

Subject

موضوع الدرس

Date / /

التاريخ / /

## Recursion

(8) مل

Type Fun (Param) {

if (<base condition>){

1-

2- Fun (Param) ;

3-

Recursion

}

}

function call itself

{ (n+1) until biv }

{ (0 < n) } }

ex

void fun (int n) {

if (n > 0) { ← recursion

printf ("%d", n);

fun (n-1);

}

{ (0 < n) }

void main() {

int x = 3;

fun(x);

}

الرسالة

↓  
x ↑ value  
↓  
main ( )

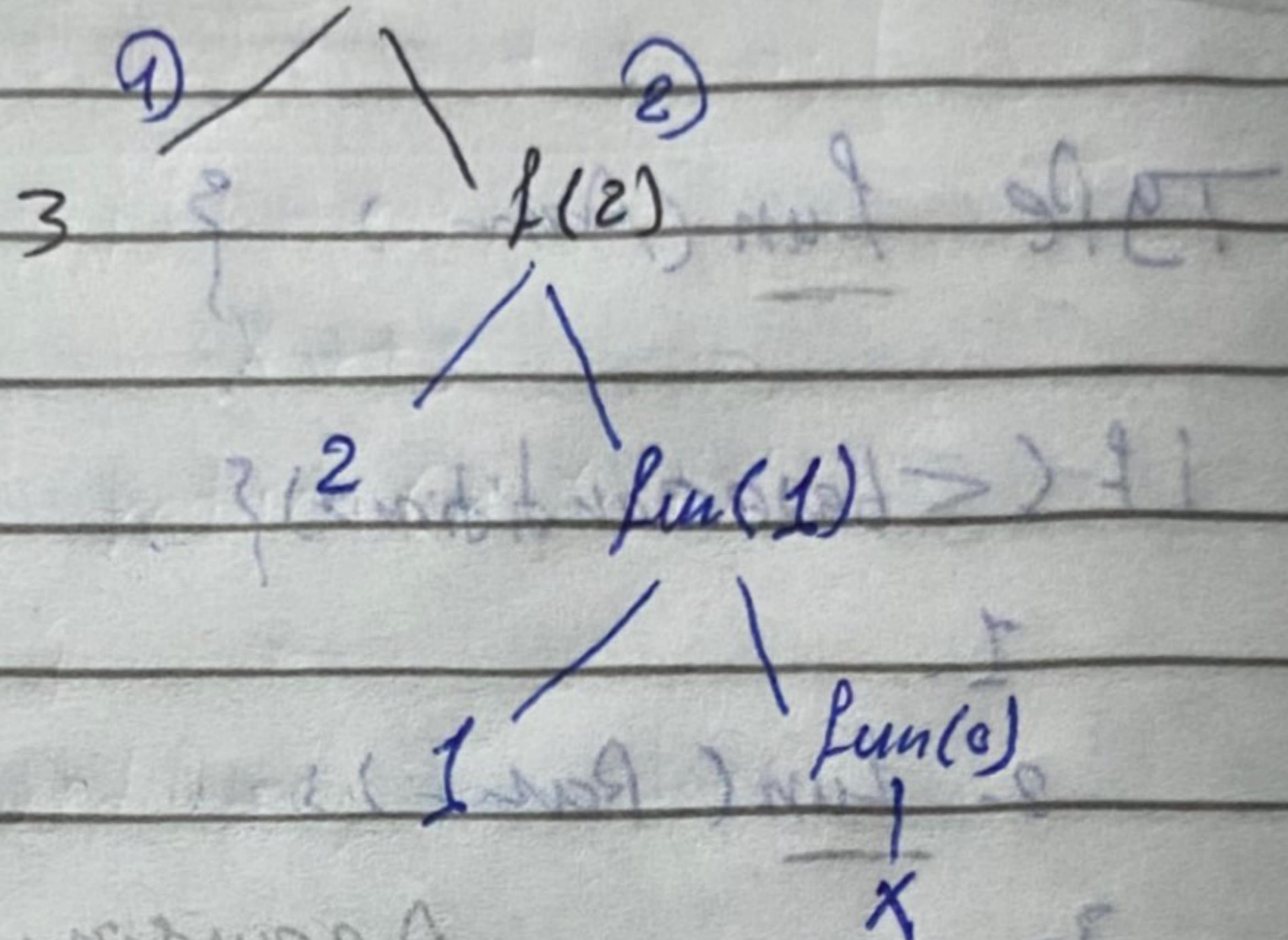
Subject

موضوع الدرس

Date / /

التاريخ / /

fun(3)



