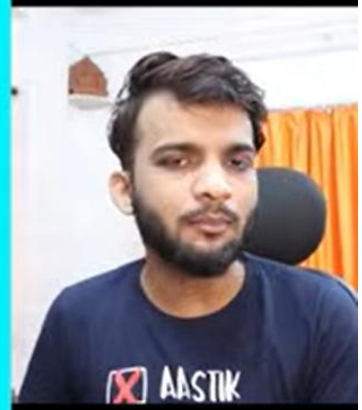
Recursion → When a function calls itself until a specified condition is met.

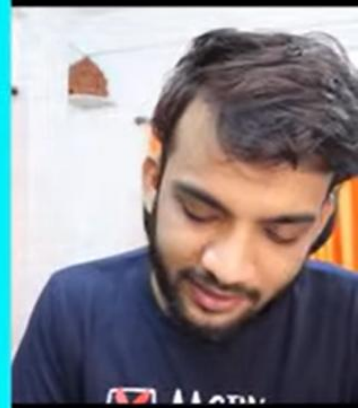


TUF

Recursion ?? → (When a function calls itself until a specified condition is met.

f()
{

f()
{



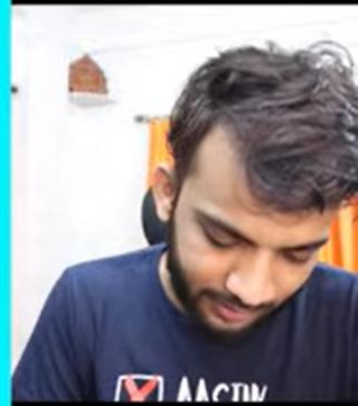
TUF

Recursion:

→ (When a function calls itself)

→ until a specified condition is met.

```
main()
{
  f();
}
```



Recursion :

→ (When a function calls itself)

→ until a specified condition is met.

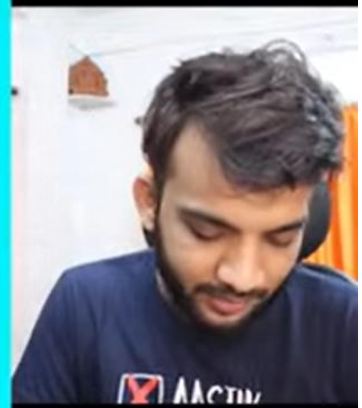
```
void f()
{
  print(1)
  f();
}

main()
{

  f();
}
```

TUF

Re 1. Introduction to Recursion | Recursion Tree | Stack Space | Strivers A2Z DSA Cou...

→ until a specified condition is met.

```
void f()
{
  print(1)
  f();
}

main()
{
  f();
}
```



3:28 / 22:45

```
void f()
{
print(1)
f();
}

main()
{
f();
}
```

Output

1

TUF

→ until a specified condition is met.

```
void f()
{
    print(1)
    f();
}

void f()
{
    print(1)
    f()
}

main()
{
    f();
}
```

Output

TUF

→ until a specified condition is met.

```
void f()
{
    print(1)
    f();
}

void f()
{
    print(1)
    f()
}

void f()
{
}

main()
{
    f();
}
```

Output

TUF

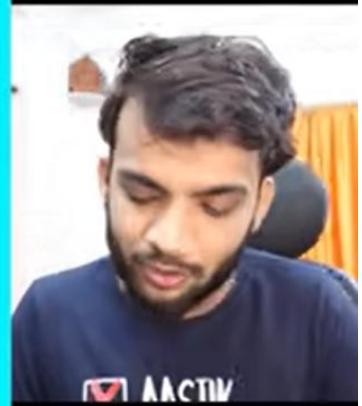
```
void f()
{
  print(1)
  f();
}

void f()
{
  print(1)
  f()
}

void f()
{
  print(1)
  f()
}

main()
{
  f();
}
```

Output

1

1

1



TUF

→ until a specified condition is met.

```
void f()
{
  print(1)
  f();
}

void f()
{
  print(1)
  f()
}

void f()
{
  print(1)
  f()
}

void f()
{
  print(1)
  f();
}

main()
{
  f();
}
```

Output

TUF

Recursion !! → (When a function calls itself)

→ until a specified condition is met.

```
void f()
{
  print(1)
  f();
}

void f()
{
  print(1)
  f()
}

void f()
{
  print(1)
  f()
}

void f()
{
  print(1)
  f();
}

main()
{
  f();
}
```

TUF

condition is met.

```
void f()
{
  print(1)
  f()
}
```

```
void f()
{
  print(1)
  f();
}
```

Output

TUF

code.cpp

```cpp
#include<bits/stdc++.h>
using namespace std;

void print() {
    cout << 1 << endl;
    print();
}

int main() {
    #ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
    #endif

    print();

    return 0;
}
```

input.txt

```
1
```

output.txt

```
299  1
300  1
301  1
302  1
303  1
304  1
305  1
306  1
307  1
308  1
309  1
310  1
311  1
312  1
313  1
```

```
bash: line 1: 32281 Segmentation fault: 11  '/Users/striver/Desktop/SublimeFolder/code'
[Finished in 2.5s with exit code 139]
[cmd: ['bash', '-c', "g++ -std=c++14 '/Users/striver/Desktop/SublimeFolder/code.cpp' -o '/Users/striver/Desktop/SublimeFolde
code' && '/Users/striver/Desktop/SublimeFolder/code'"]]
[dir: /Users/striver/Desktop/SublimeFolder]
[path: /usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin]
```

TUF

code.cpp

```cpp
#include<bits/stdc++.h>
using namespace std;

void print() {
    cout << 1 << endl;
    print();
}

int main() {
    #ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
    #endif

    print();

    return 0;
}
```

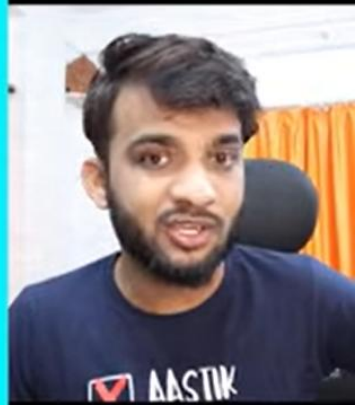
input.txt

```
1
```

output.txt

```
119 1
120 1
121 1
122 1
123 1
124 1
125 1
126 1
127 1
128 1
129 1
130 1
131 1
132 1
```

```
bash: line 1: 32711 Segmentation fault: 11  '/Users/striver/Desktop/SublimeFolder/code'
[Finished in 2.6s with exit code 139]
[cmd: ['bash', '-c', "g++ -std=c++14 '/Users/striver/Desktop/SublimeFolder/code.cpp' -o '/Users/striver/Desktop/SublimeFolde
code' && '/Users/striver/Desktop/SublimeFolder/code'"]]
[dir: /Users/striver/Desktop/SublimeFolder]
[path: /usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin]
```



TUF

code.cpp

```cpp
#include<bits/stdc++.h>
using namespace std;

void print() {
    cout << 1 << endl;
    print();
}

int main() {
    #ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
    #endif

    print();

    return 0;
}
```

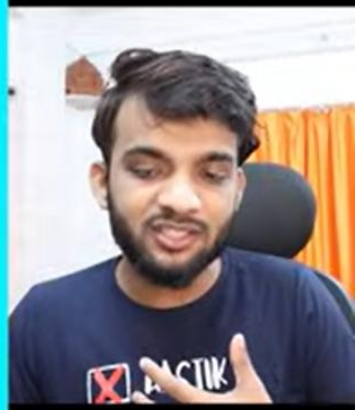
input.txt

```
1
```

output.txt

```
523024  1
523025  1
523026  1
523027  1
523028  1
523029  1
523030  1
523031  1
523032  1
523033  1
523034  1
523035  1
523036  1
523037  1
523038
```

```
bash: line 1: 32711 Segmentation fault: 11  '/Users/striver/Desktop/SublimeFolder/code'
[Finished in 2.6s with exit code 139]
[cmd: ['bash', '-c', "g++ -std=c++14 '/Users/striver/Desktop/SublimeFolder/code.cpp' -o '/Users/striver/Desktop/SublimeFolde
code' && '/Users/striver/Desktop/SublimeFolder/code'"]]
[dir: /Users/striver/Desktop/SublimeFolder]
[path: /usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin]
```



code.cpp

```cpp
#include<bits/stdc++.h>
using namespace std;

void print() {
    cout << 1 << endl;
    print();
}

int main() {
    #ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
    #endif

    print();

    return 0;
}
```

input.txt

```
1
```

output.txt

```
523024  1
523025  1
523026  1
523027  1
523028  1
523029  1
523030  1
523031  1
523032  1
523033  1
523034  1
523035  1
523036  1
523037  1
523038
```

```
bash: line 1: 32711 Segmentation fault: 11  '/Users/striver/Desktop/SublimeFolder/code'
[Finished in 2.6s with exit code 139]
[cmd: ['bash', '-c', "g++ -std=c++14 '/Users/striver/Desktop/SublimeFolder/code.cpp' -o '/Users/striver/Desktop/SublimeFolde
code' && '/Users/striver/Desktop/SublimeFolder/code'"]]
[dir: /Users/striver/Desktop/SublimeFolder]
[path: /usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin]
```

