CS & IT ENGINEERING

Programming in C

Functions and Storage Classes

Lec-04

By- Pankaj Sharma sir

Recursion

Q.1

```
void f(int n)
  {
     if(n<=0)
     return;
     printf("%d",n);
     f(n-1);
  }
```

What is the output of f(5)

$f(5) \rightarrow f(4) \rightarrow f(3) \rightarrow f(2) \rightarrow f(1) \rightarrow f(0)$

5 4 3 2 1

Arguments/value for which the function does not call itself

$\Rightarrow$ base value/base arg.

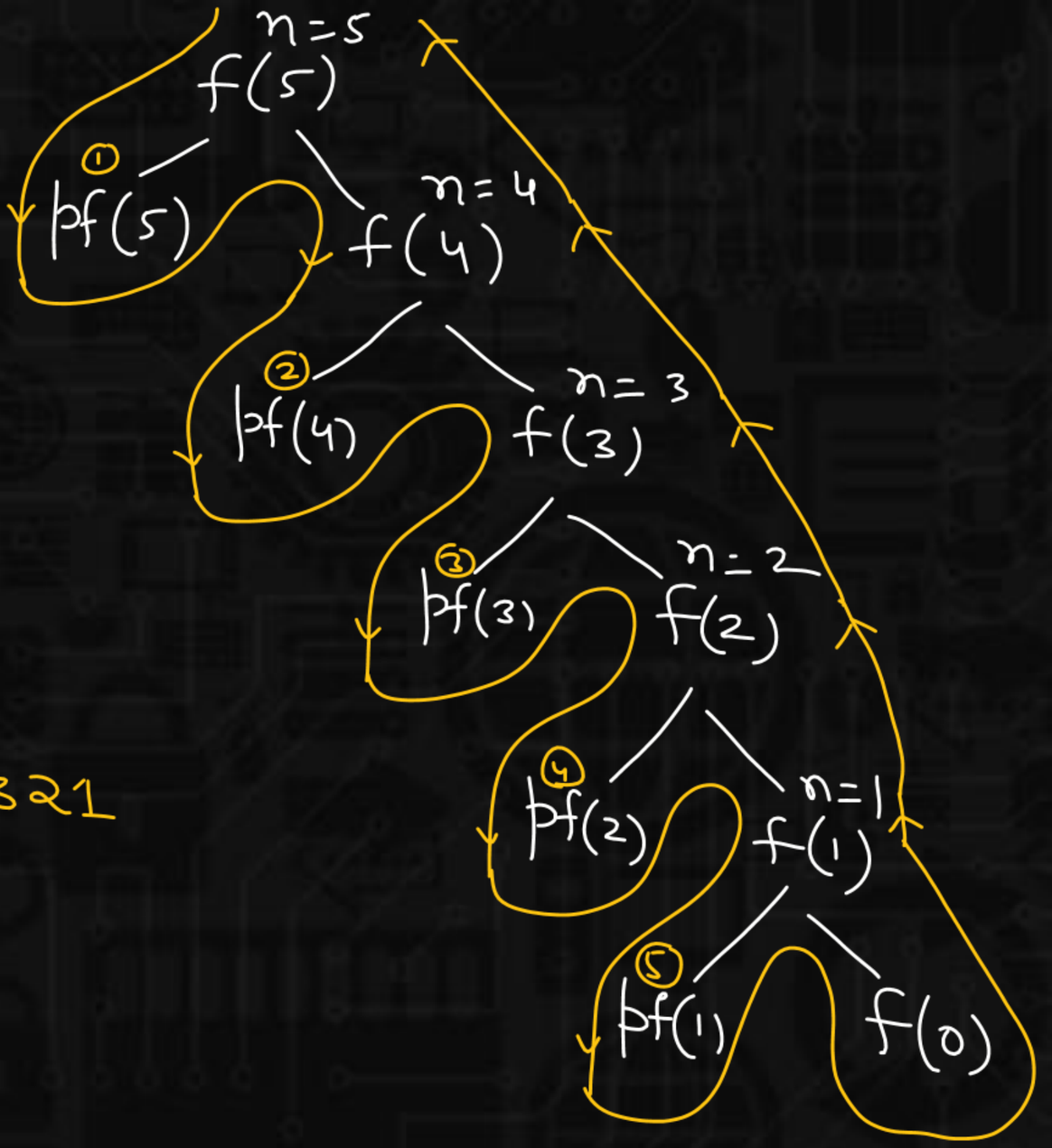
Q.1

```
void f(int n)
{
    if(n<=0)
    return;
1.  printf("%d",n);
2.  f(n-1);
}
```

base case

What is the output of f(5)

54321



Q.2

```
void f(int n)
   {
       if(n<=0)
       return;
       f(n-1);
       printf("%d",n);
   }
```

