Theme:

Date: / /

Sat Sun Mon Tue wed Thu Fri

code:

```
while (a>0 && b>0)
{  if (a>b) a = a%b;
   else  b = b%a;
}
if (a==0) return b;
 else return a;
```

Recursion

when a function calls itself until a specific condition is met.

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

f(), L2 | waits
f(), L2 | "
f(), L2 | "

Infinite recursion
out of memory → stack overflow

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

0
1
2
3
4 =

* Base condition →

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

Goes till 4 and stops

1
2
3

Date: / /

Sat Sun Mon Tue wed Thu Fri

Theme:

Recursion Tree: f() → f() → f() → f()

Basic Recursion Problems:

Q. Print name N times.

```
void (i, n) {
  if (i > n) return; → Base case
  Print ("Ray");
  f(i+1, n);
}
main() { int n;
         cin>>n;
         f(01, n);
       }
```

Ray
Ray
Ray

Q. Print linearly from 1 to N

```
f(i, n) { if (i>n) return;
          print(i);
          f(i+1, n);
        }
main() { Input (n);
         f (1, n)
       }
```

Date: / /

Sat Sun Mon Tue wed Thu Fri

Theme:

Q. Print integers N → 1

```
f(i,n) { if(i<1) return;
         print(i)
         f(i-1, N)
       }
main() { input(n);
         f(n,n);
       }
```

Q. Print for 1 → N (not with f(i+1,n) ✗ ⟹ Backtracking

```
f(i,n) { if(i<1) return;
         f(i-1), N);
         print(i);
       }
main() { input(n);
         f(n,n);
       }
```

Q. Print for N → 1 (not with i-1, n)

```
f(i,i) { if(n<i) return;
         printf(n, i+1);
             print(i);
       }
main() { input(n);
         f(n,1);
       }
```

Theme: Parameterised function

Date: / /
Sat Sun Mon Tue wed Thu Fri

Q. Summation of 1 -> N

```
f( i, sum) {
        if (i<1) { print (sum);
                   return ;
                 }
        f(i-1, sum+i) ;
    }
main() { input (n) ;
         f(n, 0) ; -> zero.
       }
```

functional

```
f(n) { if(n== 0) return 0;
       return n + f(n-1) ;
     }
main () { n ;
          printf(f(n)) ;
        }
```

Q. Factorial of N

```
f(n) { if(n==0) return 1 ;
       return n * f(n-1);
     }
main () { n ;
          prin(f(n)) ;
```

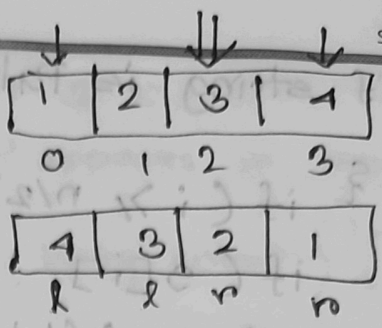
TC -> O(N)
SC -> O(N)

Theme:

Date: / /

☐ Sat ☐ Sun ☐ Mon ☐ Tue ☐ wed ☐ Thu ☐ Fri

Q. Reverse an array

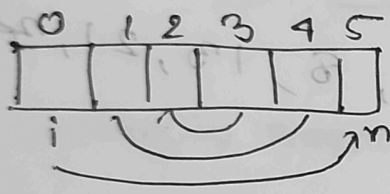


Ptr swap middle no swap.

moment they cross/ overlap → dont

two pointers:

```
f(l,r)
{ if(l >= r) return;
  swap(a[l], a[r]);
  f(l+1, r-1);
}
main(){ input(arr);
        f(0, n-1);
}
```



till now hong? $i \geq n/2$

```
f(i) { if (i >= n/2) return;
       swap(a[i], a[n-i-1]);
       f(i+1);
     }
main() {
       input(arr);
       f(arr, n);
```

Theme:

Date: / /

Sat Sun Mon Tue wed Thu Fri

Q. Check if string is Palindrome

rev
rar → rar

functional

bool f(i) {
    if (i >= n/2) return true;
    if (s[i] != s[n-i-1]) ret false;
    return f(1+i);
}

main() { string s; print f(f(0,s)); }

TC → $\frac{N}{2}$

SC → Auxiliary stack space $\frac{1}{2}$

---

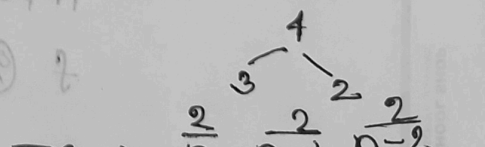
* Multiple Recursion Calls

Fibonacci : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34

Nth → f(n) → f(n-1) + f(n-2);

f(n) { if (n <= 1) return n;
    last = f(n-1)
    secondlast = f(n-2)
    return last + secondlast;
}

main() { n
    f(n);
}



TC → 2^n near about exponential nature.