ex<sup>2</sup>

Void fun(int n){

if (n>0){

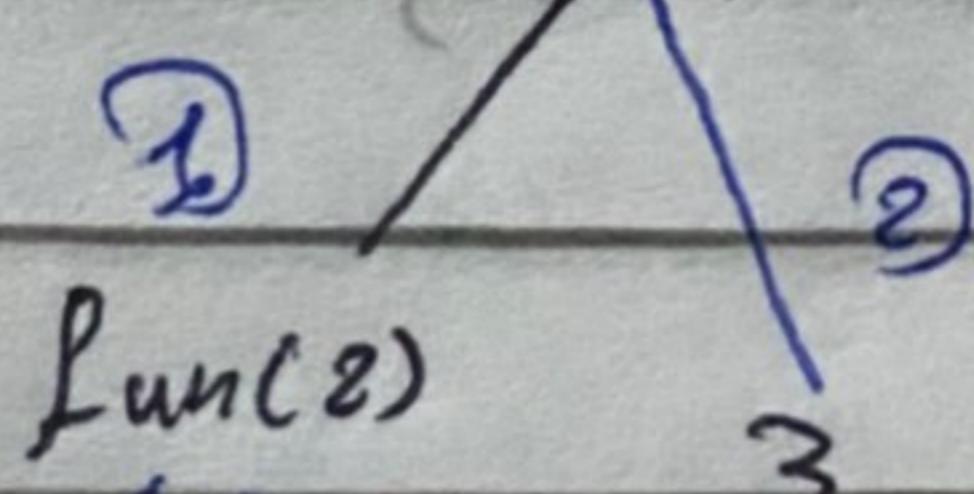
\* fun(n-1);

\* printf("%d", n);

