

}

main()



15:54 / 19:47

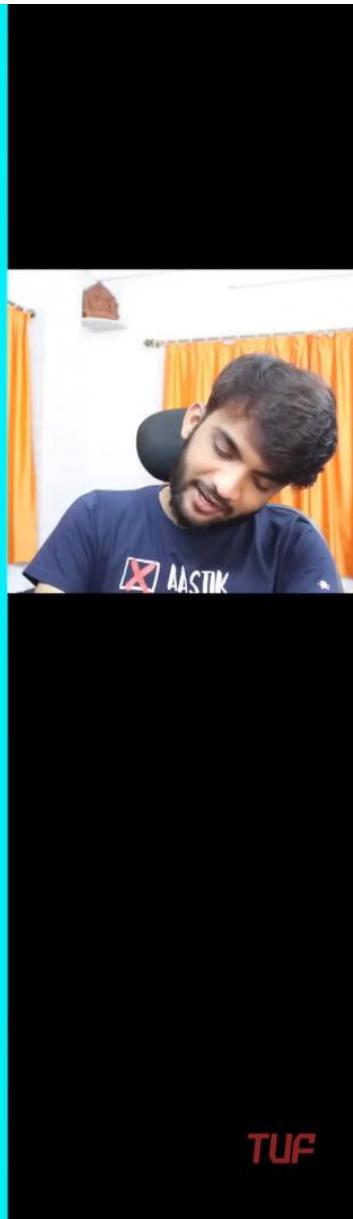


TUF



frame;  
     $\hat{y}^{(1)}$   
     $\hat{y}^{(2)} = 1$

```
    if (c[i] != c[3])  
        f(2)
```



frame;  
    ~~$\vec{y}_1$~~   
     $\vec{y}_1$  )  
e;

$\vec{y}_1$  )  
     $\vec{y}_1$  x  
 $\vec{y}_1$  ( $\vec{y}_1$  ! =  $\vec{y}_2$ )  
    ~~ent~~ ~~ent~~

$\vec{y}_2$  )  
     $\vec{y}_2$  )  
     $\vec{y}_2$  =  $\vec{y}_1$  ✓  
ekhn Tm

~~M A D C M~~

$j(i)$

$i >= n/2$  return true;

$i < i = \lfloor n - i \rfloor$

return false;

return

$H[i]$

}

main()

$\zeta$

$f(1)$

$y()$

$e(j) = \lfloor s \rfloor$

return

$I$

TUF



~~M A D C M~~

$f(i)$

$\hat{y}(i \geq n_1)$  sub true;

$\hat{y}(\lfloor s_i \rfloor) = \lfloor s_{n-i+1} \rfloor$

sub if false;

