CS & IT ENGINEERING

Programming in C

Functions and Storage Classes

Lec-05

By- Pankaj Sharma sir

Recursion

Q.19

What does the following function do?

```
int fun(int x, int y)
{
  if (y == 0)   return 0;
  return (x + fun(x, y-1));
}
```

A. x + y

B. x + x * y

C. x * y

D. pow(x, y)

$x + fun(x, y-1)$

$x + x + fun(x, y-2)$

$= 2x + fun(x, y-2)$

$3x + fun(x, y-3)$

$y+x$ $\quad$ $y-y$

fun(5, 4) — x = 5, y = 4 — 20

fun(5, 3) — x = 5, y = 3 — 15

fun(5, 2) — x = 5, y = 2 — 10

fun(5, 1) — x = 5, y = 1 — 5

$5 - 1 + fun(x, 0)$ — y = 0

Q.20

What does fun2() do in general?

```
int fun(int x, int y)
{
    if (y == 0)  return 0;
    return (x + fun(x, y-1));
}

int fun2(int a, int b)
{
    if (b == 0) return 1;
    return fun(a, fun2(a, b-1));
}
```

A. x * y

B. x + x * y

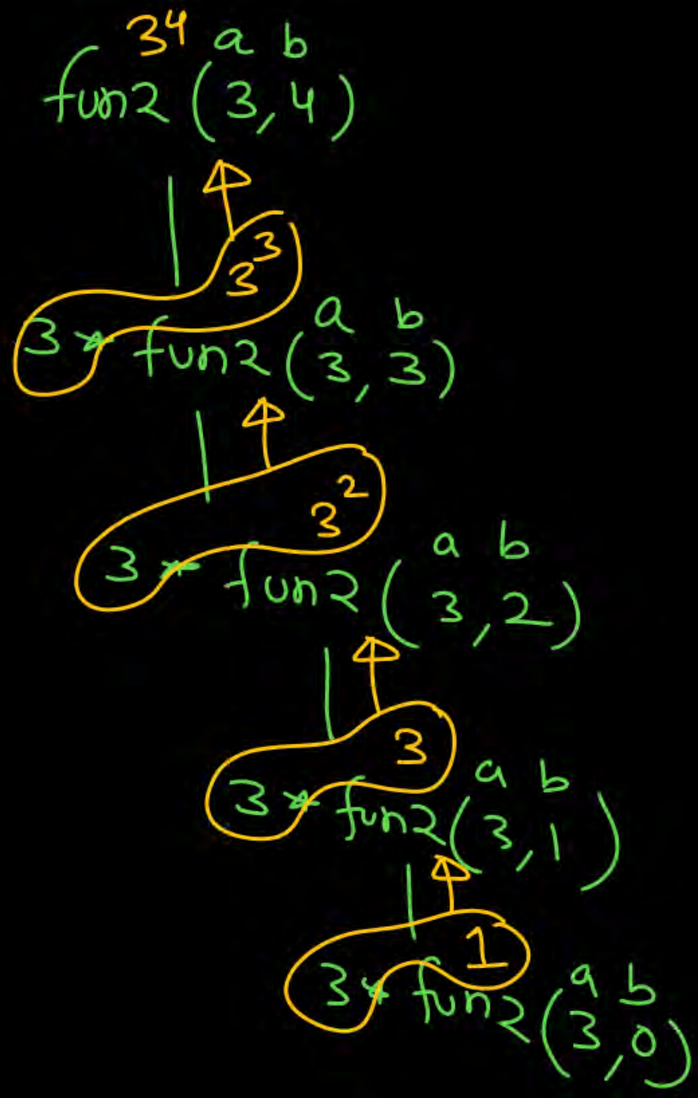
C. pow(x, y)

D. pow(y, x)

fun(x,y) ⟹ x+y

```
int fun2(int a, int b)
{
    if(b==0)
        return 1

    return a*fun2(a,b-1);
}
```



Q.20

What does fun2() do in general?

```
int fun(int x, int y)
{
   if (y == 0)   return 0;
   return (x + fun(x, y-1));
}

int fun2(int a, int b)
{
   if (b == 0) return 1;
   return fun(a, fun2(a, b-1));
}
```

fun(x,y) = x*y

A. x * y

B. x + x * y

C. pow(x, y)

D. pow(y, x)

Q.21

Output of following program?