TUF

```
void f()
{
    print(1)
    f();
}

void f()
{
    print(1)
    f()
}

void f()
{
    print(1)
    f()
}

void f()
{
    print(
    f()
}

main()
{
    f();
}
```

Output

1

→ Until a specified [illegible]

```
void f()
{
    print(1)
    f();
}
```

```
void f()
{
    print(1)
    f()
}
```

```
void f()
{
    print(1)
    f()
}
```

```
void f(
{
    print(
    f()
}
```

```
main()
{
    f();
}
```

Output

1

TUF

→ Init a specific area

```
void f()
{
  print(1)
  f();
}

void f()
{
  print(1)
  f()
}

void f()
{
  print(1)
  f()
}

void f(
{
  print(
  f()
}

main()
{
  f();
}
```

Stack

Output

1

TUF

void f()
{
print(1)
f();
}

void f()
{
print(1)
f()
}

void f()
{
print(1)
f()
}

void f(
{
print(
f()
}

main()
{
f();
}

Stack

f(), L2
f(), L2
f(), L2

Output

1

TUF

void f()
{
print(1)
f();
}

void f()
{
print(1)
f()
}

void f()
{
print(1)
f()
}

void f(
{
print(
f()
}

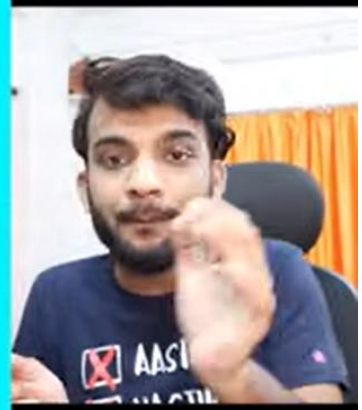
main()
{
f();
}

Stack

f(), L2
f(), L2
f(), L2

Output

1



print (1)
f();

print (1)
f()
}

print
f()
}

f();
}

main ()
{
f();
}

Stack

f(), L2
f(), L2
f(), L2

Output

1
1
1

TUF

print (1)

f();

main ()

{

f();

}

Stack

print (1)

f();

}

f()

f()

f()

f(), L2

f(), L2

f(), L2

print (1)

f()

}

f();

}

Segmentation

Stack overflow

Output

1

1

1

TUF

```
print(1)
f();
```

```
print(1)
f()
}
```

```
print(1)
f()
}
```

```
f();
}
```

```
main()
{
f();
}
```

Stack

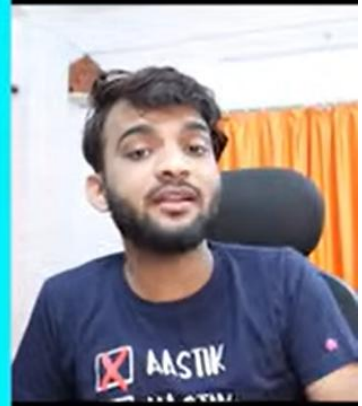
f()
f()
f()
f(), L2
f(), L2
f(), L2

Segmentation

Stack overflow

Output

1
1
1



TUF

```
void f()
{
print(1)
f();
}
```

```
void f()
{
print(1)
f()
}
```

```
{
print(1)
f()
}
```

```
{
print(1
f();
}
```

```
main()
{
f();
}
```

Stack

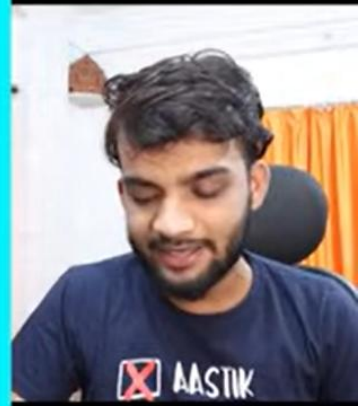
f()
f()
f()
f(), 22
f(), 22
f(), 12

Segmentation

Stack overflow

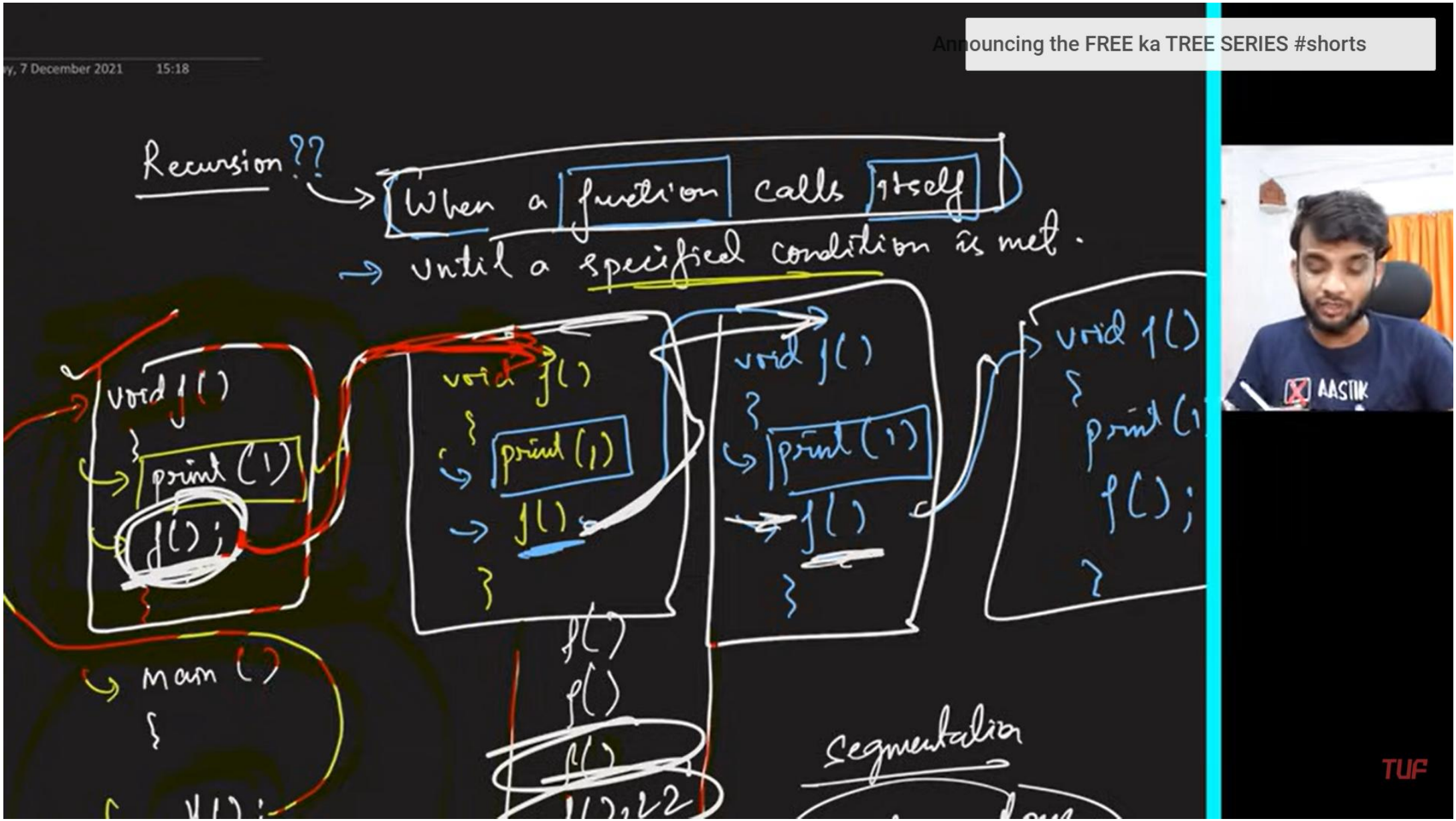
Output

1
1
1

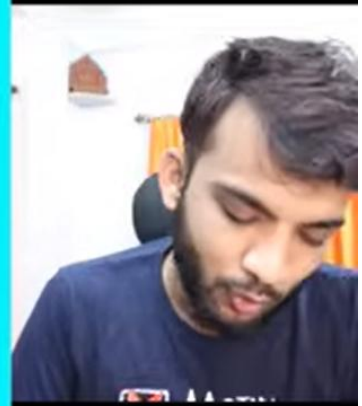


TUF

Announcing the FREE ka TREE SERIES #shorts



```
cnt = 0
f()
{
  print(cnt)
  cnt++;
  f();
}
```