sub

$i \in I$

main

$f(1)$

$\hat{y}(1) = x$

$\hat{y}(\lfloor s_1 \rfloor) = \lfloor s_3 \rfloor$

sub

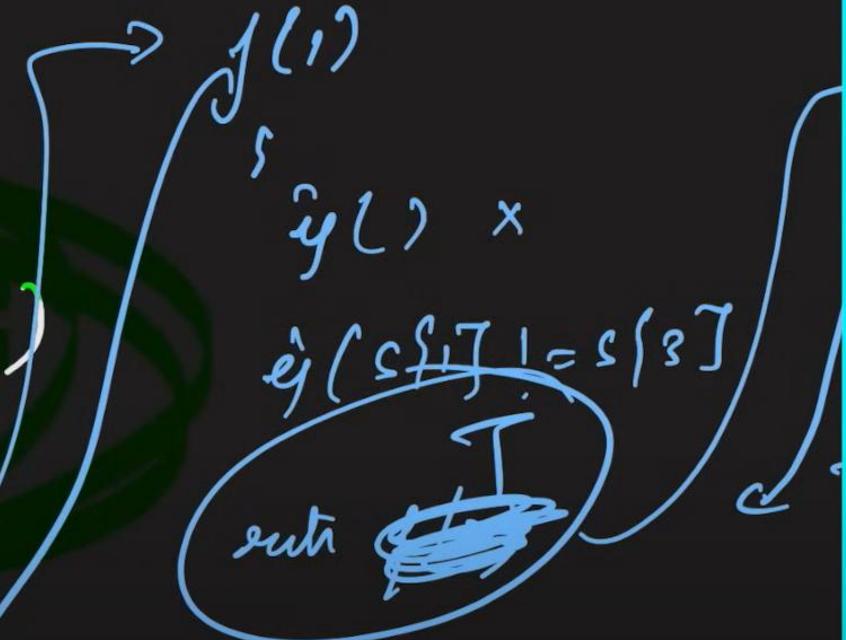
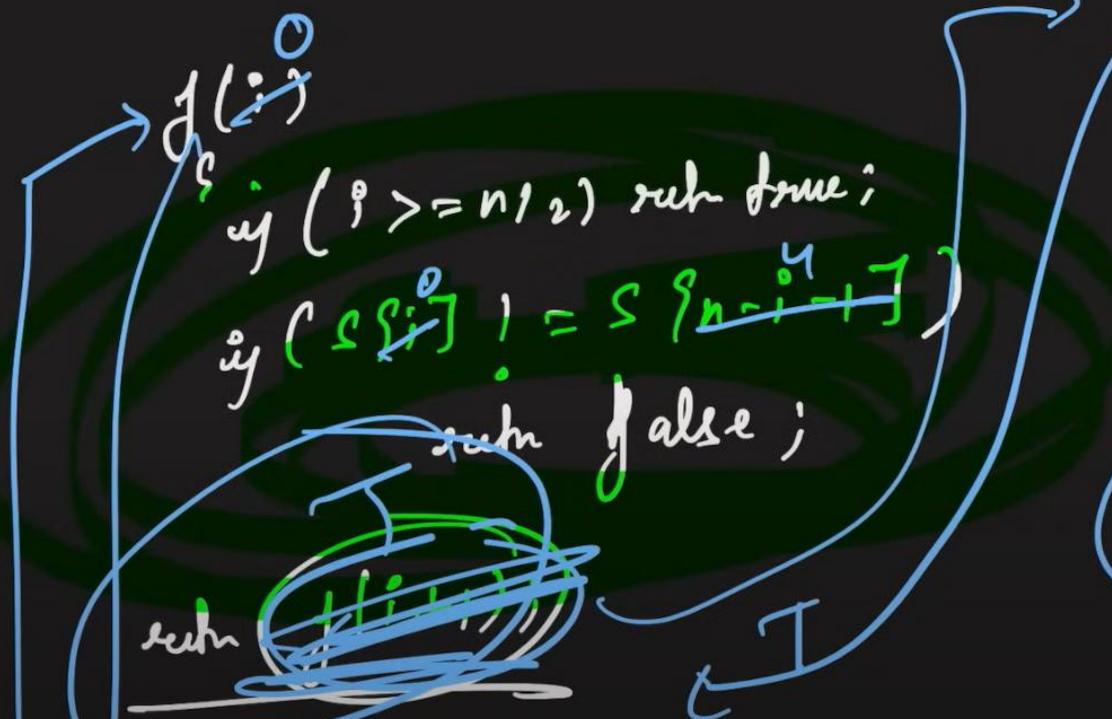
TUF