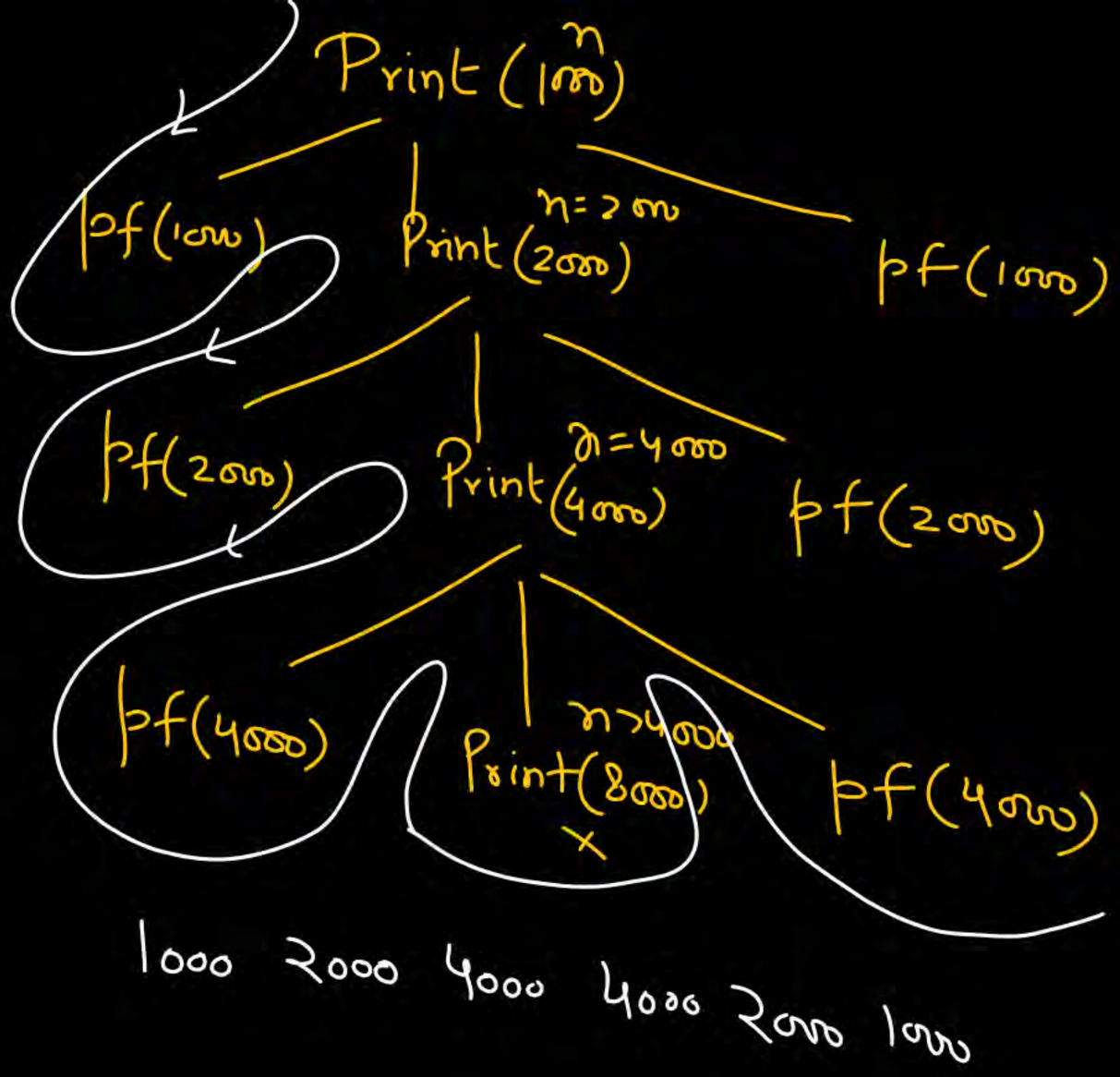
```
#include<stdio.h>
void print(int n){
    if (n > 4000)
    return;
    printf("%d ", n);
    print(2*n);
    printf("%d ", n);
}
int main()
{
    print(1000);
    [redacted]
    return 0;
}
```

A. 1000 2000 4000

B. 1000 2000 4000 4000 2000 1000

C. 1000 2000 4000 2000 1000

D. 1000 2000 2000 1000


Print(1000)
n
pf(1000)
Print(2000)
n=2000
pf(1000)
pf(2000)
Print(4000)
n=4000
pf(2000)
pf(4000)
Print(8000)
n>4000
pf(4000)
1000 2000 4000 4000 2000 1000

Q.22

What does the following function do?

```
int fun(unsigned int n)
{
    if (n == 0 || n == 1)
        return n;

    if (n%3 != 0)
        return 0;
    return fun(n/3);
}
```

$f(1) = 1$

$f(3) = f(1) = 1$

$f(6) = f(2) = 0$

$f(3^k) = f(3^{k-1}) = f(3^{k-2}) \cdots = f(3^1)$

$= f(1)$

$= 1$

1, 2, 4, 5, 7, 8, ...