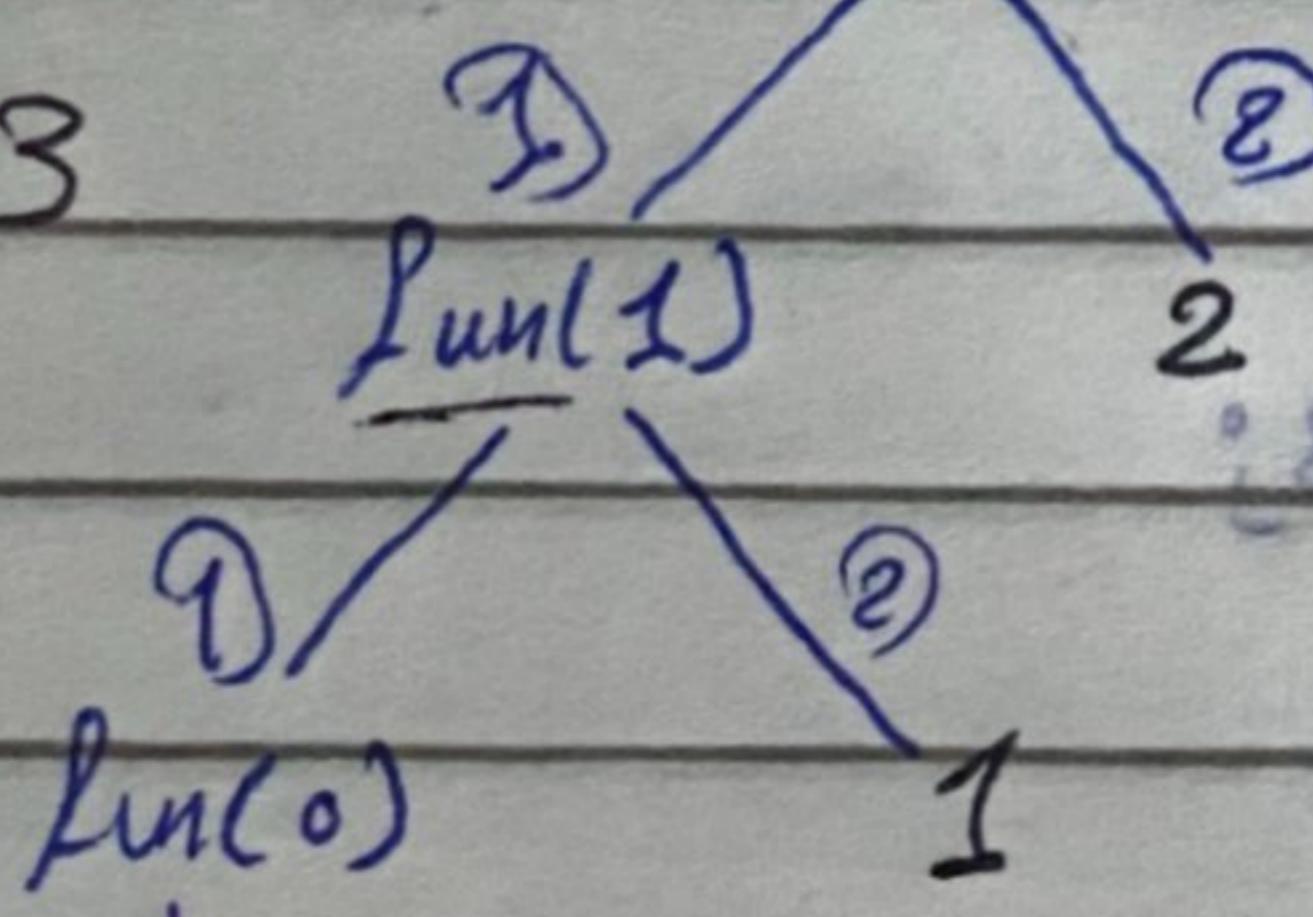
}

}

Fun(3)



outPut 123



① Should be done before start to do ②

Subject

موضوع الدرس

Date

التاريخ

How many calls  $\rightarrow$  Stack = 14

size of the memory =  $n = 4$

Static Variables in Recursion

~~int funct(int n)~~

~~If ( $n > 0$ ) {~~

~~return fun(n-1) + n;~~

~~return a;~~

~~}~~

Stack = (2)

~~int main () {~~

~~int a = 5~~

~~printf("%d", fun(a));~~

~~}~~

Stack = (5)

fun(5) = 15

fun(3) + 5 = 10

fun(2) + 3 = 6

fun(1) + 2 = 3

fun(0) + 1 = 1

int funC(int n) {

\* static int x = 0; just one time

if (n > 0) {

x++;

return (n - 1) + x;

}

return 0;

}

$$f(5) = \underline{\underline{25}}$$

we will add the final  $x = 5$

$$f(4) + \underline{\underline{5}} = 25$$

$$f(3) + \underline{\underline{5}} = 20$$

$$f(2) + \underline{\underline{5}} = 15$$

$$f(1) + \underline{\underline{5}} = 10$$

$$f(0) + \underline{\underline{5}} = 5$$

If we do that

int x = 0;

int funC(int n) {

the result will be same

Subject

موضوع الدرس

Date

التاريخ

## Tail Recursion replace with loop ✓

Void fun(int n){

if(n>0){

printf("%d", n),

Tail Recursion → \* fun(n-1);

} (return value)

} (return value)

كل حاصل على ترتيب دفعات ممكن حاصد ممكّن وفقاً

Recursive tree.

## Head Recursion

use with loop X

Void fun(int n){

If(n>0){

\* fun(n-1);

{int i=1;

while(i<=n)

printf(

, i++;

};

الاول نفذها خشوف العمليات

Processes will do at returning time

Subject

موضوع الدرس

Date / /

التاريخ / /

The Recursion, The function calls itself more than 1 time

Void fun( int n){

if( n > 0)

{ " " ) + + n

Printf("%d", n);

\* fun(n-1);

\* fun(n-1);