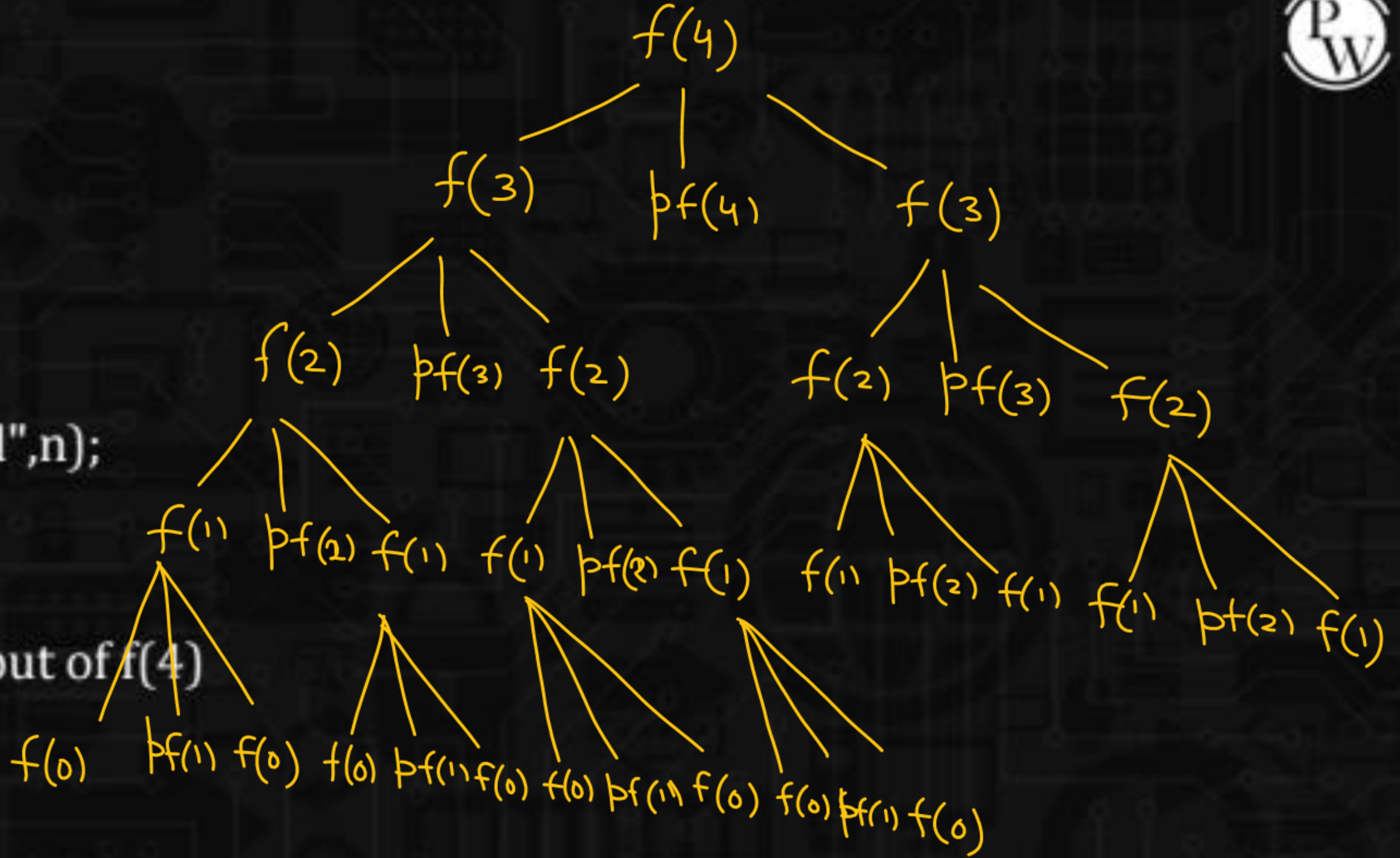
What is the output of f(5)

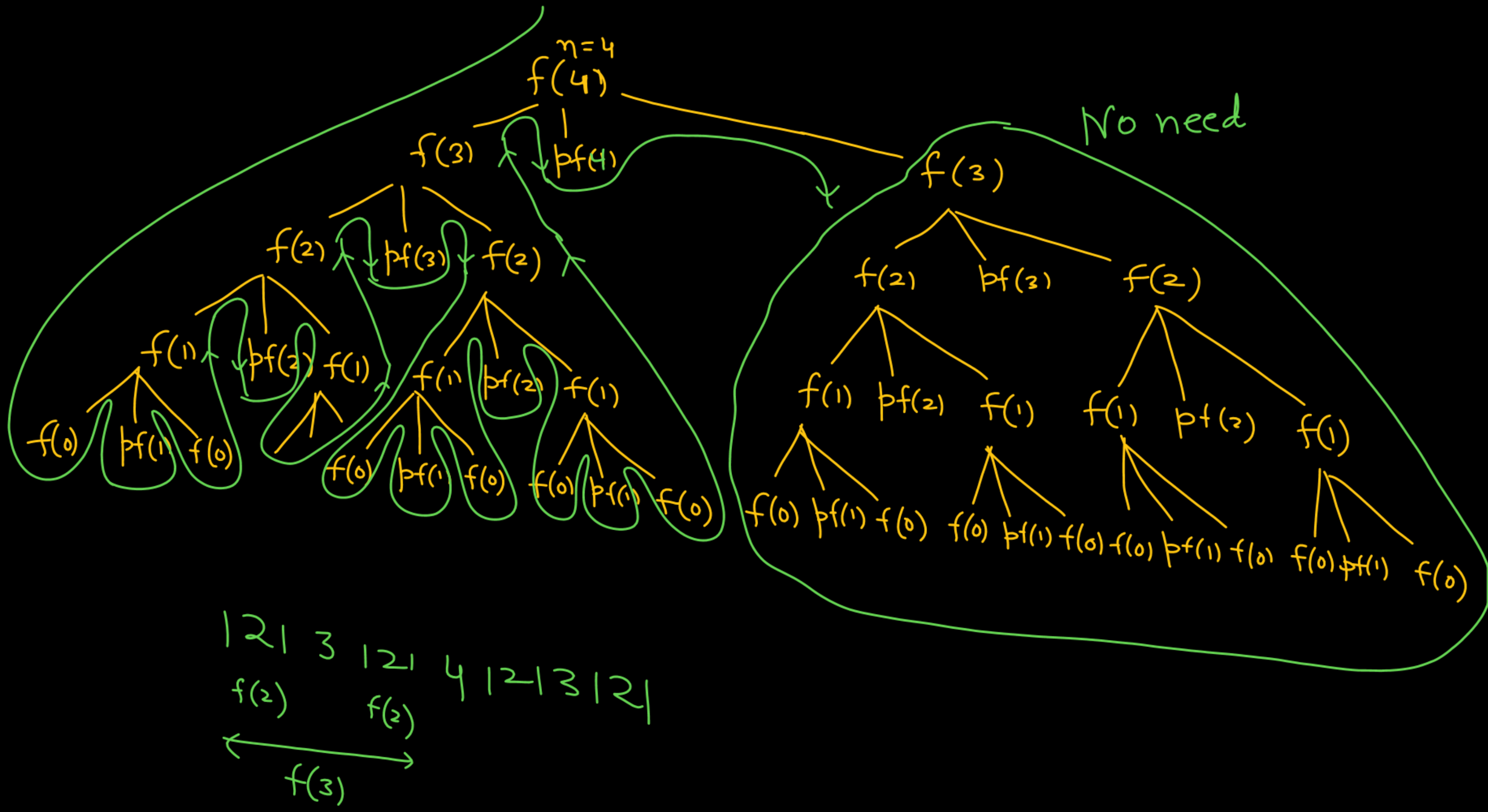
12345

Q.3

```
void f(int n)
{
    if(n<=0)
    return;
1.  f(n-1);
2.  printf("%d",n);
3.  f(n-1);
}
```