workout → examples

A. It returns 1 when n is a multiple of 3, otherwise returns 0

B. It returns 1 when n is a power of 3, otherwise returns 0

C. It returns 0 when n is a multiple of 3, otherwise returns 1

D. It returns 0 when n is a power of 3, otherwise returns 1

① $3n+1$ ⇒ remainder ⇒ 1

② $3n+2$ ⇒ rem = 2

③ $3n$

Q.23

Predict the output of following program

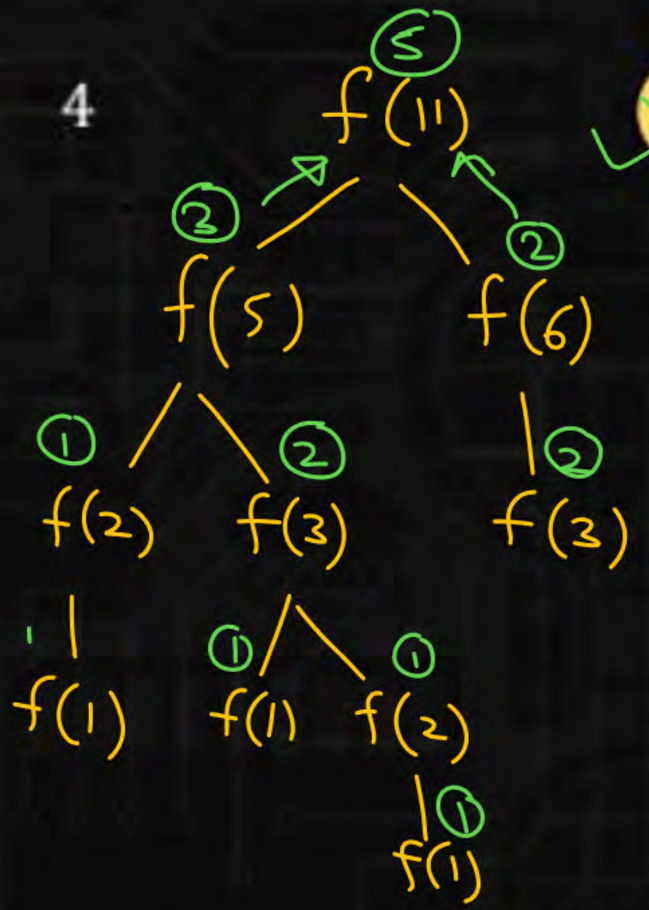
```
#include <stdio.h>
int f(int n)
{
    if(n <= 1)
        return 1;
    if(n%2 == 0)
        return f(n/2);
    return f(n/2) + f(n/2+1);
}
 int main()
{
    printf("%d", f(11));
    return 0;
}
```

A. Stack Overflow

B. 3

C. 4

D. 5



Q.24

Consider the following C function:

```
int f(int n)
{
  static int i = 1;
  if (n >= 5)
    return n;
  n = n+i;
  i++;
  return f(n);
}
```