## Re 4. Problems on Functional Recursion | Strivers A2Z DSA Course



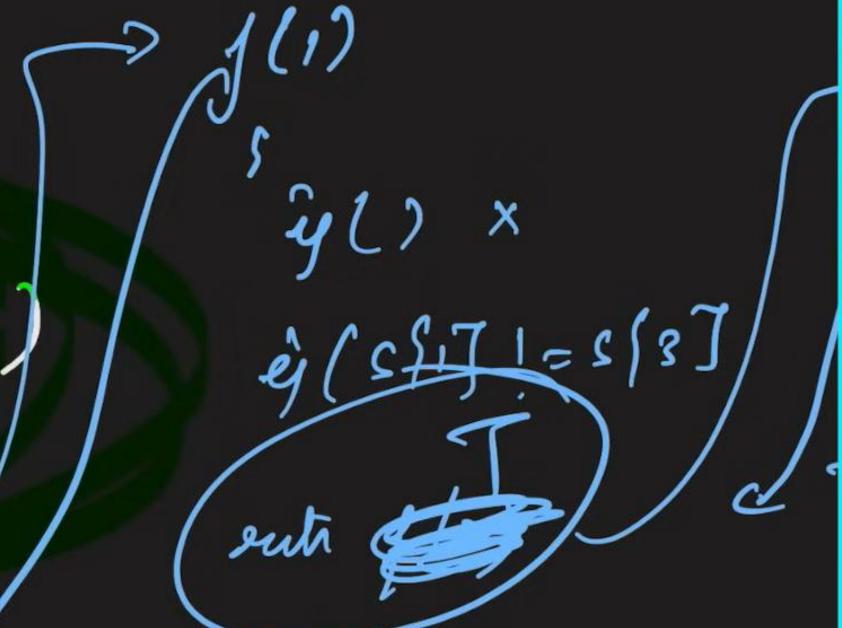
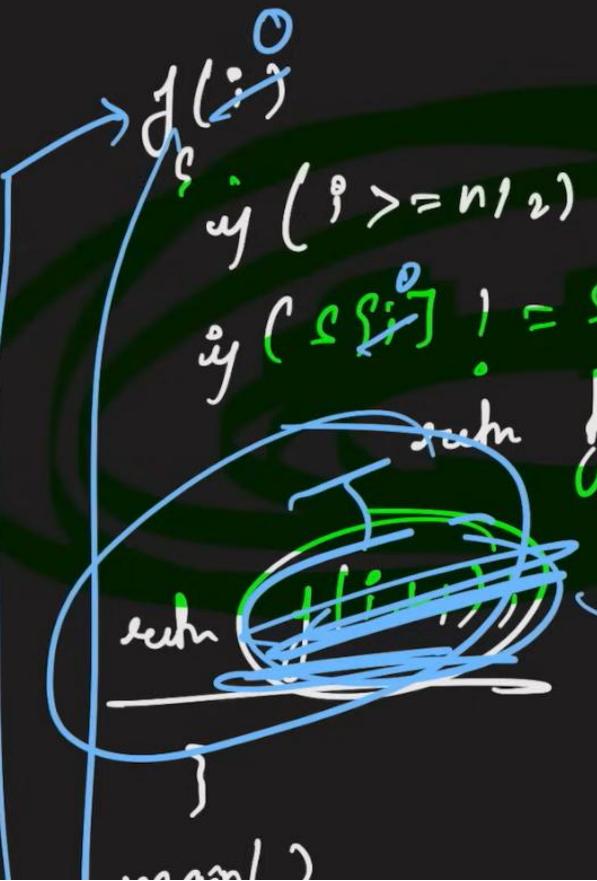
~~M A D ACM~~

'MADSM'



~~M A D C M~~

"MADSM"



~~MAP DOCUMENT~~

0

$f(x)$

$y \geq n_1$  sub frame;

$y \leq n_1 \quad i = \lfloor n - 1 \rfloor$

subn } else;

subn

$H_{i+1}$

main()

"MAP DOCUMENT"