What is the output of f(4)





1 2 1 3 1 2 1 4 1 2 1 3 1 2 1

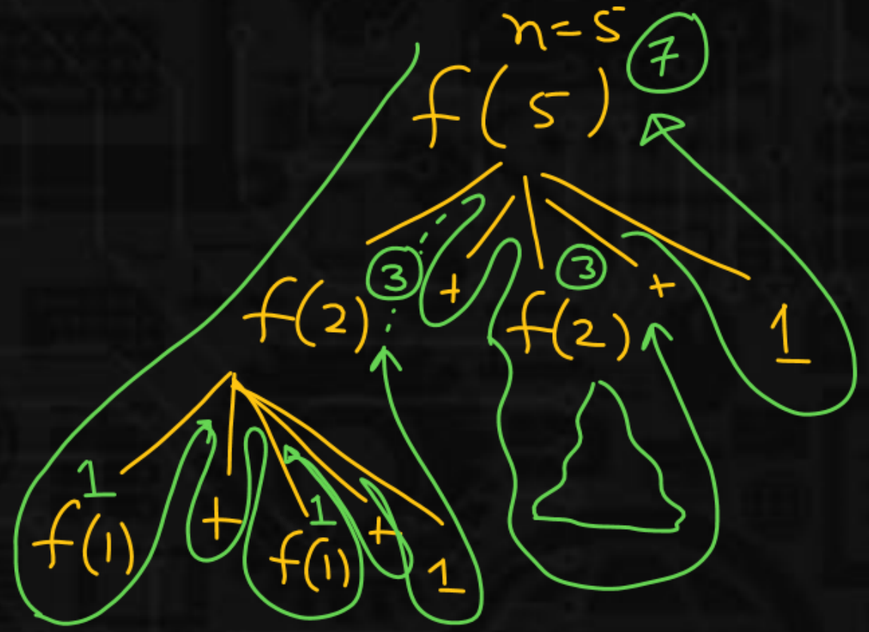
f(2) f(2)

f(3)

Q.4

```
int f(int n)
    {
        if(n<=1)
        return n;
        return f(n/2) + f(n/2) + 1;
    }
```

What is the output of f(5)

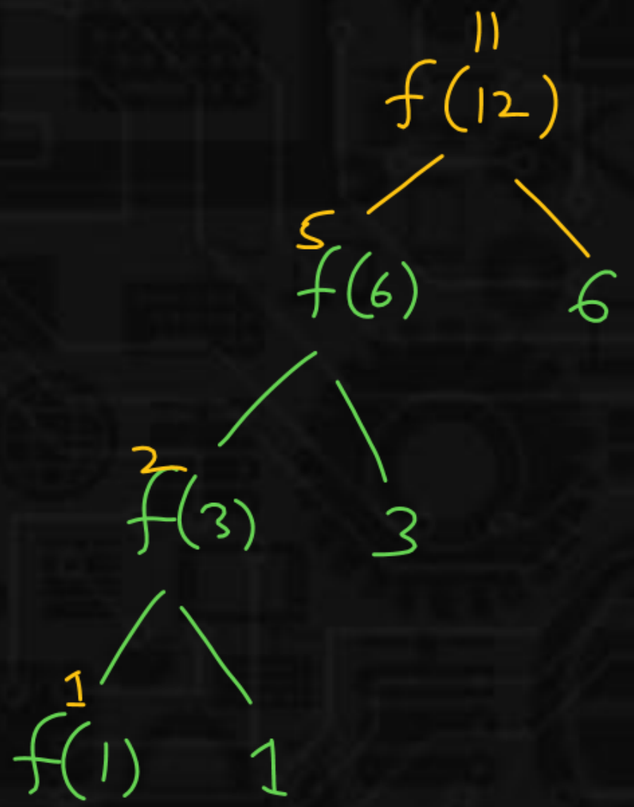


Recursion
- DAC
- Backtracking
- Dynamic Programming

Q.5

```
int f(int n)
    {
        if(n<=1)
        return n;
        return f(n/2) + n/2 ;
    }
```

What is the output of f(12)



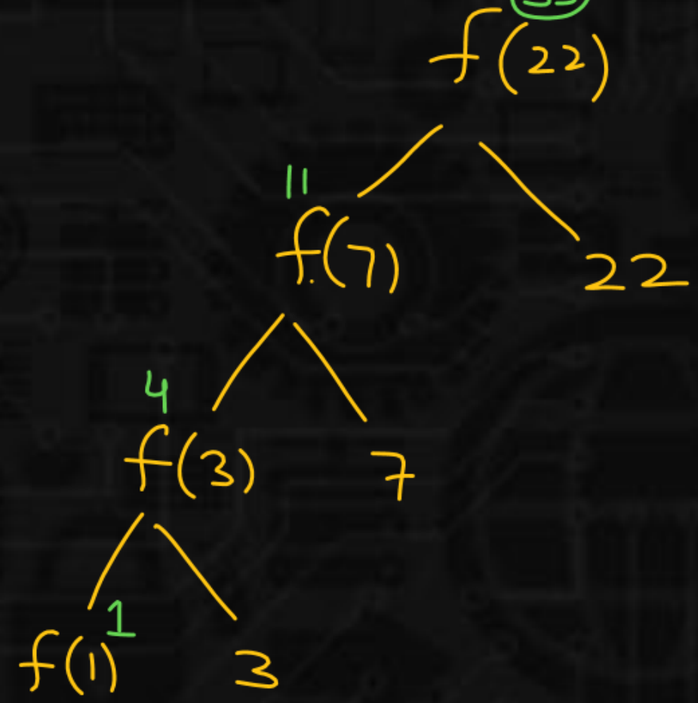
Q.6

```
int f(int n)
  {
    if(n<=1)
    return n;
    if(n%2)
    return f(n/2) + n;
    return f(n/3) + n;
  }
output of f(22) ?
```

odd

Even

if n is even ⇒ n%2 == 0 ⇒ false



n=11

```
if (n%2)
    printf("1");
```

11%2

⇒ 1

if (1) — True

pf("1")

n=10

10%2 ⇒ 0

if (0)

pf("1") ✗

Q.7

Consider the code :

```
/* Assume that n>=0 */
void fun(int n)
   {
      if(n==0)
      return ;
      fun(n/2);
      printf("%d",n%2);
   }
```

output of f(11)?