The value returned by f(1) is

A. 5

B. 6

C. 7

D. 8

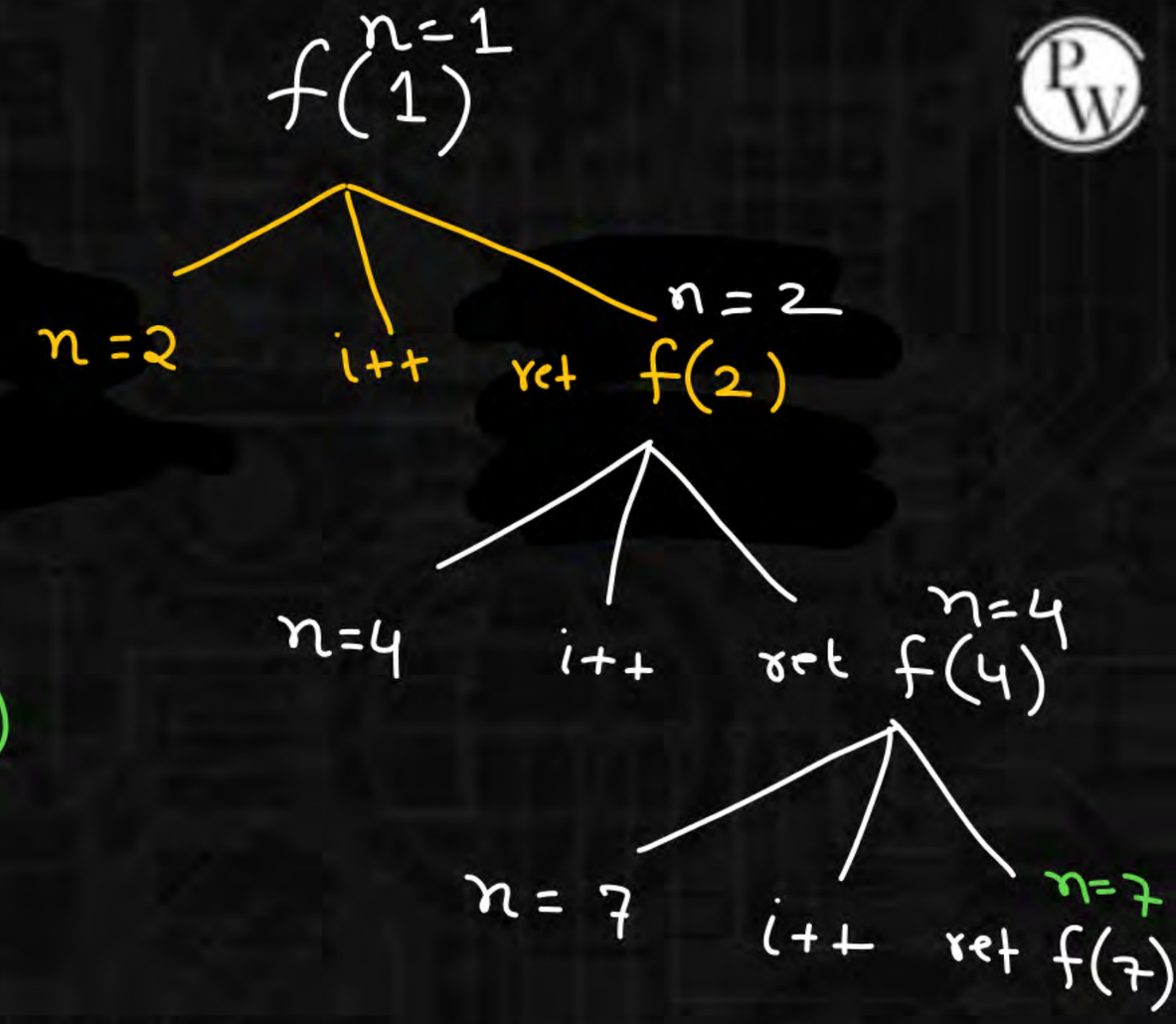
Q.24

Consider the following C function:

```
int f(int n)
{
  static int i = 1;
  if (n >= 5)
    return n;
  n = n+i;
  i++;
  return f(n);
}
```

The value returned by f(1) is

i 1 2 3 4



7

Q.25

Consider the following C function.

```
int fun (int n)
{
 int x=1, k;
 if (n==1) return x;
 for (k=1; k<n; ++k)
 x = x + fun(k) * fun(n – k);
 return x;
}
```

The return value of fun(5) is _____.

A. 0

B. 26

C. 51

D. 71

Q.25

Consider the following C function.

```
int fun (int n)
{
  int x=1, k;
  if (n==1) return x;
  for (k=1; k<n; ++k)
  x = x + fun(k) * fun(n – k);
  return x;
}
```

The return value of fun(5) is _____.

local var.

$fun(1) = 1$

$fun(2)$ $n = 2$

for(k=1; k<2; k++)

$x = x + fun(k) * fun(n-k);$

$x = 1 + fun(1) * fun(2-1)$

$x = 1 + 1 \times 1 = 2$

$fun(2) \Rightarrow 2$

Q.25

Consider the following C function.

```
int fun (int n)
{
  int x=1, k;
  if (n==1) return x;
  for (k=1; k<n; ++k)
  x = x + fun(k) * fun(n – k);
  return x;
}
```

The return value of fun(5) is _____.

local var.

fun(1) = 1

fun(2) n = 2

for(k=1; k<2; k++)

x = x + fun(k) * fun(n-k);

fun(3) loop

(i) k = 1

x = 1 + fun(1) * fun(2)

x = 1 + 1×2 = 3

(ii) k = 2

x = 3 + fun(2) * fun(1)

= 3 + 2×1 = 5

Q.25

Consider the following C function.

```
int fun (int n)
{
  int x=1, k;
  if (n==1) return x;
  for (k=1; k<n; ++k)
  x = x + fun(k) * fun(n – k);
  return x;
}
```

local var.

The return value of fun(5) is _____.

fun(1) = 1

fun(2) n = 2

for(k=1; k<2; k++)

x = x + fun(k) * fun(n-k);

fun(4), x = 1

(i) k = 1

x = 1 + fun(1) * fun(3) = 1 + 1 × 5 = 6

(ii) k = 2

x = 6 + fun(2) * fun(2) = 10

(iii) k = 3

x = 10 + fun(3) * fun(1) = 15

Q.25

Consider the following C function.

```
int fun (int n)
{
  int x=1, k;
  if (n==1) return x;
  for (k=1; k<n; ++k)
  x = x + fun(k) * fun(n – k);
  return x;
}
```

local var.

The return value of fun(5) is 51.

fun(1) = 1

fun(2) n = 2

for(k=1; k<2; k++)

x = x + fun(k) * fun(n-k);

fun(5)

x = 1

(i) k=1 x = 1 + fun(1) * fun(4) = 16

(ii) k=2 x = 16 + fun(2) * fun(3) = 26

(iii) k=3 x = 26 + fun(3) * fun(2) = 36

(iv) k=4 x = 36 + fun(4) * fun(1) = 51

Q.25

Consider the following C function. (best)

```
int fun (int n)
{
  int x=1, k;
  if (n==1) return x;
  for (k=1; k<n; ++k)
  x = x + fun(k) * fun(n – k);
  return x;
}
```

local Var.

The return value of fun(5) is ____51____.

A. 0

B. 26

C. 51

D. 71

fun(1) = 1
fun(2) = 2
fun(3) = 1 + fun(1)*fun(2) + fun(2)*fun(1) (5)
fun(4) = 1 + fun(1)*fun(3) + fun(2)*fun(2) + fun(3)*fun(1) (1×5, 2×2, 5×1)
= 15
f(5) = 1 + fun(1)*fun(4) + fun(2)*fun(3) + fun(3)*fun(2) + fun(4)*fun(1) = 51

Q.26

Consider the following recursive C function. If get(6) function is being called in main() then how many times will the get() function be invoked before returning to the main()?