$f(i)$

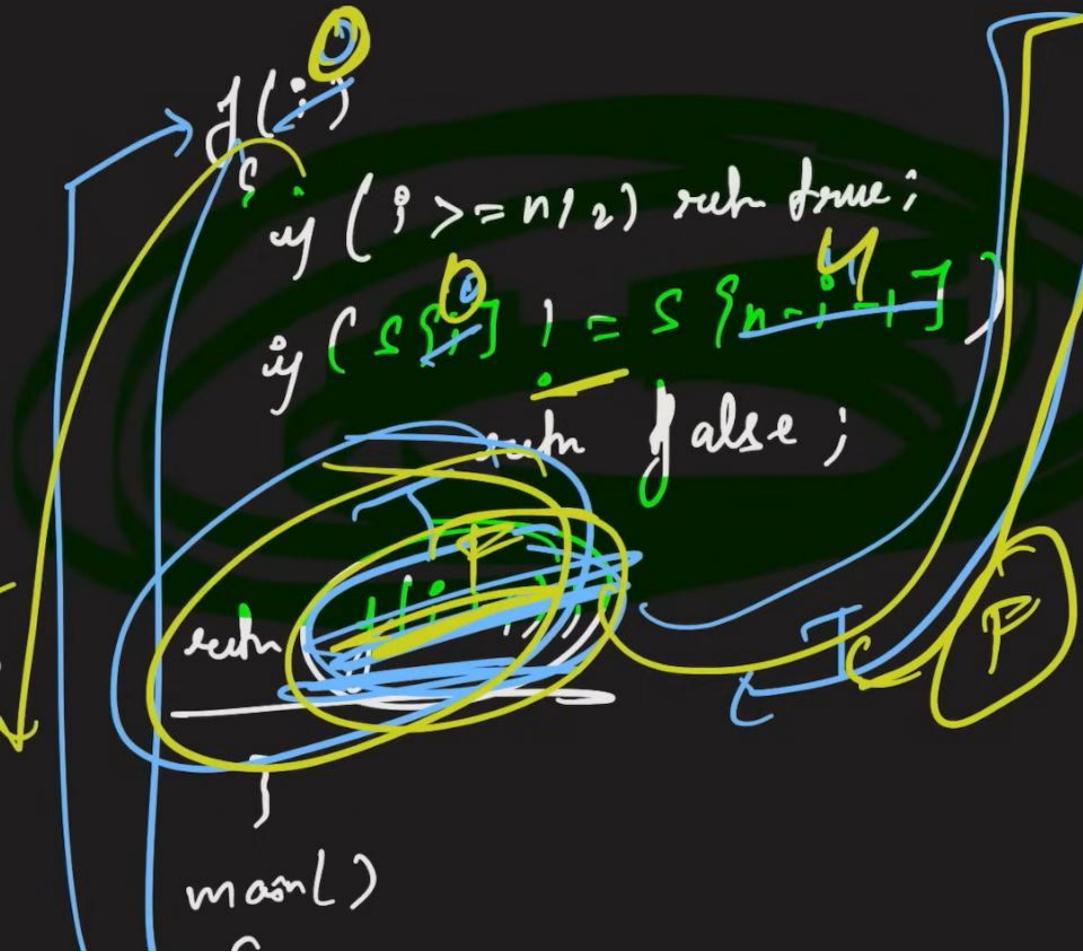
$y(x)$

$e \leq 1 \quad i = 3$

subn

TUF

~~MAP DATUM~~



"MAP DATUM"