2	11	Rem
2	5	1
2	2	1
2	1	0
	0	1

Reverse order of Printing

Q.7

Consider the code :

```
/* Assume that n>=0 */
void fun(int n)
   {
      if(n==0)
      return ;
      fun(n/2);
      printf("%d",n%2);
   }
```

output of f(11)?

Q.8

Consider the following C program :

```
void foo(int n , int sum) {
    int k=0,j=0;
    if(n==0)
    return;
    k=n%10;
    j=n/10;
    sum=sum + k;
    foo(j,sum);
    printf("%d",k);
}
void main(){
int a=2048,sum=0;
foo(a,sum);
printf("%d",sum);
}
```

// call by value

⇒ 0

Output?

A. 8, 4, 0, 2, 14

B. 8,4,0,2,0

C. 2,0,4,8,14

D. 2,0,4,8,0

Q.8

Consider the following C program :

```
void foo(int n , int sum) {
    int k=0,j=0;
    if(n==0)
    return;
    k=n%10;
    j=n/10;
    sum=sum + k;
    foo(j,sum);
    printf("%d",k);
}
void main(){
int a=2048,sum=0;
foo(a,sum);
printf("%d",sum);
}
```

2048 0

k = 8

j = 204

8

// call by value

⇒ 0

Output?

A. 8, 4, 0, 2, 14

B. 8,4,0,2,0

C. 2,0,4,8,14

D. 2,0,4,8,0

foo(2048,0)

k = 8
j = 204
sum = 8
bf → wait

foo(204,8)

Q.9

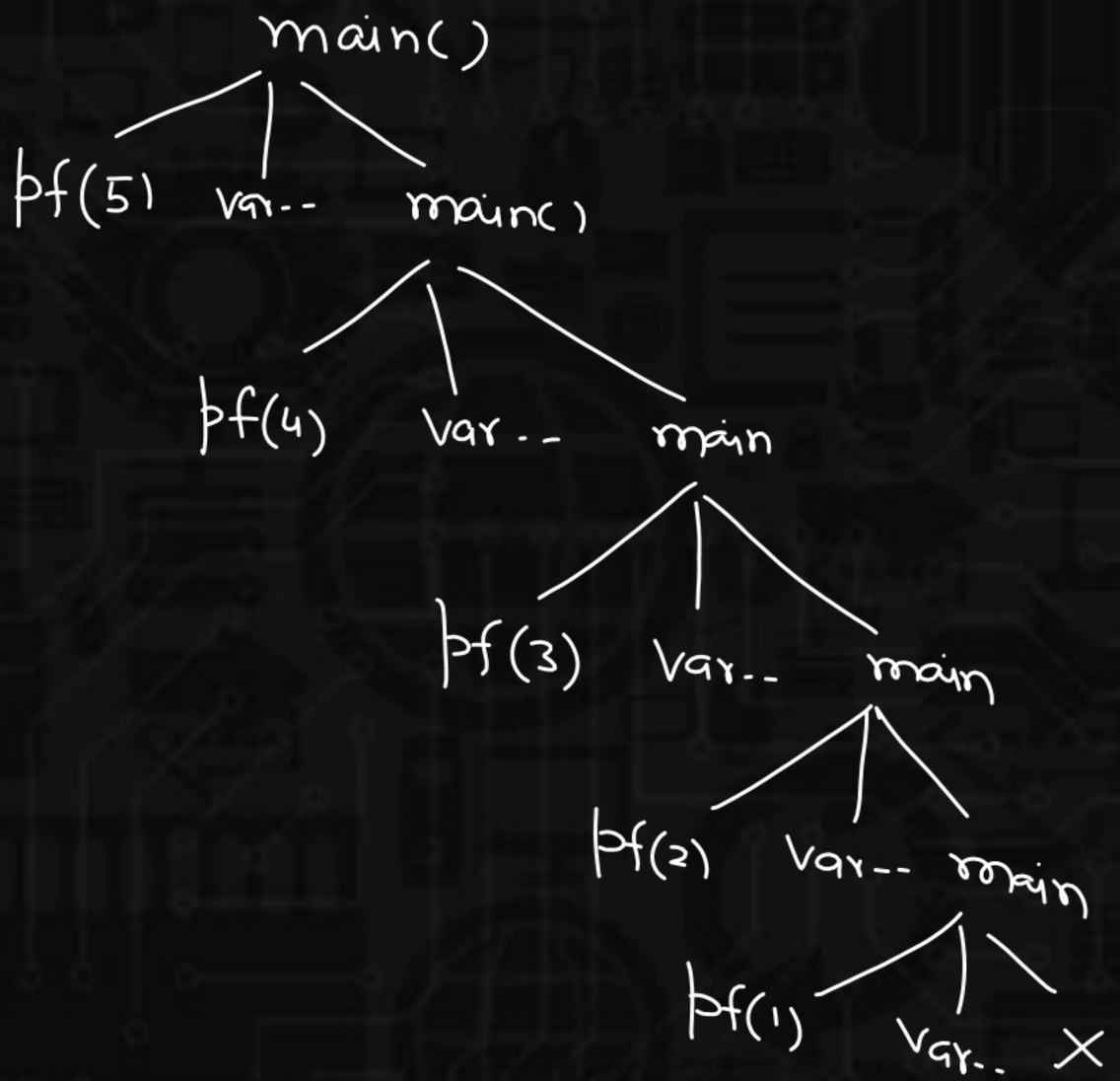
```
void main()
{
    static int var=5;
    printf("%d",var--);
    if(var)
    main();
}
```