TUF

```
cnt = 0

f()
{
  print(cnt)
  cnt++;

  f();
}

main()
{

  f();
}
```

TUF

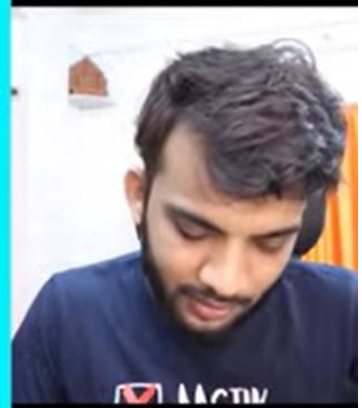
```
cnt = 0 + 2

f()
{
    print(cnt)
    cnt++;
    f();
}

main()
{
    f();
}
```

```
f()
{
    print(cnt)
    cnt++;
    f();
}
```

Output

0

1



```
cnt = 0 + 2

f()
{
  print(cnt)
  cnt++;
  f();
}

main()
{
  f();
}
```

```
f()
{
  print(cnt)
  cnt++;
  f();
}
```

```
f()
{
  print(cnt)
  cnt++
  f()
}
```

Output

0

1

2

TUF

cnt = ~~0 + 2~~ 3

```
f()
{
  print(cnt)
  cnt++;
  f();
}
```

```
f()
{
  print(cnt)
  cnt++;
  f();
}
```

```
f()
{
  print(cnt)
  cnt++
  f()
}
```

Base Condition

```
main()
{
  f();
}
```

Output

0
1
2

cnt = ~~0 + 2~~ 3

```
f()
{
  print(cnt)
  cnt++;
  f();
}
```

```
f()
{
  print(cnt)
  cnt++;
  f();
}
```

```
f()
{
  print(cnt)
  cnt++
  f()
}
```

(Base Condition)

```
main()
{
  f();
}
```

Output

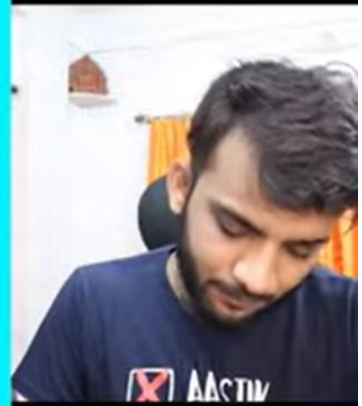
0

1

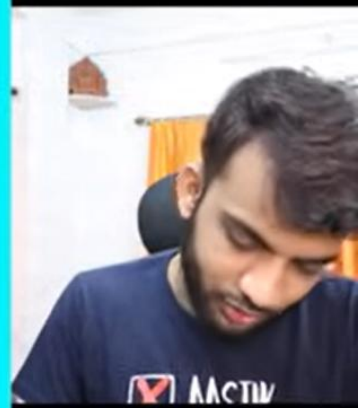
2

TUF

```
cnt=0
f()
{
  if(cnt==4)
      return;
  print(cnt)
  cnt++
  f()
  }
n
```



```
cnt=0
f()
{
  if(cnt==4)
      return;
  print(cnt)
  cnt++
  f()
  }
main()
 {
  f();
 }
```



TUF

```
cnt=0
f()
{
  if(cnt==4)
    return;
  print(cnt)
  cnt++
  f()
}
main()
{
  f();
}
```

Output

TUF

```
cnt = 0 1
f()
{
  if (cnt == 4)  X
    return;
  print(cnt)
  cnt++
  f()
}
main()
{
  f();
}
```

Output

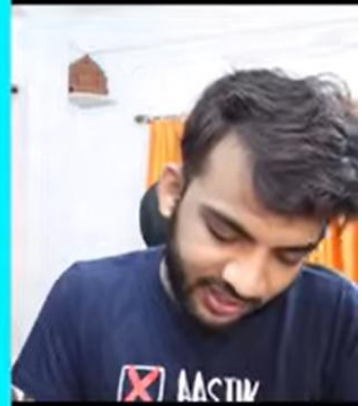
0

TUF

```
}

main()
{
    f();
}
```

Output

0



TUF

3

cnt = ~~0~~ 1

```
f()
{
  if(cnt == 4)
    return;
  print(cnt)
  cnt++
  f()
}

main()
{
  f();
```

```
f()
{
  if(cnt == 4)
    return;
  print(cnt)
  cnt++;
  f();
}
```

Output

0

TUF

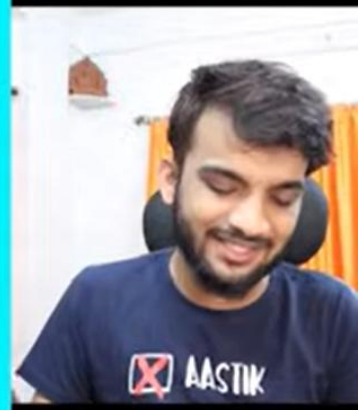
```
cnt = 0
f()
{
  if(cnt == 4)
    return;
  print(cnt)
  cnt++
  f()
}
main()
{
  f();
}
```

```
f()
{
  if(cnt == 4)
    return;
  print(cnt)
  cnt++;
  f();
}
```

Output

0

1



TUF

cnt = ~~0~~ 2

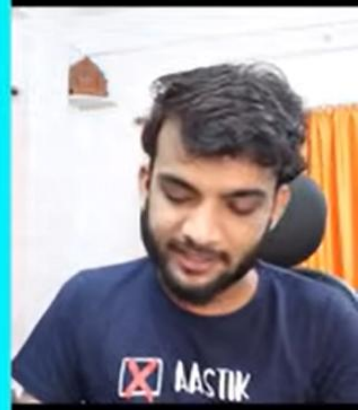
```
f()
{
  if(cnt == 2)
    return;
  print(cnt)
  cnt++
  f()
}

main()
{
  f();
```

```
f()
{
  if(cnt == 2)
    return;
  print(cnt)
  cnt++;
  f();
}
```

```
f()
{
  if(cnt == 3)
    return
  print(cnt)
  cnt++
```

Output

0



TUF

```
f()
}
```

```
f();
}
```

```
f();
}
```

```
main()
{
f();
}
```

Output

0

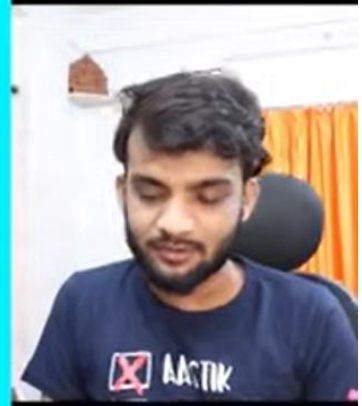
1

2

f(), L5

f(), L5

f(), L5



TUF

cnt = ~~2~~ 3

```
f()
{
  if(cnt == 3) ×
    return;
  print(cnt)
  cnt++
  f()
}
```

```
f()
{
  if(cnt == 2) ×
    return;
  print(cnt) ✓
  cnt++; ✓
  f();
}
```