$f(1)$

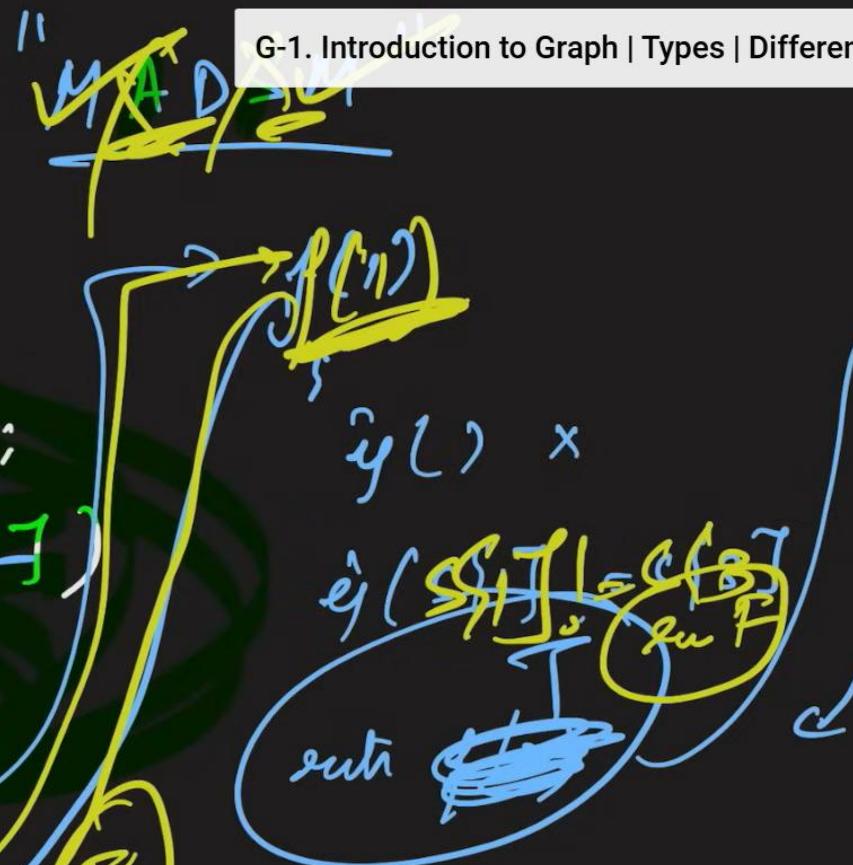
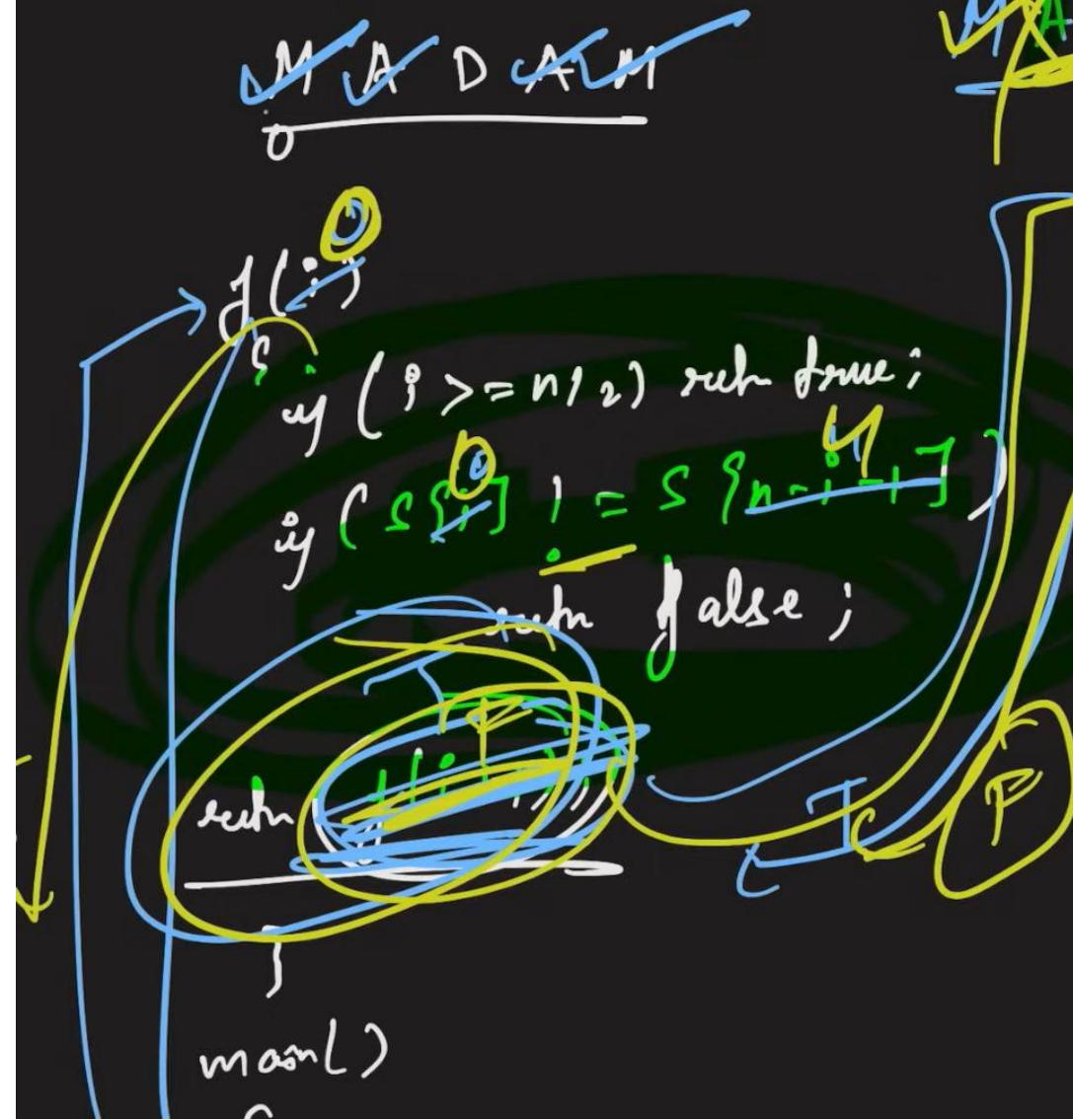
$y(1) x$