PYQ

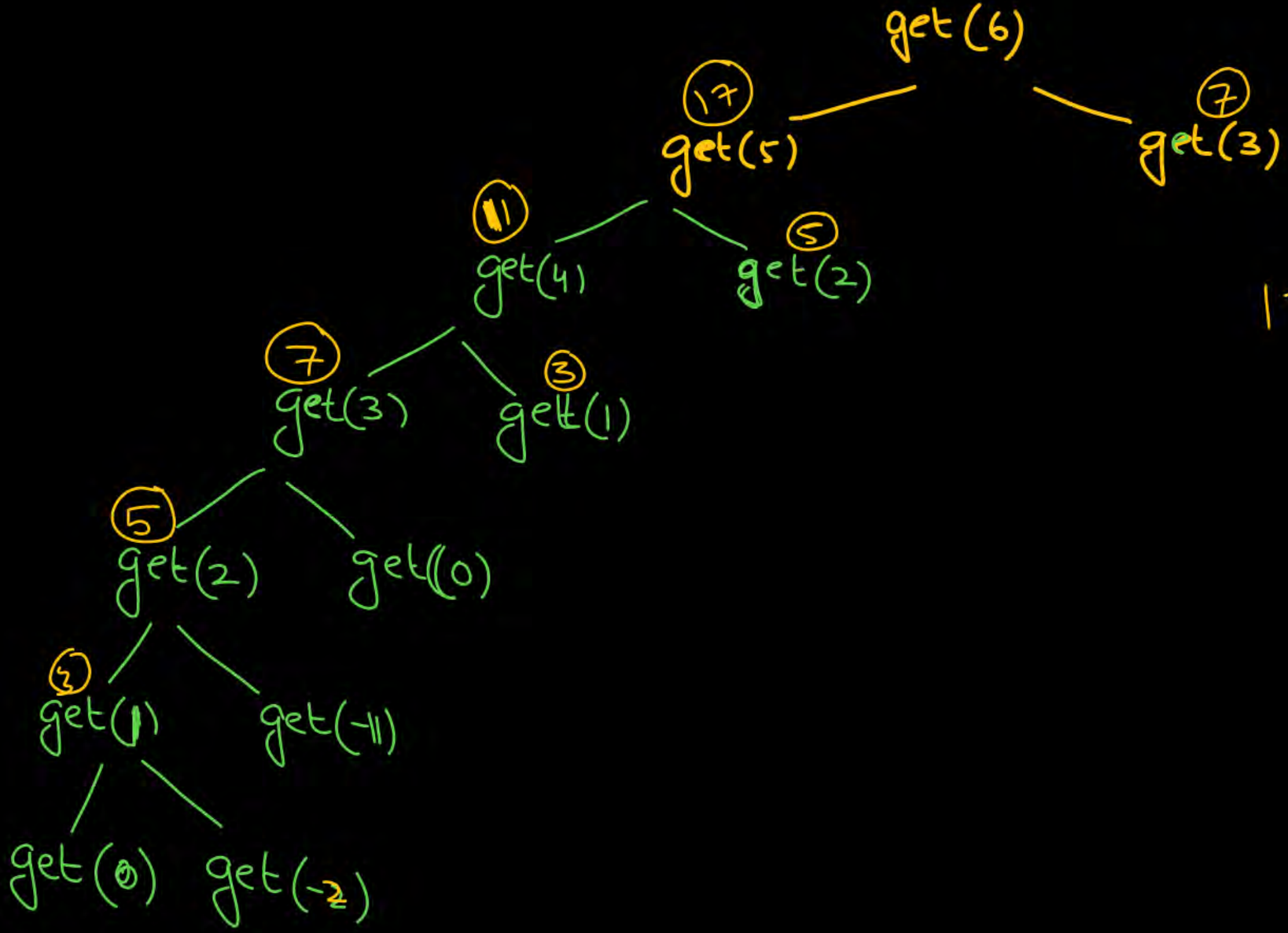
```
void get (int n)
{
  if (n < 1) return;
  get(n-1);
  get(n-3);
  printf("%d", n);
}
```

A. 15

B. 25

C. 35

D. 45


get(6)
17
get(5)
7
get(3)
11
get(4)
5
get(2)
7
get(3)
3
get(1)
5
get(2)
get(0)
3
get(1)
get(-1)
get(0)
get(-2)

17 + 7 + 1

= 25

Q.27

What will be the output of the following C program?

```
void count(int n)
{
   static int d = 1;
   printf("%d ", n);
   printf("%d ", d);
   d++;
   if(n > 1) count(n-1);
   printf("%d ", d);
}
int main()
{
   count(3);
}
```