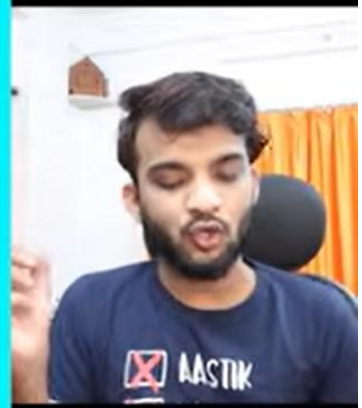
```
f()
{
  if(cnt == 3) ×
    return
  print(cnt)
  cnt++
  f();
}
```

```
main()
{
  f();
```

Output

0

1



TUF

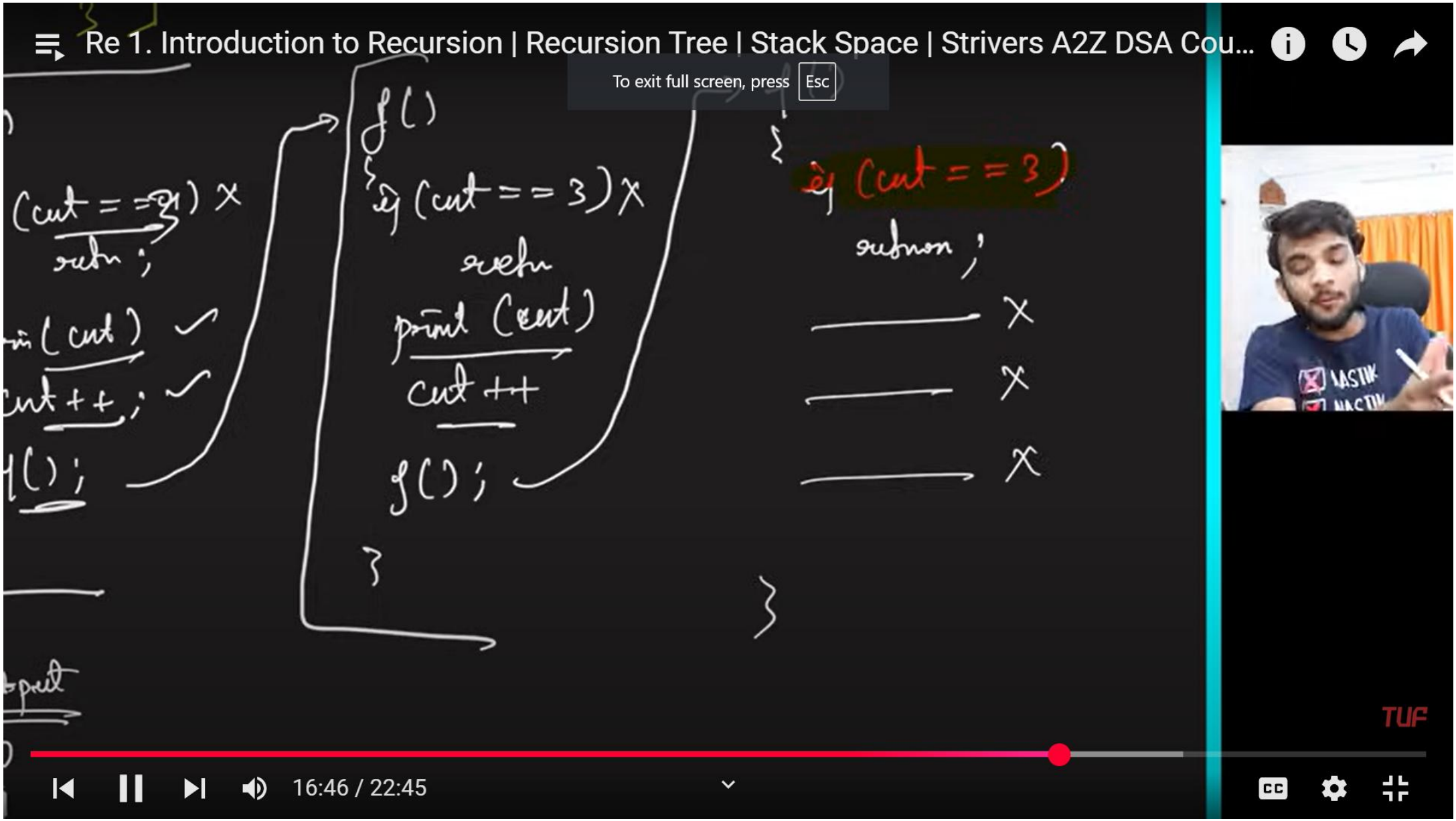
Let's give back to the community :)

```
(cnt == 3) ✗
return;
print(cnt) ✓
cnt++; ✓
f();
```

```
f()
{
if (cnt == 3) ✗
return
print(cnt)
cnt++
f();
}
```

```
f()
{
if (cnt == 3)
return;
```

TUF


Re 1. Introduction to Recursion | Recursion Tree | Stack Space | Strivers A2Z DSA Cou...
To exit full screen, press Esc
f()
if (cnt == 3)
return
print (cnt)
cnt++
f();
if (cnt == 3)
return;
TUF
16:46 / 22:45

```
(cnt == 3) ✗
return;
print(cnt) ✓
cnt++; ✓
f();
```

```
f()
{
if (cnt == 3) ✗
return
print(cnt)
cnt++
f();
}
```

```
f()
{
if (cnt == 3)
return;
}
```

✗

✗

✗

cnt = ~~0~~ ~~1~~ 3

```
f()
{
  if(cnt == ~~1~~ 3)
    return;
  print(cnt)
  cnt++
  f()
}
```

```
f()
{
  if(cnt == 1)
    return;
  print(cnt)
  cnt++;
  f();
}
```

```
f()
{
  if(cnt == 3)
    return
  print(cnt)
  cnt++
  f();
}
```

```
main()
{
  f();
}
```

Output

0

1

TUF

```
f()
{
  if(cnt == 3)
    return;
  print(cnt)
  cnt++
  f()
}
main()
{
  f();
}
```

```
f()
{
  if(cnt == 1)
    return;
  print(cnt)
  cnt++;
  f();
}
```

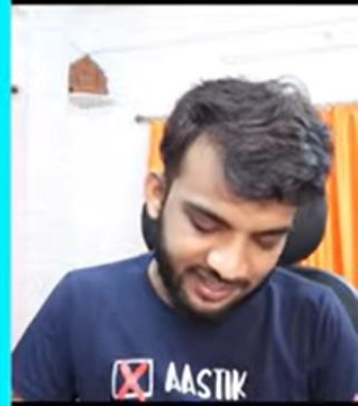
```
f()
{
  if(cnt == 3)
    return
  print(cnt)
  cnt++
  f();
}
```

Output

0

1

2



TUF

```
cnt = 0

f()
{
  if(cnt == 2)
    return;
  print(cnt)
  cnt++
  f()
}

main()
{
  f();
}
```

```
f()
{
  if(cnt == 2)
    return;
  print(cnt)
  cnt++;
  f();
}
```

```
f()
{
  if(cnt == 3)
    return
  print(cnt)
  cnt++
  f();
}
```

Output

0

1

2

TUF

esday, 7 December 2021 15:18

Recursion??

When a function calls itself until a specified condition is met.

```
void f()
{
    print(1)
    f();
}
```

```
void f()
{
    print(1)
    f()
}
```

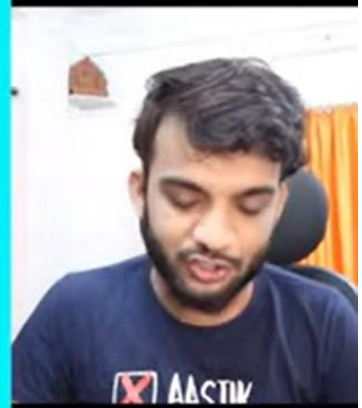
```
void f()
{
    print(1)
    f()
}
```

```
void f(
{
    print(
    f()
}
```

```
main()
{
    f();
```

f()
f()
f()
f()2

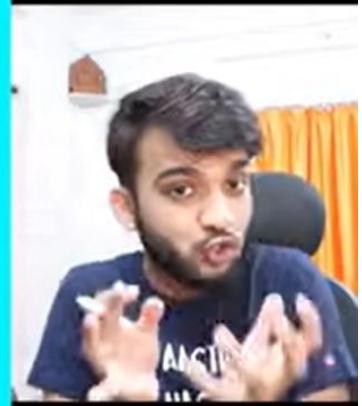
segmentation



TUF

```
f()
{
if (cnt == 3)
return
print (cnt)
cnt++
f();
}
```

```
f()
{
if (cnt == 3)
return;
}
```