$e(1)$   $\lceil \frac{n-i-1}{2} \rceil = \lceil \frac{3}{2} \rceil$

subn

TUF

## G-1. Introduction to Graph | Types | Different Conventions Used



TUF

A screenshot of a video conference interface. On the left, a code editor window shows a C++ file named 'code.cpp'. The code implements a recursive function 'f' to reverse an array segment, with a main function reading from 'input.txt' and writing to 'output.txt'. The code editor has tabs for 'code.cpp' and 'stdc++.h'. To the right of the code editor are two terminal windows. The top terminal window is titled 'input.txt' and contains the input array: 1 5 2 3 4 5. The bottom terminal window is titled 'output.txt' and shows the reversed output array: 1 5 4 3 2 1. A vertical cyan line separates the terminal windows from a video feed on the right, which shows a young man with a beard. At the bottom left, the text 'Finished in 1.5s]' is displayed. In the bottom right corner, there is a red watermark or logo that says 'TUF'.

```
code.cpp
1 #include<bits/stdc++.h>
2 using namespace std;
3 void f(int i, int arr[], int n) {
4     if(i >= n/2) return;
5     swap(arr[i], arr[n-i-1]);
6     f(i+1, arr, n);
7 }
8 int main() {
9     #ifndef ONLINE_JUDGE
10    freopen("input.txt", "r", stdin);
11    freopen("output.txt", "w", stdout);
12 #endif
13    int n;
14    cin >> n;
15    int arr[n];
16    for(int i = 0;i<n;i++) cin >> arr[i];
17    f(0, arr, n);
18    for(int i = 0;i<n;i++) cout << arr[i] << " ";
19    return 0;
20 }
```

Finished in 1.5s]

TUF

## Re 4. Problems on Functional Recursion | Strivers A2Z DSA Course

```
1 #include <iostream>
2 using namespace std;
3 bool f(int i, string &s) {
4     if()
5 }
6 int main() {
7     #ifndef ONLINE_JUDGE
8     freopen("input.txt", "r", stdin);
9     freopen("output.txt", "w", stdout);
10    #endif
11    string s = "madam";
12    cout << f(0, s);
13    return 0;
14 }
```

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



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

Finished in 1.5s]

TUF

◀ ▶ ▷ 🔍 18:01 / 19:47



CC HD

## Re 4. Problems on Functional Recursion | Strivers A2Z DSA Course

```
1 //include <iostream>
2 using namespace std;
3 bool f(int i, string &s) {
4     if()
5 }
6 int main() {
7     #ifndef ONLINE_JUDGE
8     freopen("input.txt", "r", stdin);
9     freopen("output.txt", "w", stdout);
10    #endif
11    string s = "madam";
12    cout << f(0, s);
13    return 0;
14 }
```

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



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

Finished in 1.5s]

TUF

◀ ▶ ▷ 🔍 18:01 / 19:47



CC HD

## Re 4. Problems on Functional Recursion | Strivers A2Z DSA Course

```
1 #include <iostream>
2 using namespace std;
3 bool f(int i, string &s) {
4     if(i >= s.size())
5 }
6 int main() {
7     #ifndef ONLINE_JUDGE
8     freopen("input.txt", "r", stdin);
9     freopen("output.txt", "w", stdout);
10    #endif
11    string s = "madam";
12    cout << f(0, s);
13 }
14 }
```

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



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

Finished in 1.5s]

TUF



18:04 / 19:47



A screenshot of a video conference interface. On the left, a code editor window shows a C++ file named 'code.cpp'. The code defines a function 'f' that checks if a string is a palindrome. It includes logic to skip punctuation and spaces. The main function reads from 'input.txt' and writes to 'output.txt'. A terminal window below the code editor shows the command 'g++ code.cpp -o code'. On the right, a video feed of a young man with a beard, wearing a dark t-shirt, is visible. He appears to be explaining something. Below the video feed is a terminal window showing the output of the program, which is '1 5 4 3 2 1'. At the bottom of the screen, a status bar displays the message 'Finished in 1.5s].

```
code.cpp
1 #include<bits/stdc++.h>
2 using namespace std;
3 bool f(int i, string &s) {
4     if(i >= s.size() / 2) return true;
5     if(s[i] != s[i])
6 }
7 int main() {
8     #ifndef ONLINE_JUDGE
9         freopen("input.txt", "r", stdin);
10        freopen("output.txt", "w", stdout);
11    #endif
12    string s = "madam";
13    cout << f(0, s);
14 }
15 }
```

```
input.txt
1 5
2 1 2 3 4 5
```

```
output.txt
1 5 4 3 2 1
```

