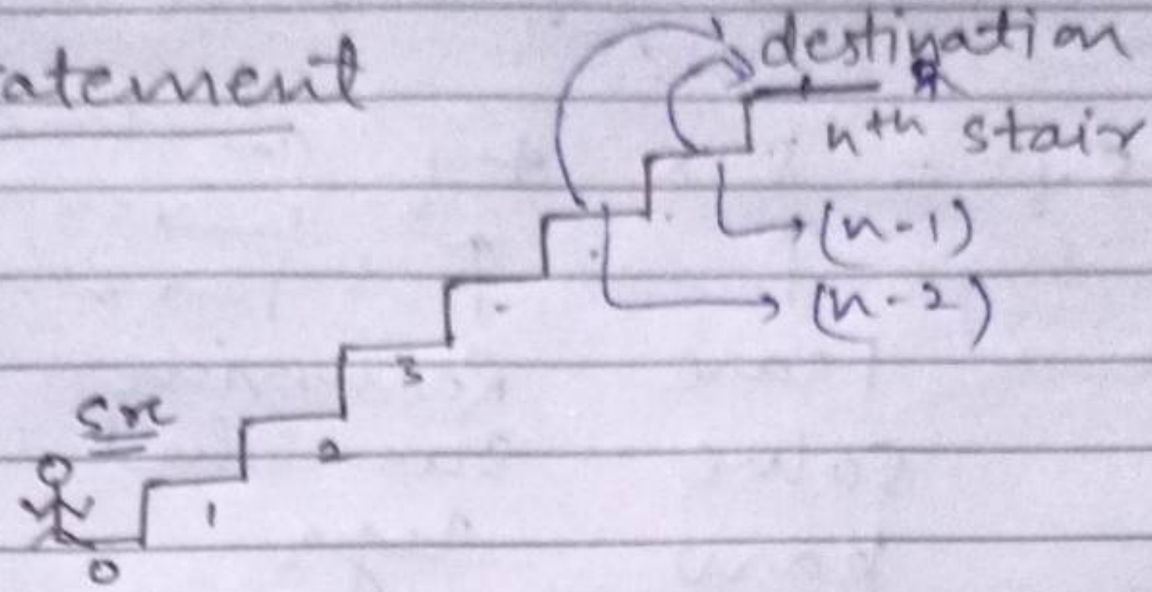
Recursion Lecture 2

Climb Stairs

Problem Statement



steps allowed ⇒
- 1 stair at a time
- 2 stairs at a time

Find no. of ways to reach nth stair

One can reach nth stair only by 2 ways.