→ data segment

Q.9

```
void main()
{
   static int var=5;
1  printf("%d",var--);
2. if(var)
3. main();
}
```

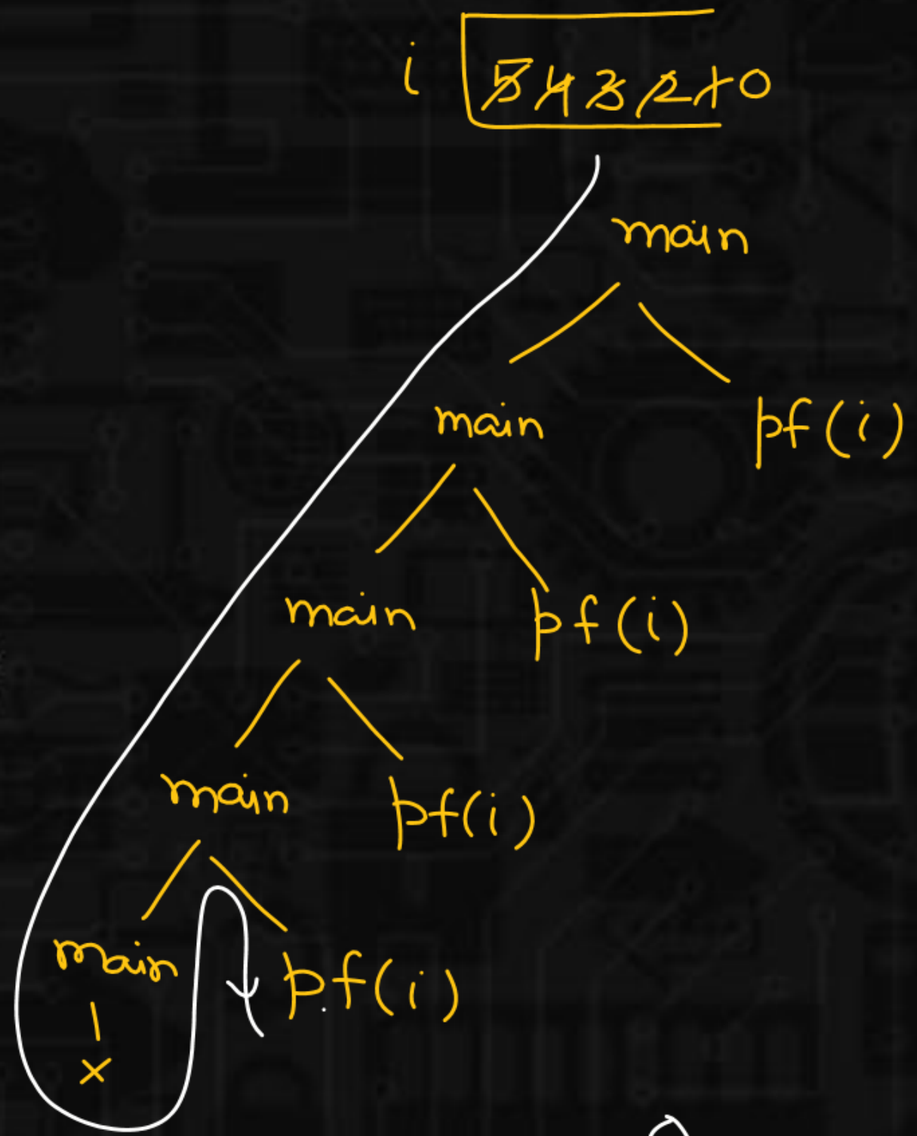
var 5 4 3 2 1 0



5 4 3 2 1

Q.10

```
void main()
{
  static int i =5;
   if(--i)
     {
       main();
       printf("%d",i);
     }
}
```

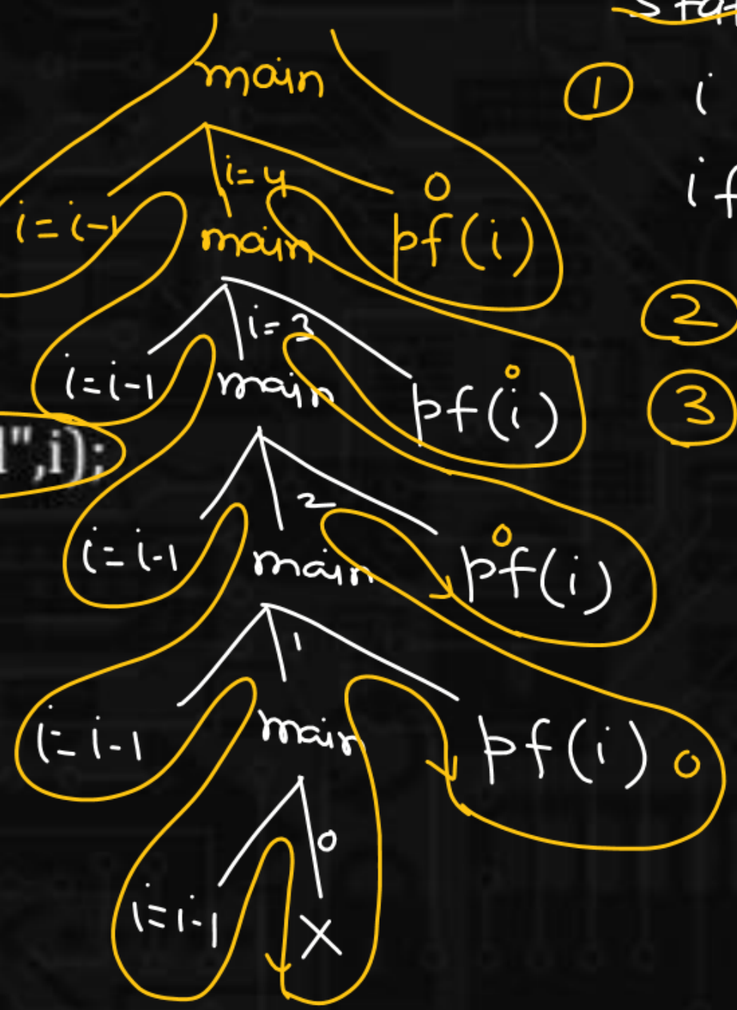


Q.10

```
void main()
{
   static int i =5;
   if(--i)
     {
   main();
   printf("%d",i);
     }
}
```