Finished in 1.5s]

```
code.cpp stdc++.h
1 #include<bits/stdc++.h>
2 using namespace std;
3 bool f(int i, string &s) {
4     if(i >= s.size() / 2) return true;
5     if(s[i] != s[s.size() - i - 1]) return false;
6 }
7
8 int main() {
9     #ifndef ONLINE_JUDGE
10    freopen("input.txt", "r", stdin);
11    freopen("output.txt", "w", stdout);
12    #endif
13    string s = "madam";
14    cout << f(0, s);
15    return 0;
16 }
```

input.txt

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

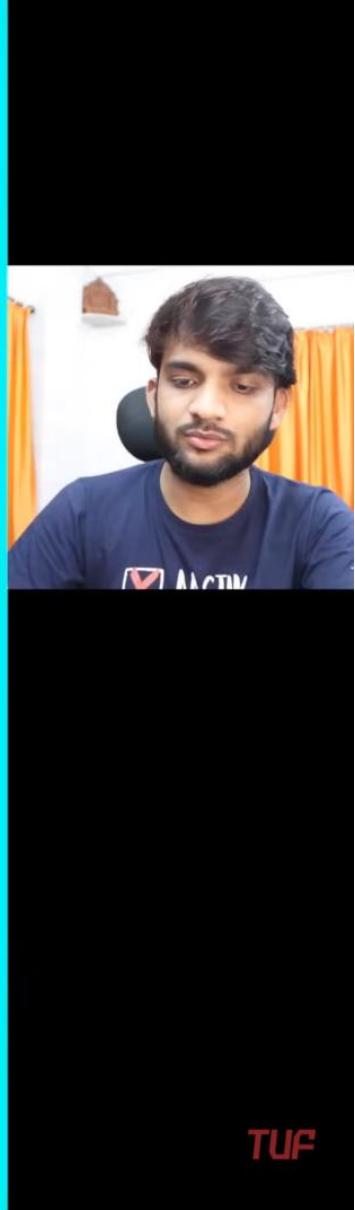
output.txt

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



Finished in 1.5s]

TUF



A screenshot of a video conference interface. On the left, a code editor window shows a C++ file named "code.cpp". The code defines a function `f` that checks if a string is a palindrome. It includes logic to skip the first half of the string and compare characters from the center outwards. The main function reads from "input.txt" and writes to "output.txt", testing the string "madam". On the right, a video feed of a man with dark hair and a beard, wearing a blue t-shirt, is visible. He is looking at the screen. The video feed has a watermark in the bottom right corner that says "TUF".

```
code.cpp
1 #include<bits/stdc++.h>
2 using namespace std;
3 bool f(int i, string &s) {
4     if(i >= s.size() / 2) return true;
5     if(s[i] != s[s.size() - i - 1]) return false;
6     return f(i+1, s);
7 }
8 int main() {
9     #ifndef ONLINE_JUDGE
10    freopen("input.txt", "r", stdin);
11    freopen("output.txt", "w", stdout);
12    #endif
13    string s = "madam";
14    cout << f(0, s);
15    return 0;
16 }
```

input.txt

1 5  
2 1 2 3 4 5

output.txt

1 5 4 3 2 1

The image shows a developer's workspace with a code editor and a video feed.

**Code Editor:**

- File:** code.cpp
- Content:**

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 bool f(int i, string &s) {
4     if(i >= s.size() / 2) return true;
5     if(s[i] != s[s.size() - i - 1]) return false;
6     return f(i+1, s);
7 }
8 int main() {
9     #ifndef ONLINE_JUDGE
10    freopen("input.txt", "r", stdin);
11    freopen("output.txt", "w", stdout);
12 #endif
13    string s = "madam";
14    cout << f(0, s);
15    return 0;
16 }
```

**Input File:** input.txt

1	5
2	1 2 3 4 5

**Output File:** output.txt

1	1
---	---

**Video Feed:** A young man with dark hair and a beard, wearing a blue t-shirt, is visible on the right side of the screen. He appears to be speaking or explaining something.

**Text at Bottom:** Finished in 1.2s]

**Logo:** TUF