TUF

```
f()
{
if (cnt == 3)
return
print (cnt)
cnt++
f();
}
```

```
if (cnt == 3)
return;
```

TUF

```
f()
{
}
main()
{
f();
}
```

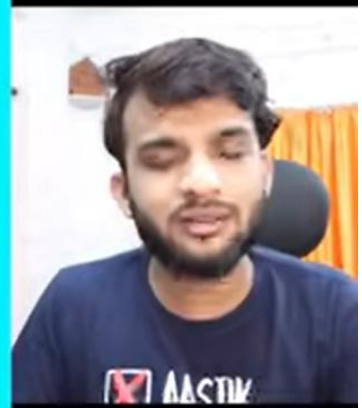
Output

0

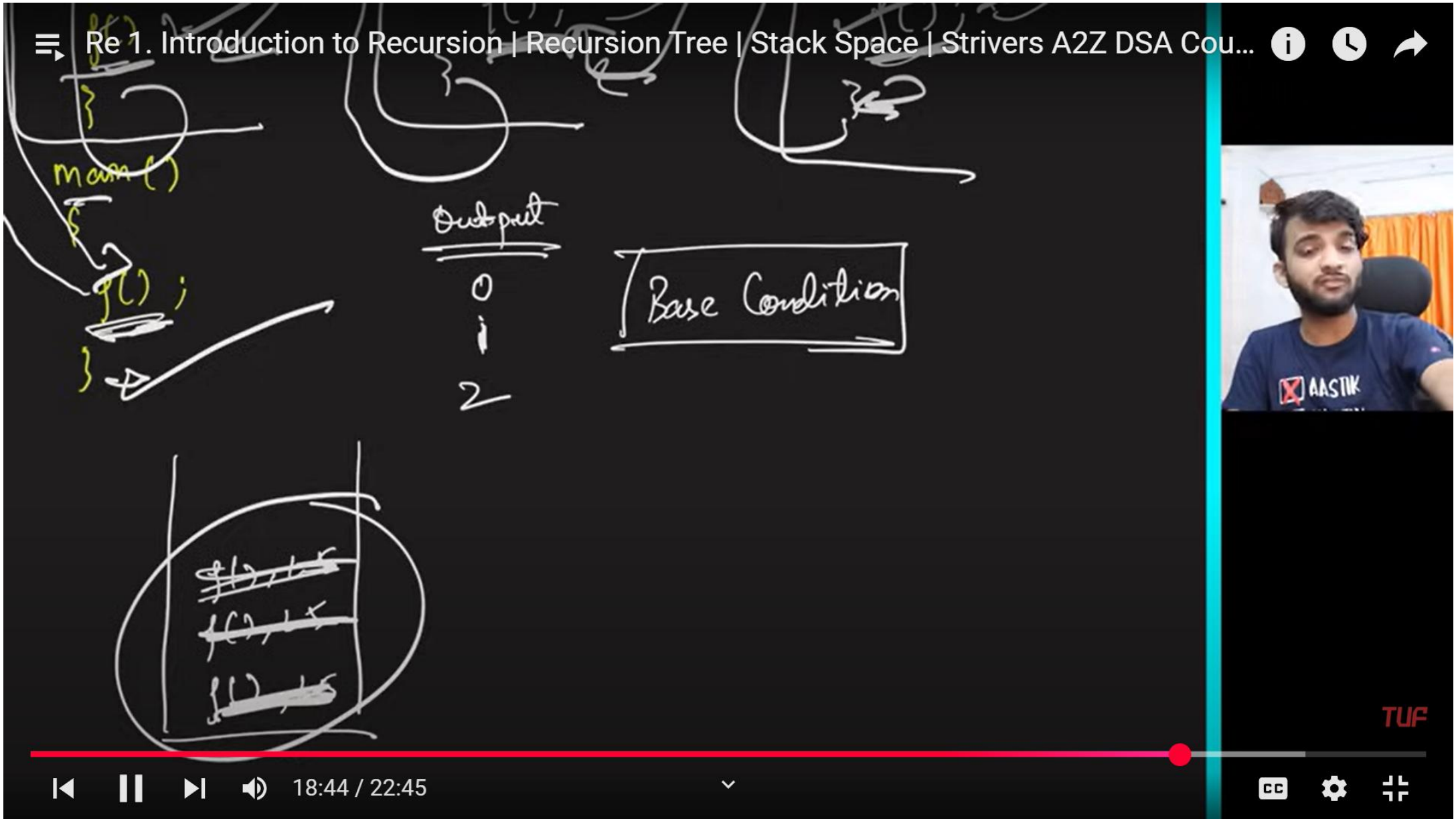
1

2

Base Condition



Re 1. Introduction to Recursion | Recursion Tree | Stack Space | Strivers A2Z DSA Cou...



18:44 / 22:45

code.cpp

```cpp
#include<bits/stdc++.h>
using namespace std;

void print() {
    cout << 1 << endl;
    print();
}

int main() {
    #ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
    #endif

    print();

    return 0;
}
```

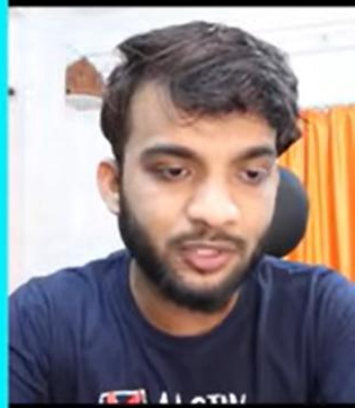
input.txt

```
1
```

output.txt

```
523024  1
523025  1
523026  1
523027  1
523028  1
523029  1
523030  1
523031  1
523032  1
523033  1
523034  1
523035  1
523036  1
523037  1
523038
```

```
bash: line 1: 32711 Segmentation fault: 11  '/Users/striver/Desktop/SublimeFolder/code'
[Finished in 2.6s with exit code 139]
[cmd: ['bash', '-c', "g++ -std=c++14 '/Users/striver/Desktop/SublimeFolder/code.cpp' -o '/Users/striver/Desktop/SublimeFolde
code' && '/Users/striver/Desktop/SublimeFolder/code'"]]
[dir: /Users/striver/Desktop/SublimeFolder]
[path: /usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin]
```



code.cpp

```cpp
#include<bits/stdc++.h>
using namespace std;
int cnt = 0;
void print() {
    if(cnt == 3) return;
    cout << cnt << endl;
    cnt++;
    print();
}

int main() {
    #ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
    #endif

    print();

    return 0;
}
```

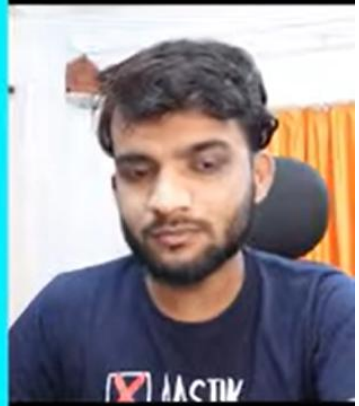
input.txt

```
1
```

output.txt

```
523024  1
523025  1
523026  1
523027  1
523028  1
523029  1
523030  1
523031  1
523032  1
523033  1
523034  1
523035  1
523036  1
523037  1
523038
```

```
bash: line 1: 32711 Segmentation fault: 11  '/Users/striver/Desktop/SublimeFolder/code'
[Finished in 2.6s with exit code 139]
[cmd: ['bash', '-c', "g++ -std=c++14 '/Users/striver/Desktop/SublimeFolder/code.cpp' -o '/Users/striver/Desktop/SublimeFolde
code' && '/Users/striver/Desktop/SublimeFolder/code'"]]
[dir: /Users/striver/Desktop/SublimeFolder]
[path: /usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin]
```



code.cpp

```cpp
#include<bits/stdc++.h>
using namespace std;
int cnt = 0;
void print() {
    if(cnt == 3) return;
    cout << cnt << endl;
    cnt++;
    print();
}

int main() {
    #ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
    #endif

    print();

    return 0;
}
```

input.txt

```
1
```

output.txt

```
0
1
2
```

[Finished in 1.4s]

cnt = ~~2~~ 3

```
f()
{
  if(cnt == 3)
    return;
  print(cnt)
  cnt++
  f()
}
main()
{
  f();
```

```
f()
{
  if(cnt == 2)
    return;
  print(cnt)
  cnt++;
  f();
}
```