Change

i 5 4 3 2 1 0

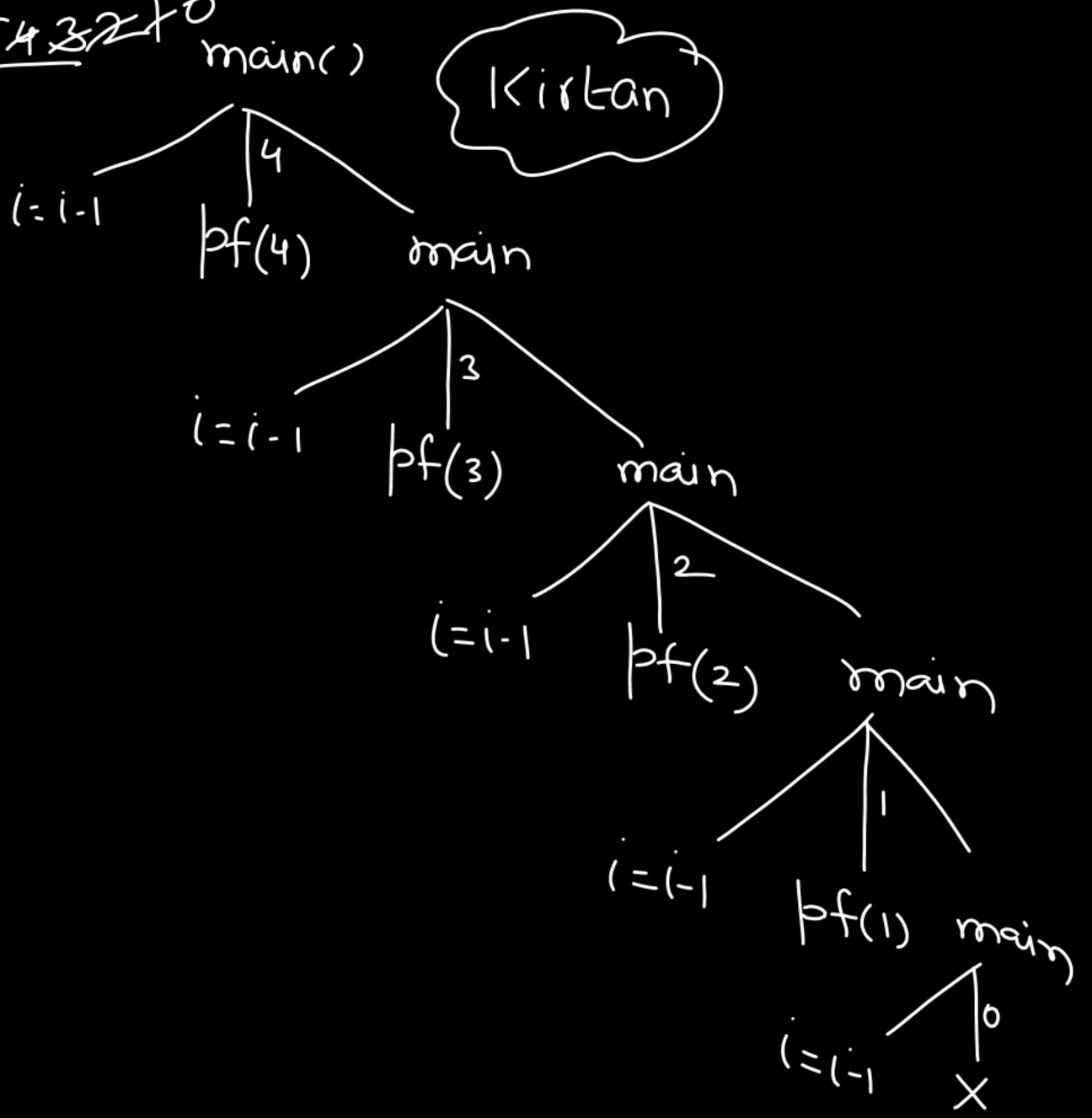



```
void main(){
  Static int i=5;
① i = i-1;
  if(i){
② main();
③ pf(i);
  }
}
```

0 0 0 0

```
void main(){
    static int i = 5;
    i = i-1;
  if(i){
  printf("%d", i);
  main();
      }
  }
```

i 5 4 3 2 1 0



Q.11

H.W

predict the output

```
int fun(int x)
   {
      if(x%2==0)
      return fun(fun(x-1));
   else
      return(x++);
   }
int main()
   {
      printf("%d",f(12));
      getchar();
      return 0;
   }
```

A. 10

B. 11

C. 12

D. None of these

Q.12

```
int fun(int a,int b)
    {
      if(b==0)
      return 0;
      if(b%2==0)
      return fun(a+a,b/2);
      return fun(a+a,b/2) + a;
    }

int main()
    {
      printf("%d",fun(4,3));
      return 0;
    }
```

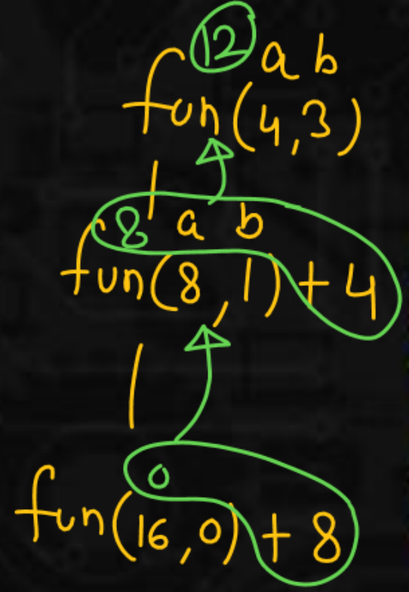
b is odd

A. 12

B. 81

C. 64

D. 8



$a \times b = 2a \times \frac{b}{2}$

$= 4a \times \frac{b}{4}$

Q.13

Consider the following C function :