यह recursive call का part है

A. 3 1 2 2 1 3 4 4 4

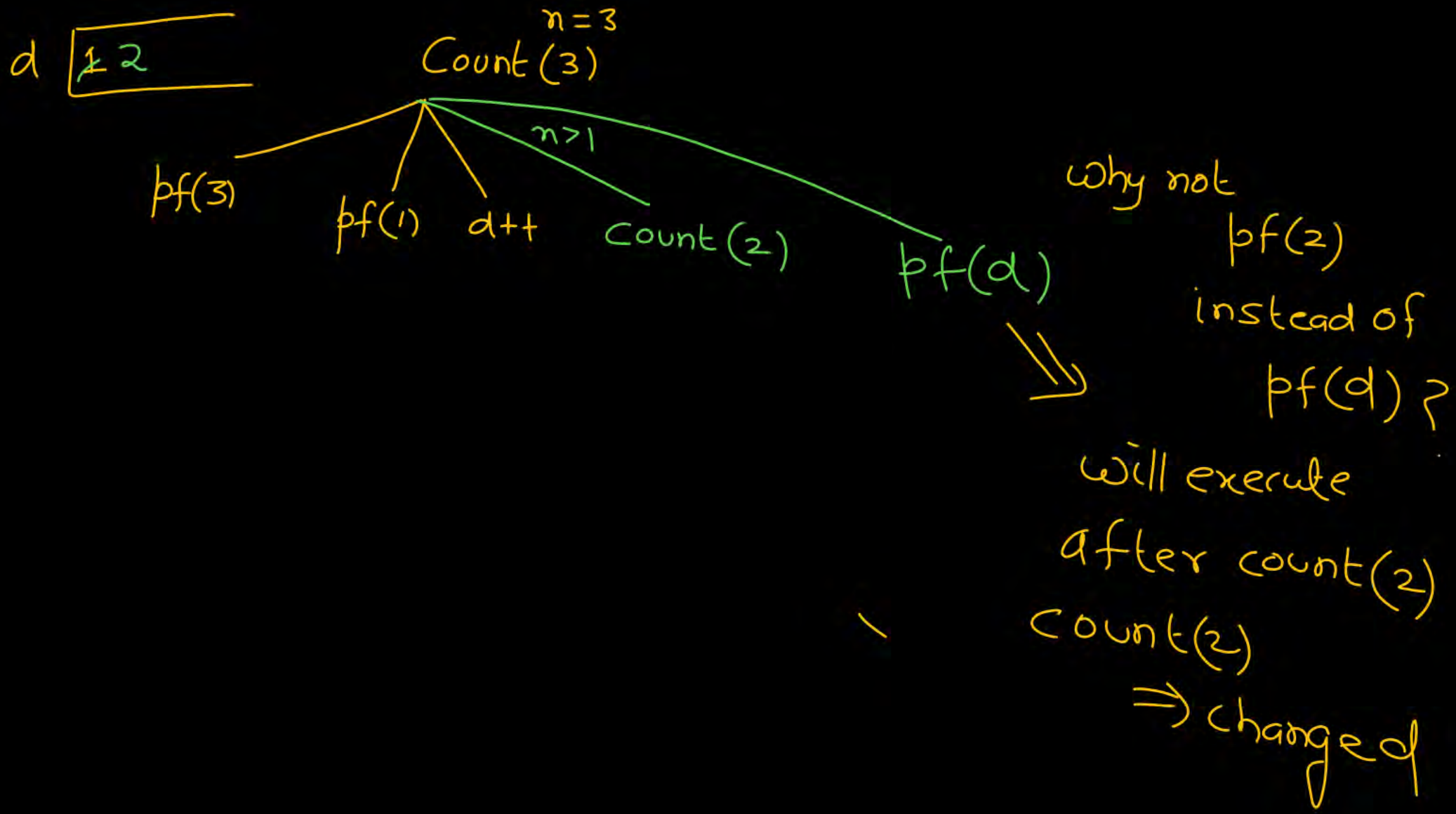
B. 3 1 2 1 1 1 2 2 2

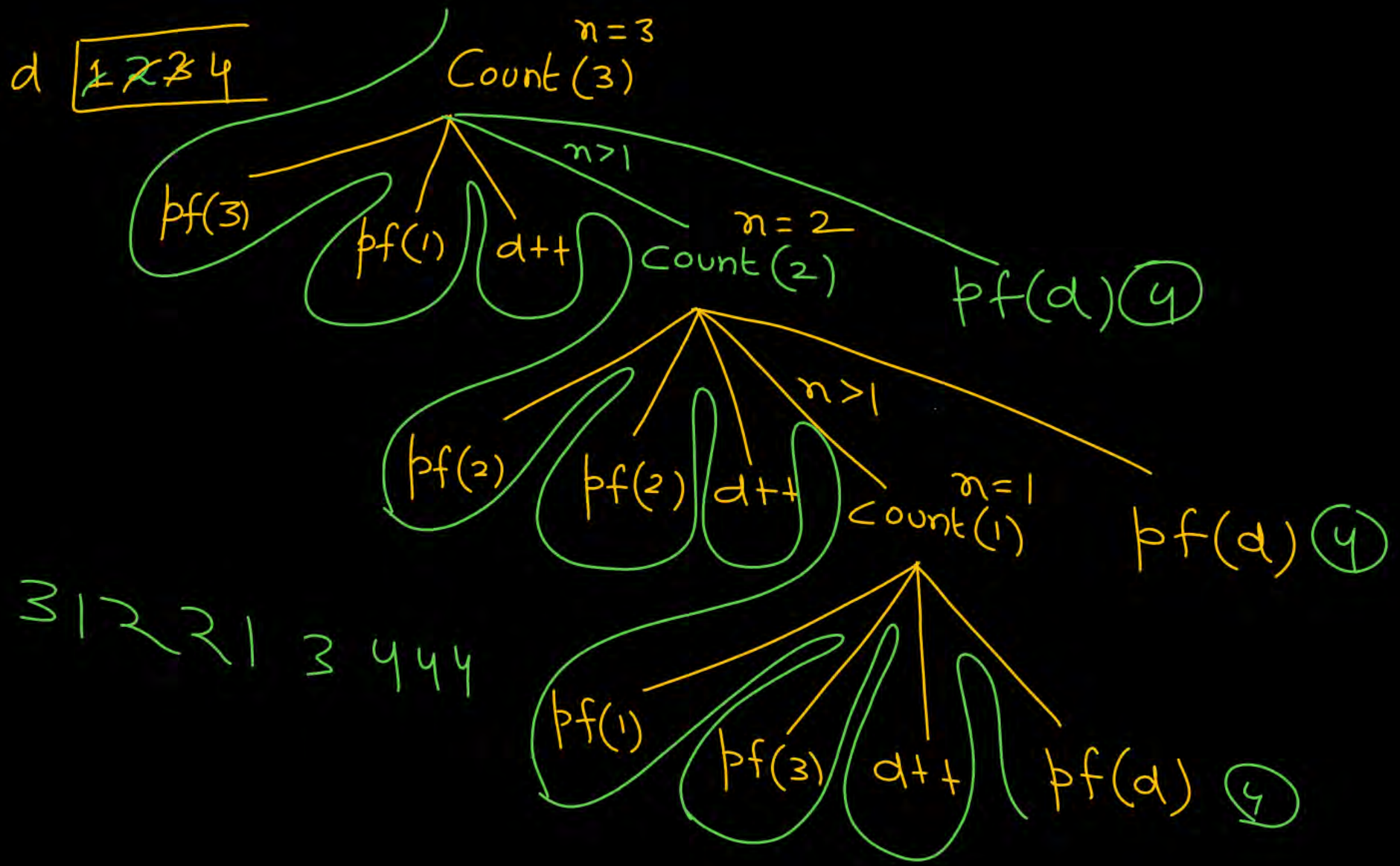
C. 3 1 2 2 1 3 4