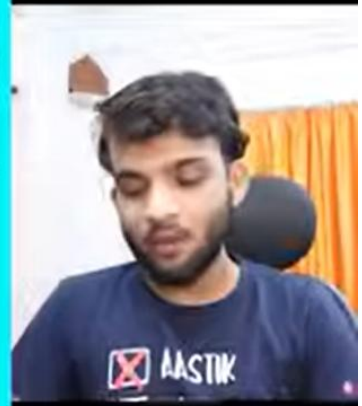
```
f()
{
  if(cnt == 3)
    return
  print(cnt)
  cnt++
  f();
}
```

Output

0

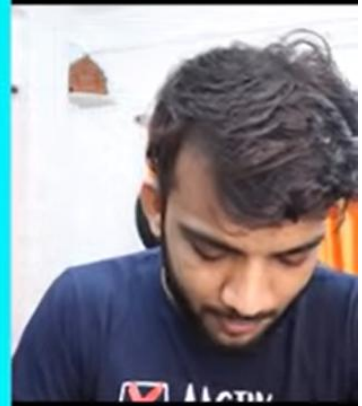
1

Base Condition



TUF

Recursion Tree



TUF

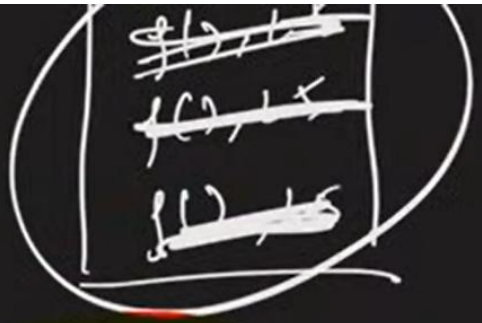
```
f()
{
if (cnt == 3)
return
print (cnt)
cnt ++
f();
}
```

```
f()
{
if (cnt == 3)
return;
}
```

```
}
```

Condition

TUF



Recursion Tree

1

TUF

main()

f();

}

Output

0

1

2

Base Condition

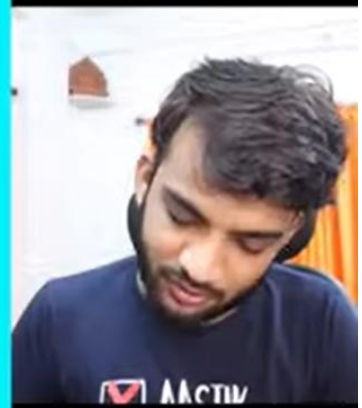
f()

f()

f()

Recursion Tree

f()



TUF

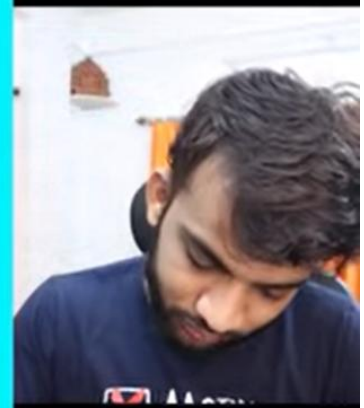
Recursion Tree

f()

f()

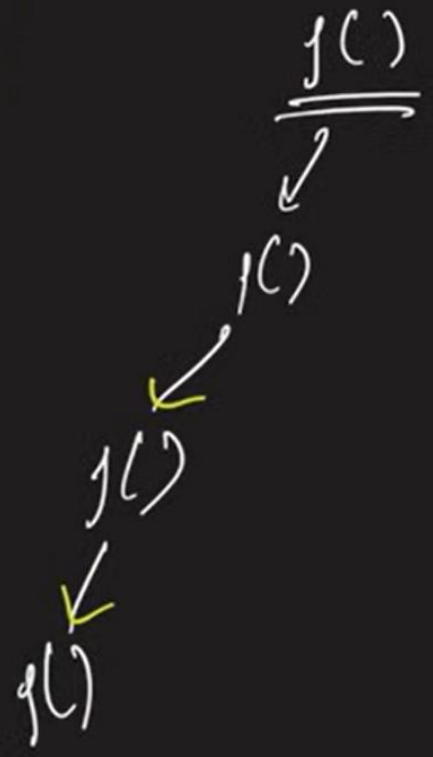
f()

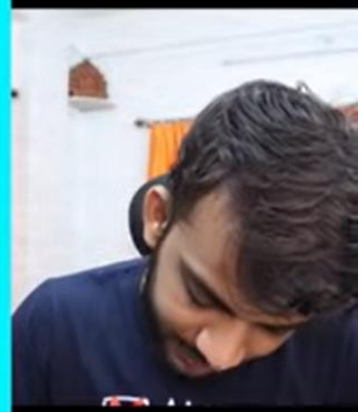
f()