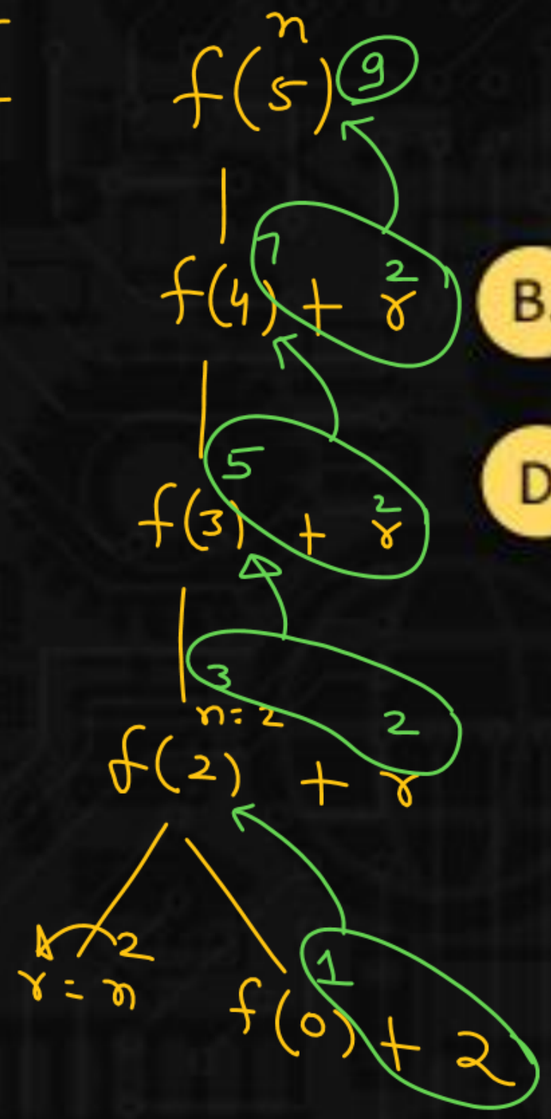
```
int f(int n)
 {
   static int r=0;
   if(n<=0)
   return 1;
   if(n<3)
   {
      r=n;
      return f(n-2) + 2;
   }
   return f(n-1) + r;
 }
```

what is the value of f(5)

r = 2

f(0), f(-1), f(-2)

f(1), f(2)



A. 5

B. 7

C. 9

D. 18

Q.14

Consider the following recursive C function

```
unsigned int foo(unsigned int n, unsigned int r)
 {
    if(n>0)
    return (n%r) + foo(n/r , r);
else
    return 0;
 }
```

output of foo(513,2)

513 = 512 + 1
2^9 2^0

2

A. 9

B. 8

C. 5

D. 2

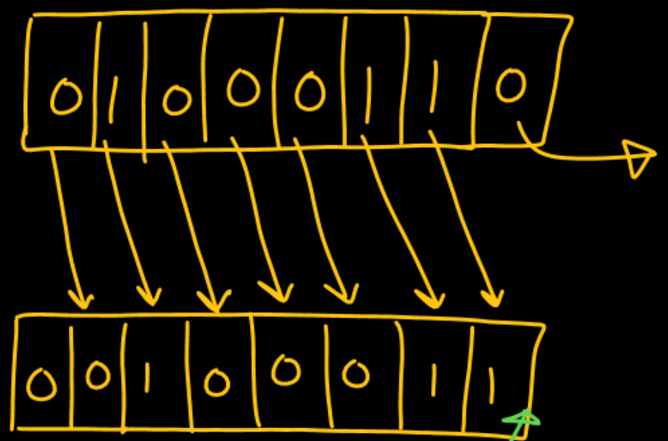
64+4+2

70 ⇒ 01000110

(i) 70 % 2 ⇒ 0

(ii) n = 70/2 = 35

35 % 2 = 1



$n = 1237$

(i) $n \% 10 \equiv$ last digit $= 7$

$n = n/10 = 123$

(ii) $n \% 10 =$ last_digit $= 3$

$n = n/10 \Rightarrow 12$

1010

① $n \% 2 = 0$

$n = n/2 = 0101$

② $n \% 2 = 1$

Q.15 Which of the following statements is/are valid?