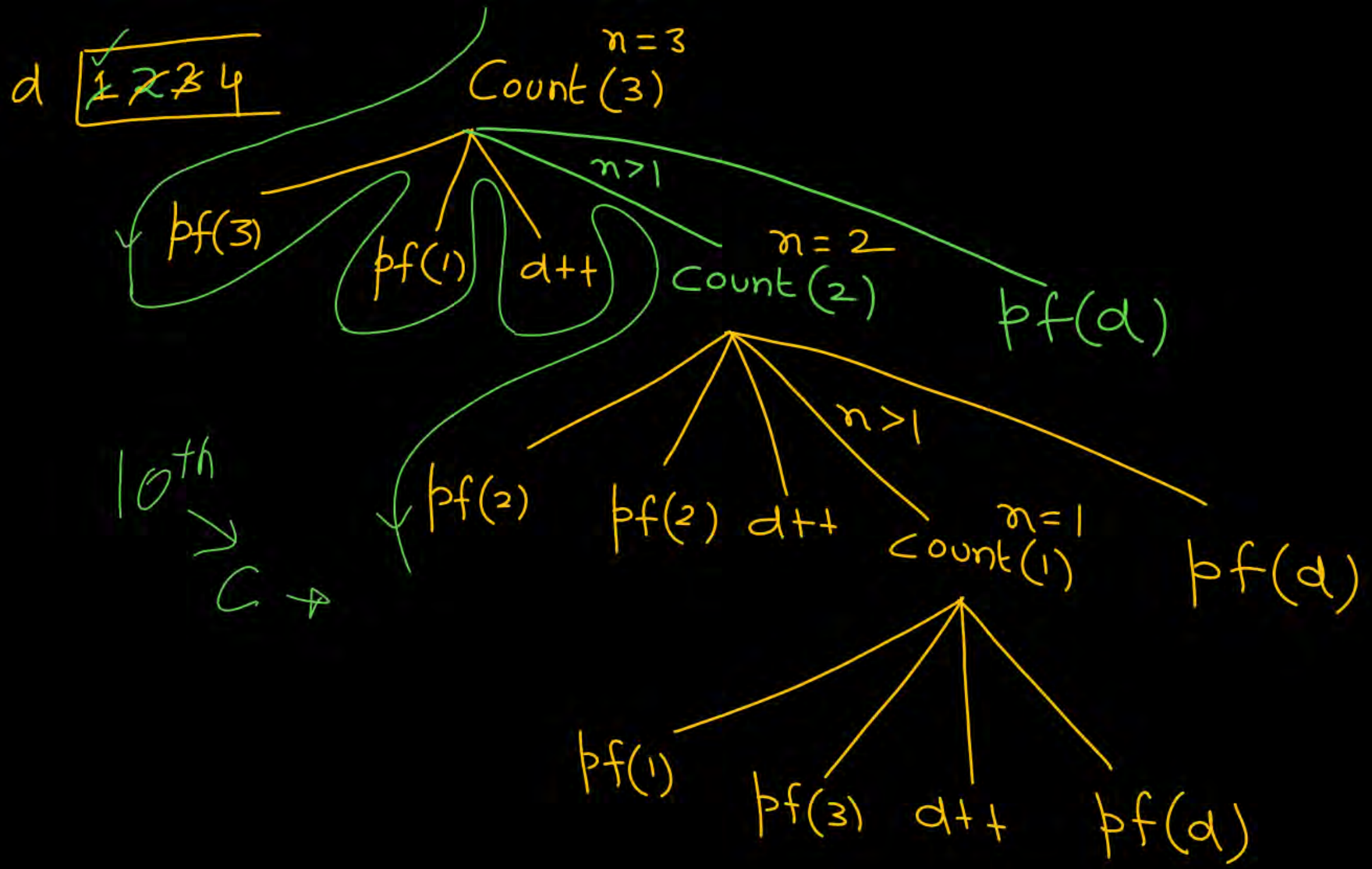
D. 3 1 2 1 1 1 2

```
fun(int n){
    if(n>1) {
        pf("Hello")(1)
    }
    (2) pf("1");
    (3) pf("0");
}
```