TUF

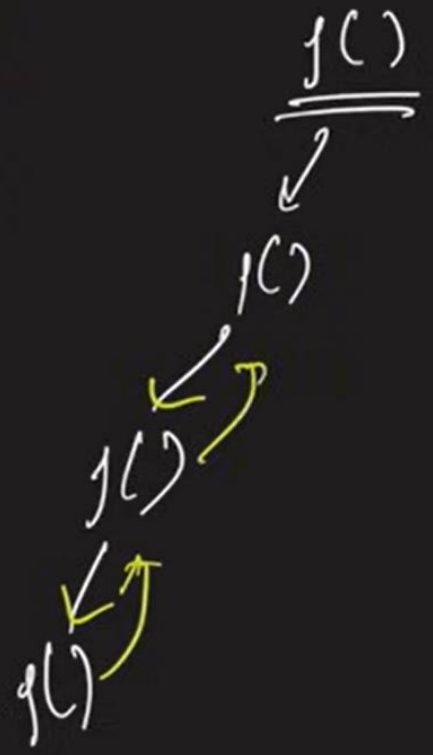
Recursion Tree

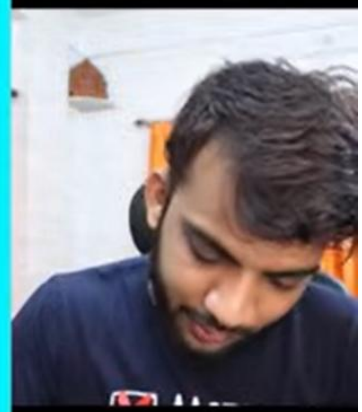




TUF

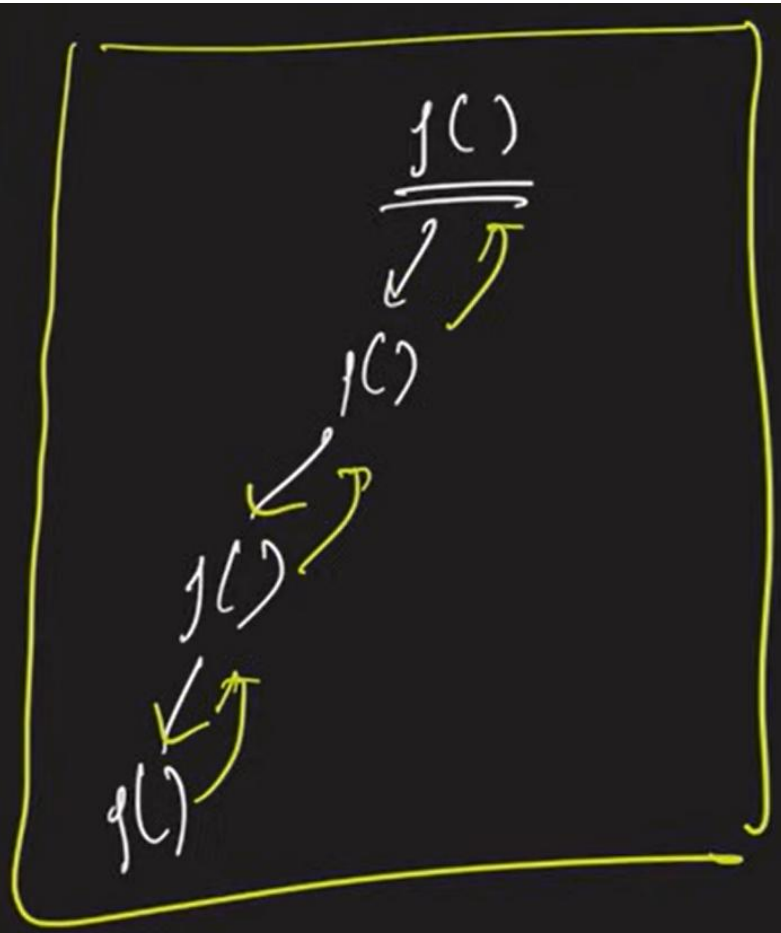
Recursion Tree

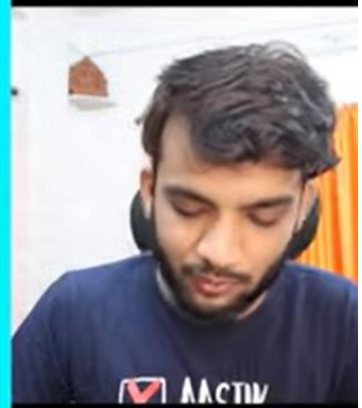




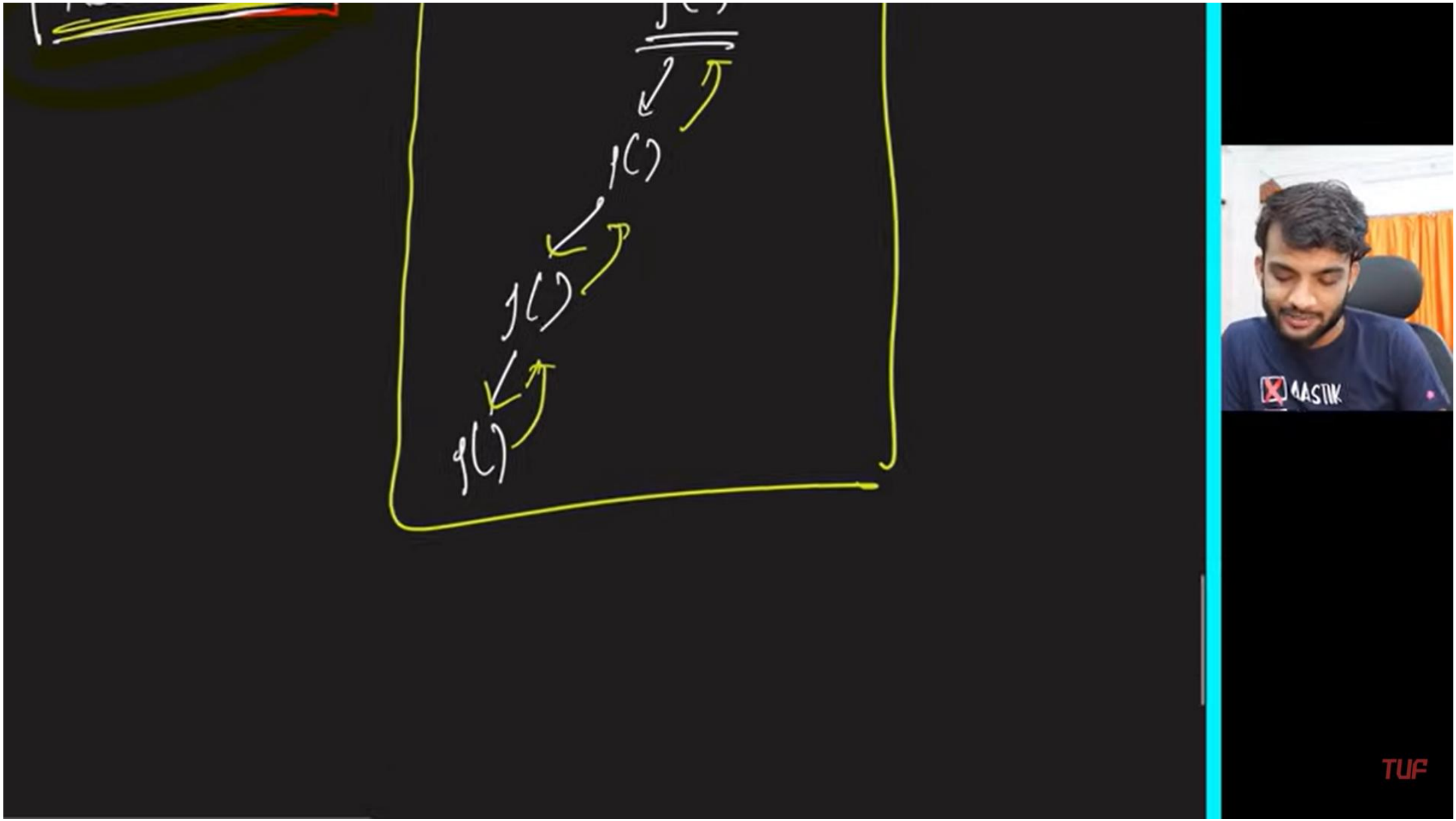
TUF

Recursion Tree





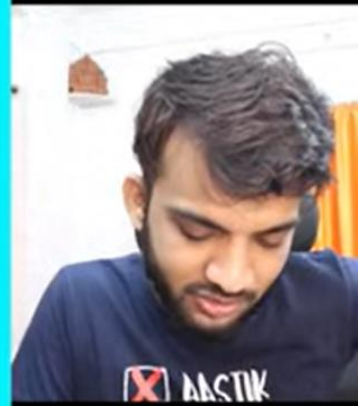
TUF


f()
f()
f()
f()
MASTIK
TUF

$f(\ )$

→ Recursion

→ Base



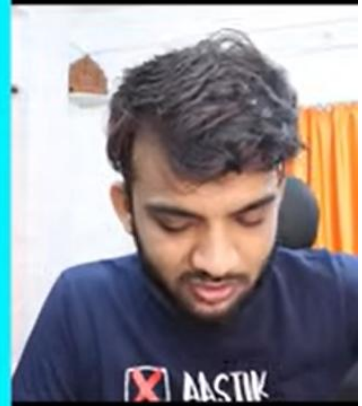
TUF

$f(\ )$

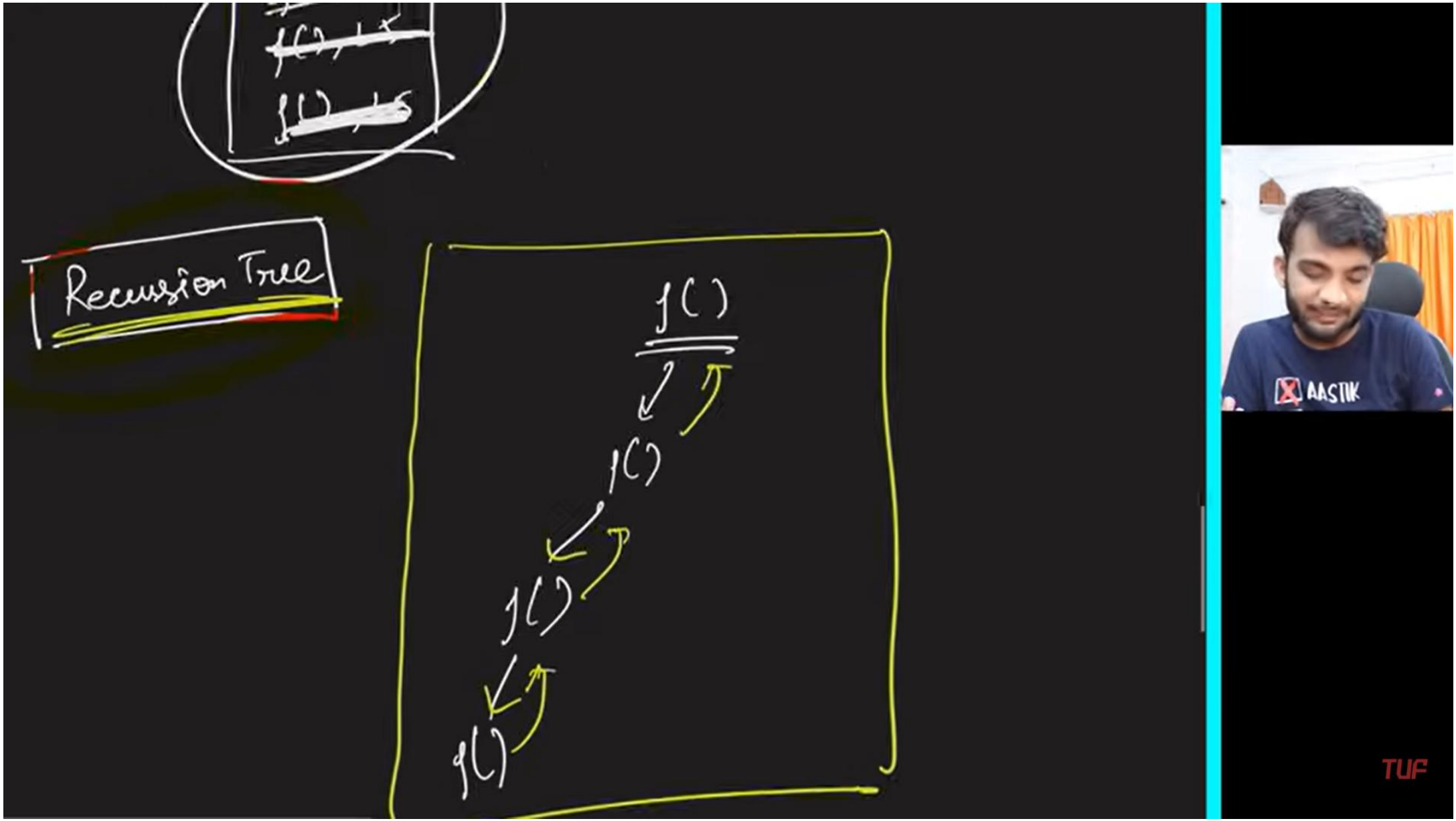
→ Recursion

→ Base Case

→ Stack overflow



TUF

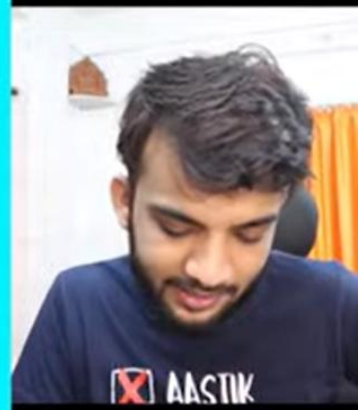

Recursion Tree
f()
f()
f()
f()
AASTIK
TUF

→ Recursion

→ Base Case

→ Stack overflow | Stack space

→ '



TUF

→ Recursion

→ Base Case

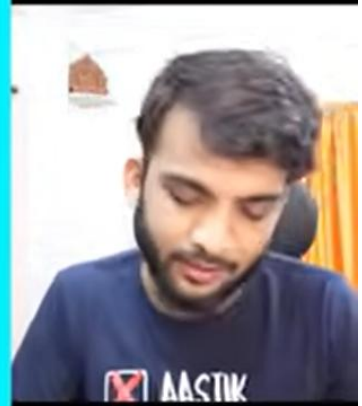
→ Stack overflow | Stack space

→ Recursion Tree

TUF

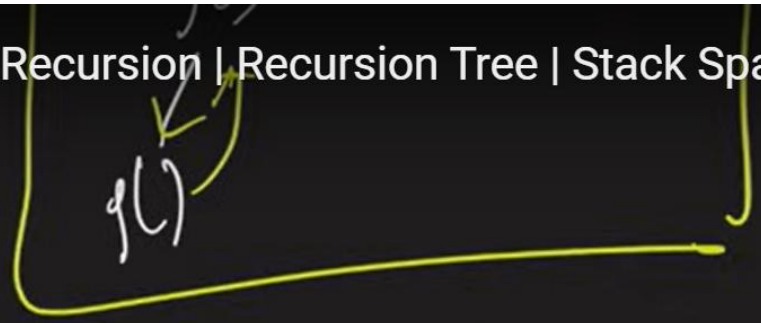
g()

- → Recursion
- → Base Case
- → Stack overflow | Stack space
- → Recursion Tree

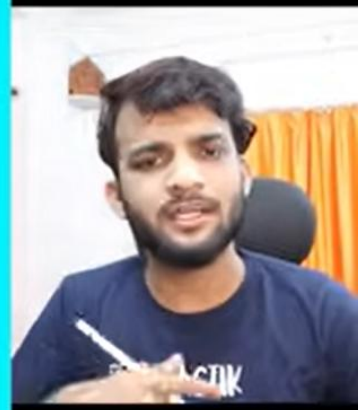


TUF

Re 1. Introduction to Recursion | Recursion Tree | Stack Space | Strivers A2Z DSA Cou...



- → Recursion
- → Base Case