A. return a+b;

B. return a,b,c;

C. return (a,b,c);

D. All of them

Comma operator

Q.16

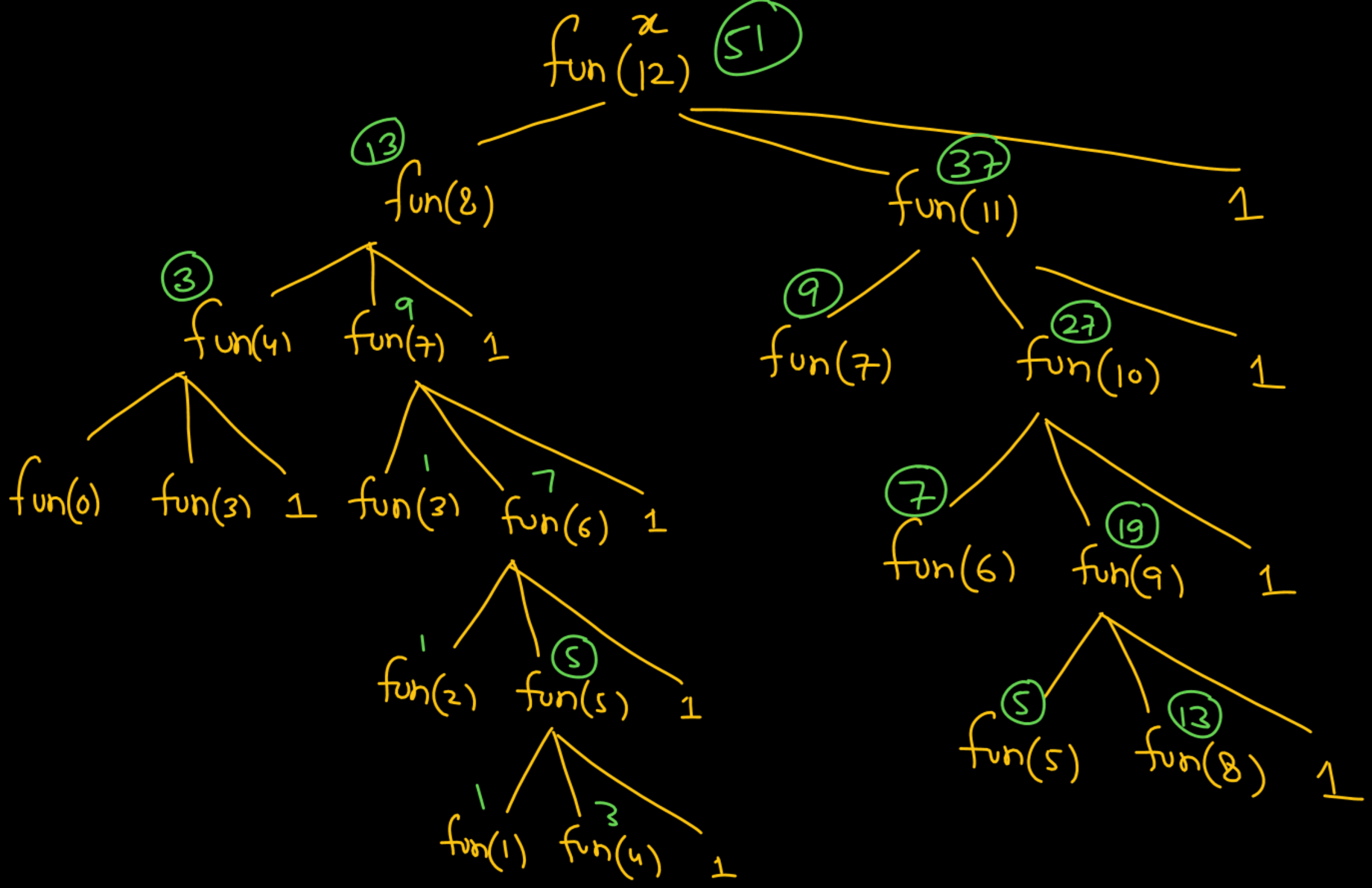
```
int fun(int x)
{
  if(x>3)
    return fun(x-4) + fun(x-1) + 1;
  return 1;
}
```

Find the value returned by fun(12)

fun(12) → fun(4), fun(7), fun(6) ...

fun(3) = fun(2) = fun(1) = fun(0) = 1

51


x
fun(12)
51
13
fun(8)
37
fun(11)
1
3
fun(4)
9
fun(7)
1
9
fun(7)
27
fun(10)
1
fun(0)
fun(3)
1
1
fun(3)
7
fun(6)
1
7
fun(6)
19
fun(9)
1
1
fun(2)
5
fun(5)
1
5
fun(5)
13
fun(8)
1
1
fun(1)
3
fun(4)
1

Q.17

Predict output of following program

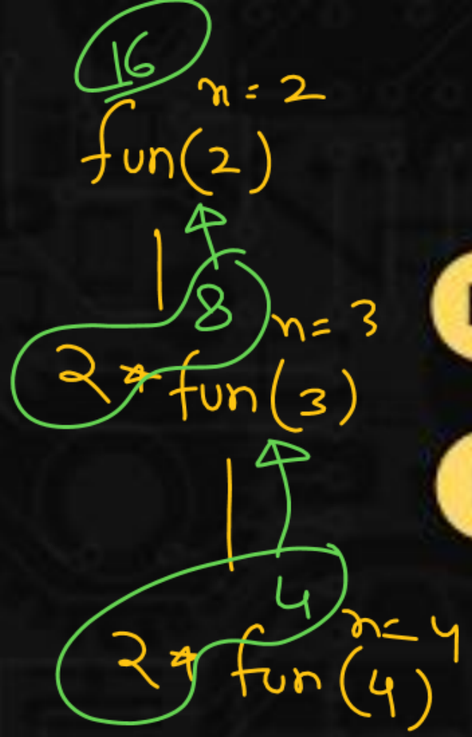
```
#include <stdio.h>
int fun(int n)
{
   if (n == 4)
   return n;
   else return 2*fun(n+1);
 }
int main()
 {
   printf("%d ", fun(2));
   return 0;
 }
```

A. 4

B. 8

C. 16

D. error



8:30 PM

Q.18

Consider the following recursive function fun(x, y). What is the value of fun(4, 3)

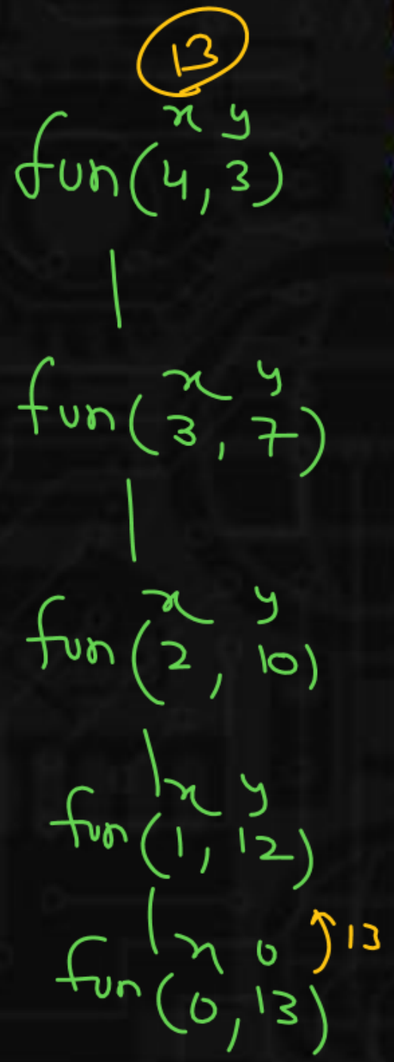
```
int fun(int x, int y)
{
   if (x == 0)
   return y;
   return fun(x - 1,  x + y);
}
```

A. 13

B. 12

C. 9

D. 10




THANK YOU GW
SOLDIERS !