fun(2)


d 1 2
n = 3
Count(3)
n > 1
pf(3)
pf(1)
d++
count(2)
pf(d)
Why not
pf(2)
instead of
pf(d) ?
will execute
after count(2)
count(2)
⇒ changed


d 1234
n=3
Count(3)
pf(3)
pf(1)
d++
n>1
n=2
Count(2)
pf(d) 4
pf(2)
pf(2)
d++
n>1
n=1
Count(1)
pf(d) 4
3 1 2 2 1 3 4 4 4
pf(1)
pf(3)
d++
pf(d) 4


d
1 2 3 4
n = 3
Count(3)
pf(3)
pf(1)
d++
n > 1
n = 2
Count(2)
pf(d)
pf(2)
pf(2)
d++
n > 1
n = 1
Count(1)
pf(d)
pf(1)
pf(3)
d++
pf(d)
10th
C

Q.28

What will be the output of the C program?

```
#include<stdio.h>
int main()
{
  function();
  return  0;
}
void function()
{
  printf("Function in C is awesome");
}
```

A. Function in C is awesome

B. no output

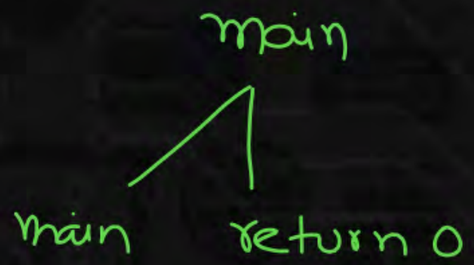
C. Runtime error

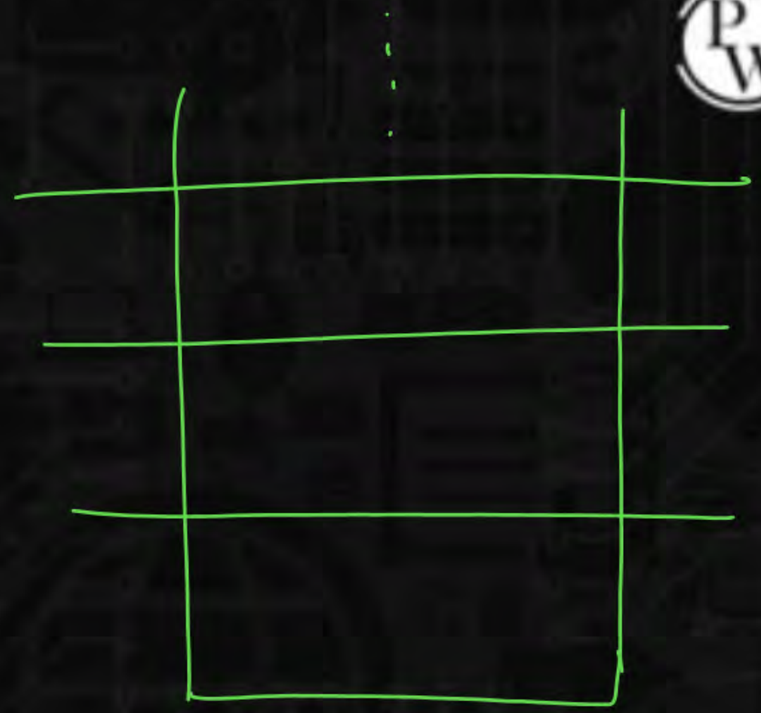
D. Compilation error

Q.29

What will be the output of the C program?

```
#include<stdio.h>
int main()
{
   main();
   return 0;
}
```





A. Runtime error

B. Compilation error

C. 0

D. None of these


THANK YOU GW
SOLDIERS !