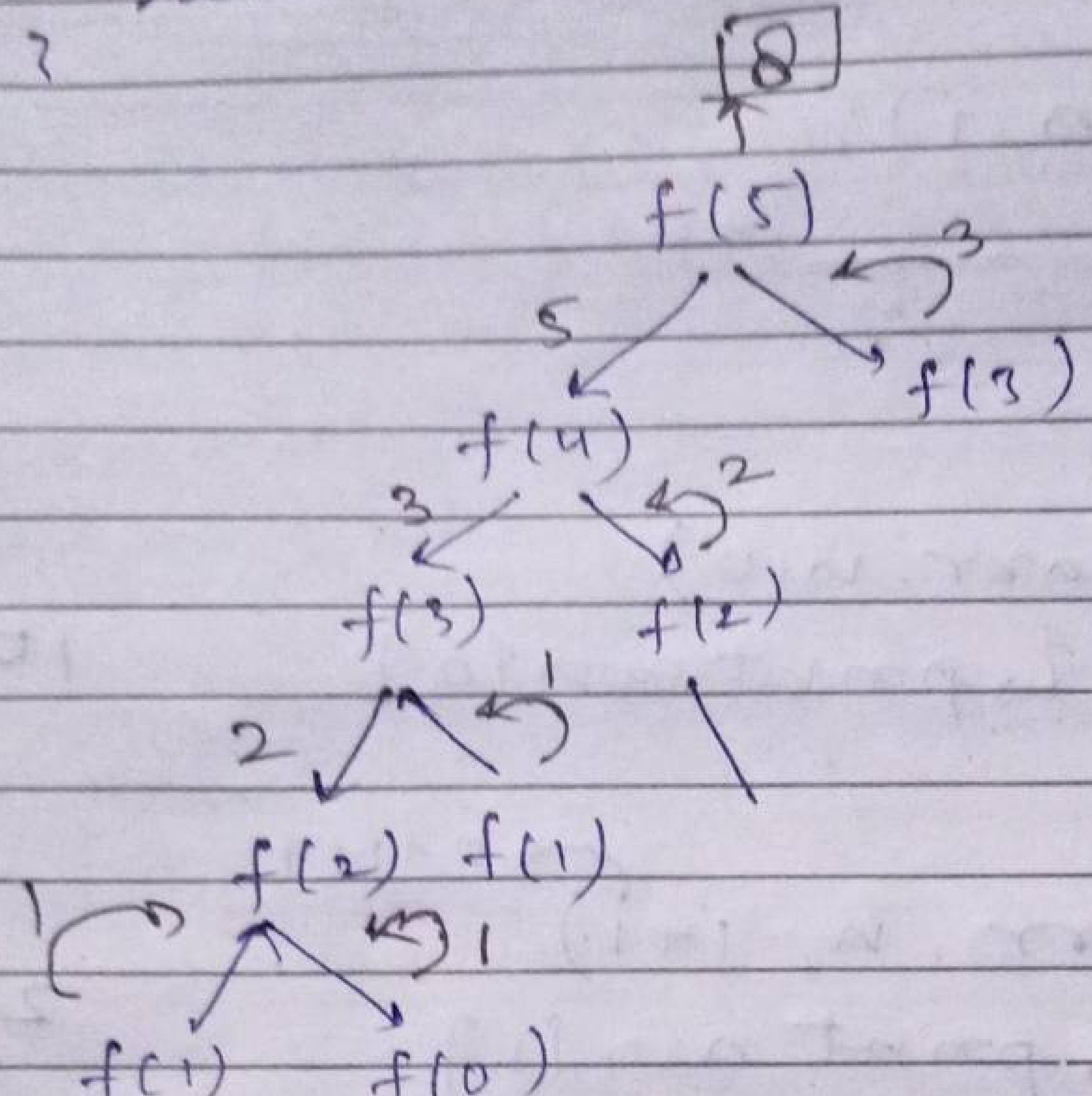
Total no. of ways to reach nth stairs

$$f(n) = f(n-1) + f(n-2)$$

Code

```
int main(){
    int n;
    cin>>n
    int ans = climbStairs(n);
    cout << ans << endl;
}
int climbStairs(int n){
    if(n==0 || n==1)
        return 1;
```

```
int ans = climbStairs(n-1) + climbStairs(n-2);
  return ans;
}
```



Ques.

10	20	30	40	50

Print all elements

With loop

```
for (int i=0; i<n; i++)
{
    cout << arr[i];
}
```

With Recursion

```
void print (int arr[], int n, int i){
    if (i<n)
    if (i>=n){
        return;
    }
    cout<< arr[i]<< " ";        // 1 case solve
    print(arr, n, i+1);
}
```

```
int main() {
  int arr[5] = {10, 20, 30, 40, 50};
   int n=5;
   int i=0;
  print(arr, n, i);
}
```

Working

arr
n=5
i=0

f(arr, n, 0)
  ↳ print arr[0] 10

↓

f(arr, n, i+1) ↷ 0+1=1
  ↳ print arr[1] 20

↓

f(ars, n, i+1) ↷ 1+1=2
  ↳ print arr[2] 30

↓

f(ars, n, i+1) ↷ 2+1=3
  ↳ print arr[3] 40

↓

f(ars, n, i+1) ↷ 3+1=4
  ↳ print arr[4] 50

↳ f(ars, n, i+1) ↷ 4+1=5

fails

Maximum no. in array using Recursion

```
int main(){
    int arr[] = {10,20,30,21,44,32,17,19};
    int n=8;
    int maxi = INT_MIN;
    int i=0;
    findmax(arr,n,i,maxi);
    cout << "Max no. is = " << maxi << endl;
return 0;
}

void findmax(int arr[], int n, int i, int &maxi){
    if(i>=n){ // array khtm, traverse hogya pura.
        return;
    }
    if(arr[i]>maxi){ // Ek care solve karo, current element check karo.
        maxi = arr[i];
    }

    findmax(arr,n,i+1,maxi)

}
```

pass by reference

Working :-

10	30	41	20
0	1	2	3

f(arr, 4, 0, INTMIN)

↓ maxi updated (10>0)

f(arr, 4, 1, 10)

↓ maxi updated (30>10)

f(arr, 4, 2, 30)

↓ maxi updated (41>30)

f(arr, 4, 3, 41)

↓ maxi not update (41>20)

f(arr, 4, 4, 41)

→ Base case so.

So maximum no there will be printed garbage value that is of INTMIN as it is called by value

So we have to call it by reference.

Thus output will be 41.