```
code.cpp      stdc++.h      input.txt      output.txt
1 #include<bits/stdc++.h>
2 using namespace std;
3 bool f(int i, string &s) {
4     if(i >= s.size() / 2) return true;
5     if(s[i] != s[s.size() - i - 1]) return false;
6     return f(i+1, s);
7 }
8 int main() {
9     #ifndef ONLINE_JUDGE
10    freopen("input.txt", "r", stdin);
11    freopen("output.txt", "w", stdout);
12    #endif
13    string s = "madsm";
14    cout << f(0, s);
15    return 0;
16 }
```

Finished in 1.0s]

1 5  
2 1 2 3 4 5

1 0

TUF



A screenshot of a terminal window showing a C++ program being run. The terminal has three tabs: 'code.cpp', 'stdc++.h', and 'input.txt'. The code.cpp tab contains the following code:

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 bool f(int i, string &s) {
4     if(i >= s.size() / 2) return true;
5     if(s[i] != s[s.size() - i - 1]) return false;
6     return f(i+1, s);
7 }
8 int main() {
9     #ifndef ONLINE_JUDGE
10    freopen("input.txt", "r", stdin);
11    freopen("output.txt", "w", stdout);
12    #endif
13    string s = "madsm";
14    cout << f(0, s);
15    return 0;
16 }
```

The input.txt tab shows the input: 1 5  
2 1 2 3 4 5. The output.txt tab shows the output: 1 0.

At the bottom left, it says 'Finished in 1.0s]' and at the bottom right, there is a 'TUF' watermark.

code.cpp

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 bool f(int i, string &s) {
4     if(i >= s.size() / 2) return true;
5     if(s[i] != s[s.size() - i - 1]) return false;
6     return f(i+1, s);
7 }
8 int main() {
9     #ifndef ONLINE_JUDGE
10    freopen("input.txt", "r", stdin);
11    freopen("output.txt", "w", stdout);
12    #endif
13    string s = "madsm";
14    cout << f(0, s);
15    return 0;
16 }
```

input.txt

1	5
2	1 2 3 4 5

output.txt

1	0
---	---

Finished in 1.0s]

TUF

A screenshot of a video conference interface. On the right side, there is a video feed of a man with dark hair and a beard, wearing a dark blue t-shirt with a white logo that includes a checkmark and the word "AASTIK". He is looking towards the camera. To his left is a code editor window with the following code:

```
code.cpp      stdc++.h
1 #include<bits/stdc++.h>
2 using namespace std;
3 bool f(int i, string &s) {
4     if(i >= s.size() / 2) return true;
5     if(s[i] != s[s.size() - i - 1]) return false;
6     return f(i+1, s);
7 }
8 int main() {
9     #ifndef ONLINE_JUDGE
10    freopen("input.txt", "r", stdin);
11    freopen("output.txt", "w", stdout);
12    #endif
13    string s = "madsm";
14    cout << f(0, s);
15    return 0;
16 }
```

The code editor has tabs for "code.cpp" and "stdc++.h". Below the code editor are two terminal windows. The top terminal window is titled "input.txt" and contains the following text:

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

The bottom terminal window is titled "output.txt" and contains the following text:

```
1 0
```

At the bottom left of the screen, there is a message: "Finished in 1.0s]".

```
code.cpp      stdc++.h      input.txt      output.txt
1 #include<bits/stdc++.h>
2 using namespace std;
3 bool f(int i, string &s) {
4     if(i >= s.size() / 2) return true;
5     if(s[i] != s[s.size() - i - 1]) return false;
6     return f(i+1, s);
7 }
8 int main() {
9     #ifndef ONLINE_JUDGE
10    freopen("input.txt", "r", stdin);
11    freopen("output.txt", "w", stdout);
12    #endif
13    string s = "madsm";
14    cout << f(0, s);
15    return 0;
16 }
```

Finished in 1.0s]

1 5  
2 1 2 3 4 5

1 0

TUF

A screenshot of a video player interface. On the left, there's a code editor window titled "code.cpp" showing C++ code for a binary search algorithm. A tooltip from a video thumbnail is overlaid on the editor, displaying the text "Binary Trees | Binary Search Trees | C++ | Java | Data Structures and Algorithm..." and "54 videos". On the right, there's a terminal window titled "input.txt" containing the input "1 5\n2 1 2 3 4 5" and another window titled "output.txt" showing the output "1 0". In the bottom right corner of the video player, there's a circular watermark with the letters "TUF".

Striver's Graph Series | Playlist for people who have limited time TO  
**GRAPH SERIES**

30 videos

UNAPOLOGETICALLY ME

Finished in 1:05