}

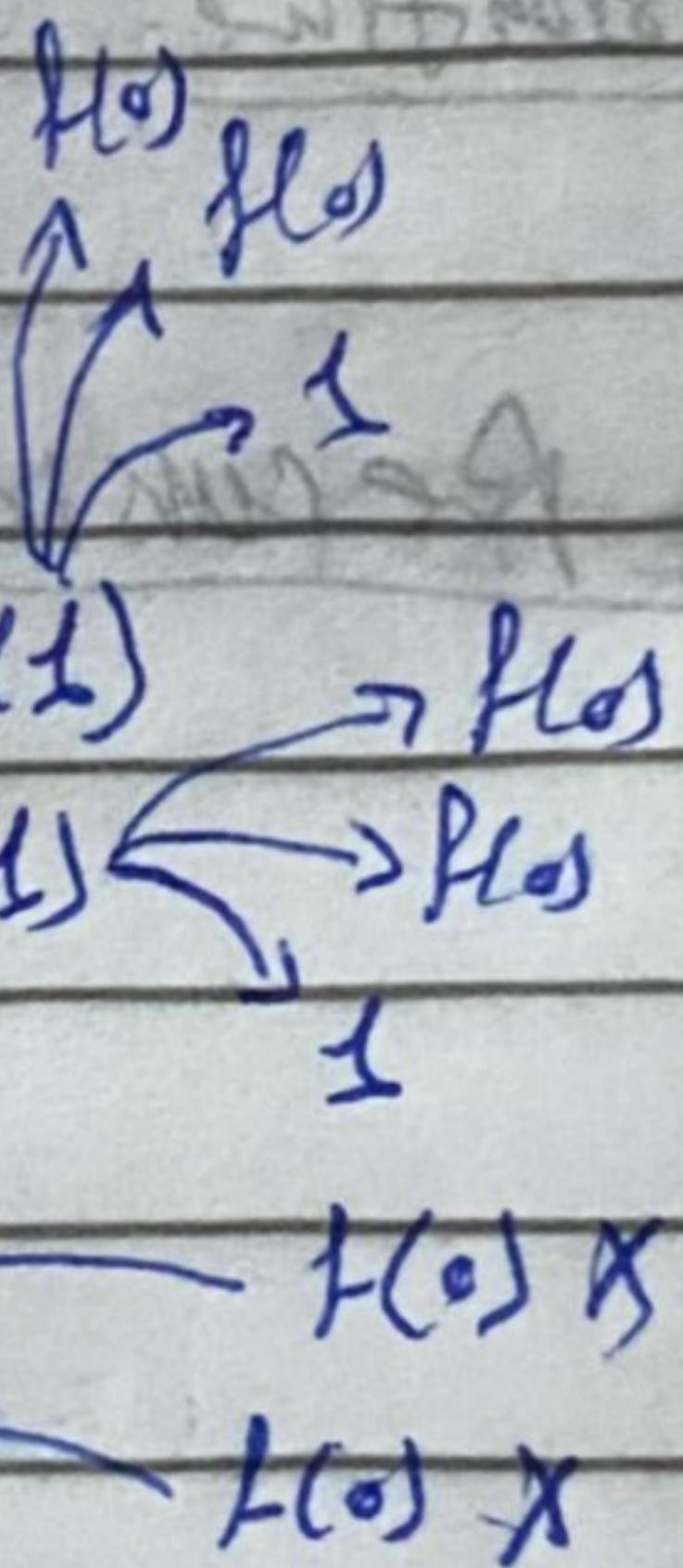
f(3)

f(2)

f(1)

f(0)

15 call



f/P : 3, 2, 1, 1, 2, 1, 1

أندلسية

Subject \_\_\_\_\_

موضوع الدرس \_\_\_\_\_

Date / /

التاريخ / /

D

Fun(3)

3  
3

2  
2

3  
3

Fun(2)

2  
3

3  
3

2  
2

3  
3

1  
1

2  
2

3  
3

1  
1

2  
2

Fun(1)  
1

1  
1

1  
1

with Black the steps

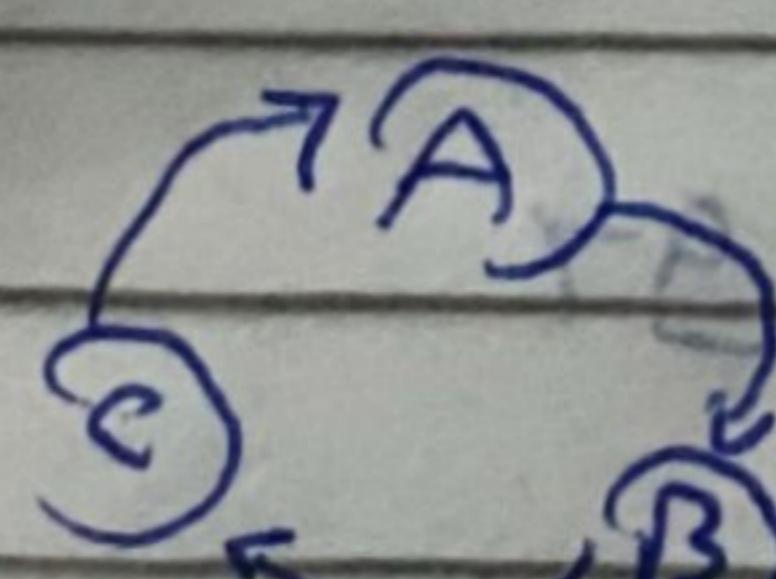
(22) NDT

with Blue the calls

(22+22) NDT

Indirect Recursion

2 functions call themself



(((22+22) NDT) NDT)

II

((22) NDT)

Subject \_\_\_\_\_

موضوع الدرس \_\_\_\_\_

Date: / /

التاريخ / /

## Nested Recursion

void fun(int n) {

if (n > 0) {

=

fun(fun(n-1));

Recursive

Ex:

int fun(int n) {

if (n > 100)

fun(95)

return n-100

else

fun(fun(95+1));

return fun(fun(n+1));

" 96

}

fun(98)

fun(95); to with n

fun(fun(fun(98+1))); 97

"

fun(97)

like,