- → Stack overflow | Stack space
- → Recursion Tree



TUF

22:06 / 22:45

g()
Instagram
striver_79
149 posts
8913 followers
468 following
Following
Tree
@STRIVER_79
RAJ
TUF

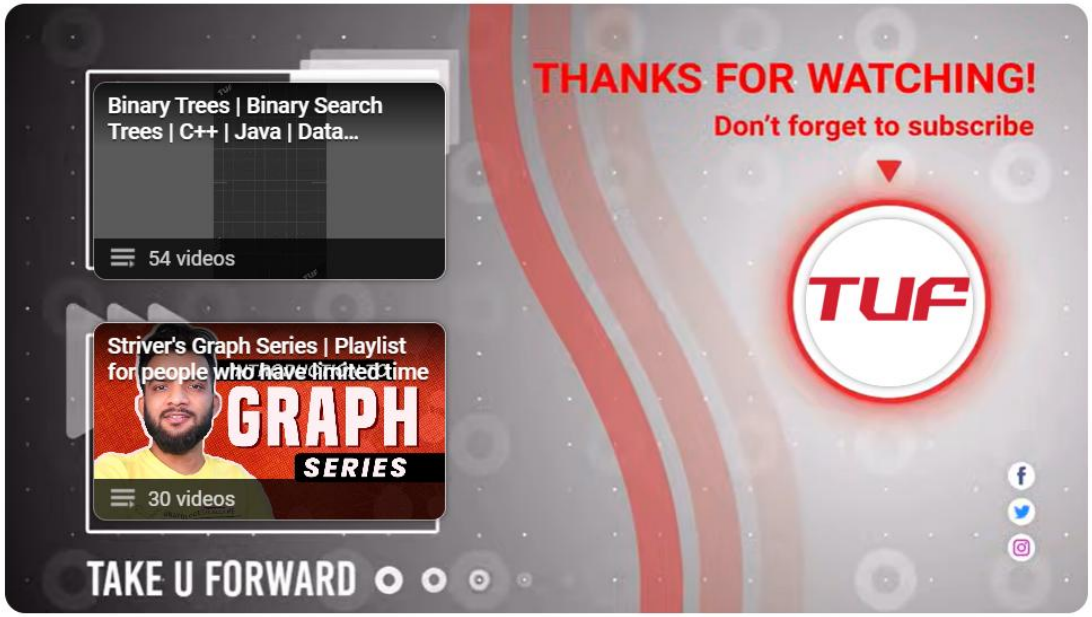
youtube.com/watch?v=yVdKa8dnKiE&list=PLgUwDviBIf0rGlzIn_7rsaR2FQ5e6ZOL9

Gmail | YouTube | Maps | Finalize Resume - Z... | Novorésumé | Accolite / ClassRoo... | Is there a lab trainin... | Technology Timesh...

YouTube IN

recursion striver

Sign in



Re 1. Introduction to Recursion | Recursion Tree | Stack Space | Strivers A2Z DSA Course

take U forward | Subscribe | 29K | Share

https://www.youtube.com/watch?v=WBgsABoClE0&list=PLgUwDviBIf0rGlzIn...

Recursion (Basics to Advanced) and ...

take U forward - 1 / 22

18 L16. M-Coloring Problem | Backtracking — take U forward (24:37)

19 L17. Palindrome Partitioning | Leetcode | Recursion | C++ |... — take U forward (24:34)

20 L19. Rat in A Maze | Backtracking — take U forward (25:10)

21 L18. K-th Permutation Sequence | Leetcode

Show chat replay

Re 2. Problems on Recursion | Strivers A2Z DSA Course

ENG 10:20 20-01-2025