10	44	30
0	1	2

(arr, 3, 0, INT_MIN)

```
void findmax(arr, n, i, maxi)
{
    if (i>=n) return;   → F
    maxi = max(maxi, arr[i])
    findMax(arr, n, i+1, maxi)
}
```

10 → maxi

(arr, 3, 1, 10)

```
void findmax(arr, n, i, maxi)
{
                        F
}
```

44 ← maxi

arr, 3, 2, 44

```
void findmax(arr, n, i, maxi)
{
                        F
}
```

44 ← maxi

arr, 3, 3, 44

```
void findmax(arr, n, i, maxi)
{
    if (i>=n) return;   True
}
```

→ cout << maxi; → in main function

so here garbage value of INT_MIN will be printed.

for maxi

```
main ()
{
    [INT_MIN] x → [10] x → [44] → [44]
}
```

same box me bar bar value update hogi main function mein

This is call by reference

Minimum no. in array using Recursion

```
void findmin( int arr[], int n, int i, int &mini ){
    if (i>=n){
        return;
    }
    mini = min(mini, arr[i]);
    findMin(arr, n, i+1, mini);
}
```

Q. I/p → string str = "love babbar"
Key → 'r'
↳ check r is present in str or not using recursion.

l	o	v	e	b	a	b	b	a	r

first wale ko check karo baki recursion sambhal lega.

```
int main(){
    string str = "lovebabbar";
    int n = str.length();
    char key = 'r', int i=0;
    bool ans = checkkey(str,i,n, key);
cout<<" ans is "<< ans<<endl.
    return 0;
}
```

```
bool checkkey (string str, int i, int n,
                char key) {
    if (i >= n) {
        return false;
    }

    if (str[i] == key)
        return true;

    bool ans = checkkey (str, i+1, n, key);
    return ans;
}
```

NOTE → Kisi variable ya data structure ko function mein pass krna ho val store karane keliye to by reference pass karao.

Q. I/p → 647
o/p → print all digit of this no.

Using loop ⇒

647 %.10 → 7
/10 ↓
64 10%. → 4
/10 ↓
6 10%. → 6
/10 ↓
0 → Rukna hai (Bare case)

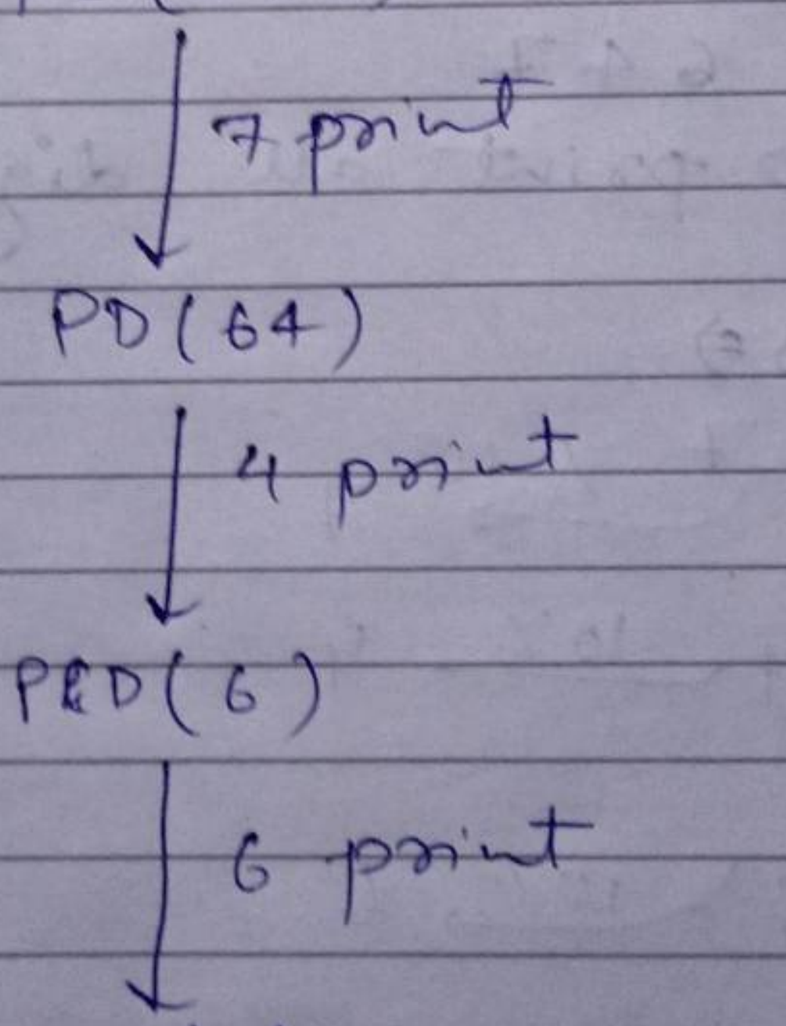
PAGE No
DATE

Code

```
int main() {
    int n = 647;
    printdigits(n);
    return 0;
}

void printdigits(int n) {
    if (n == 0) {
        return;
    }
    printdigits(newval)
    int digit = n % 10;
    cout << digit << " ";
    int newval = n/10;
}
```

Working



Correct order → Recursion call karo pehle fir print karo.

PD(647)
- digit = 647 % 10
- cout << digit
- 647 ⇒ $\frac{647}{10}$ = 64

↓

PD(64)
- 64%10 → 4
- 64/10 → 6

↓

PD(6)
- 6 % 10 → 6
- 6/10 → 0

↓

PD(0)

↳ Base case