- Arrays & Strings:
  - max/min, second largest, reverse, rotate, prefix sum, frequency count, pair sum $$a[i]+a[j]=K$$, removing duplicates.
  - String problems: anagram check, count words, remove spaces, substring search, palindrome, character frequency.

- Searching, Sorting, Basic DSA:
  - Linear / binary search, bubble / selection / insertion sort, simple use of built-in sort (Java Collections, Python `sort`).
  - Using lists/vectors/ArrayList for dynamic arrays; stack/queue using array or list (push, pop, bracket-balance).

- Functions & Recursion:
  - Recursive factorial, Fibonacci, power $$a^n$$, sum of array elements, gcd, string palindrome recursion.
  - Backtracking ప్రారంభ taste: permutations of string, generate all subsets for చిన్న constraints.

- Math / Logic Problems:
  - Number theory basics: gcd, lcm, prime sieve (up to N), count divisors, cyclic shift, bit-count (set bits).
  - Simple combinatorics / counting: ways to climb stairs (Fibonacci type), sum of series, modular arithmetic (answer % 10^9+7).

- Applied / "Story" type problems (CodeChef / HackerRank స్టైల్):
  - Given N students' marks, print rank list, cut-off count, pass/fail stats.
  - Given transaction list, compute final balance, max profit, simple leader-board, voting/majority element.

లాంగ్వేజ్-స్పెసిఫిక్ యాంగిల్స్

- C: pointers with arrays/strings, structures + array of structures (student records), dynamic memory `malloc` at least ఒక problem వరకు.

- Java: classes & objects, basic OOP (constructor, methods), using `ArrayList`, `HashMap` for frequency problems.

- Python: list/tuple/dict usage, list comprehensions simpleగా, string methods, using `set`/`dict` to solve duplicates, count-frequency problems తేలికగా.

డేటా సైన్స్ / AI / Robotics స్ట్రీమ్స్‌లో

వాళ్లకి base language questions same కానీ context data-oriented గానో, sensor/robot movement story గానో వచ్చే chance ఉంటుంది.

- CSV-like input నుండి basic stats (min, max, avg, count) తీసుకోవటం.
- 2D grid మీద robot moves: ఇవ్వబడిన commands (L,R,U,D) ప్రకారం final position కనుక్కోవటం, obstacle avoid చెయ్యటం.
- Simple simulation problems (queue of tasks, process scheduling-లాగా) using loops + arrays/queues.

difficulty & contest pattern

college multi-language contests:
2 easy implementation,
2 medium (arrays/strings),
2 little-logic/DSA problems

practice plan:
- Day 1: arrays + strings basic problems
- Day 2: search/sort + recursion చిన్న problems.
- Day 3: math/logic + 1–2 “story” questions (leaderboard/robot moves).

Easy level (15)

1. Arrayలో అన్ని elements sum & average.
2. Arrayలో max మరియు min కనుగొనాలి.
3. Arrayని reverse చేయాలి (in-place).
4. ఇచ్చిన element arrayలో ఉన్నదో లేదో (linear search).
5. String length, vowels, consonants count. [3]
6. String palindrome ఆ కాదో చెక్ చేయాలి.
7. రెండు strings equalనా కాదు (manual compare).
8. First non-repeated character in a string.

9. N వరకు prime numbers count చేయాలి (simple check).
10. Fibonacci first N terms (iterative).
11. Factorial of N (iterative, long range).
12. 2D matrix sum, ప్రతి row & column sum.
13. Marks ఇచ్చినప్పుడు pass/fail count (>=40).
14. Simple interest & compound interest calculator.
15. Decimal↔binary conversion (iterative). [4]

***

## Medium level (20)

16. Second largest element in array (single pass).
17. Arrayలో duplicates ఉన్నాయా? ఉంటే unique elements మాత్రమే print చేయాలి.
18. Frequency of each element in array (using map/dict). [1]
19. Arrayలో Kవ smallest మరియు Kవ largest element.
20. Two Sum: array & target K ఇచ్చినప్పుడు, a[i] + a[j] = K pairs అన్నీ print చేయాలి. [5]
21. Left rotate/right rotate array by K positions.
22. Merge two sorted arrays into one sorted array. [6]
23. Binary search (recursive version).
24. Bubble / insertion sort implementation; comparisons & swaps count కూడా చూపాలి.
25. Matrix addition & matrix multiplication (N×N).
26. Diagonal sums of square matrix; matrix symmetricనా కాదు చెక్ చేయాలి.
27. Stringలో ప్రతి wordను reverse చేసి sentence intactగా ఉంచాలి.
28. Remove all occurrences of a given character from string.
29. Count words, spaces, digits, special characters in string.
30. Check two strings are anagrams or not (using sort / frequency). [3]
31. Prefix sums: array prefix sum build చేసి, Range Sum Query [L,R] O(1)లో answer చేయాలి. [1]
32. Majority element (N/2 కంటే ఎక్కువ సార్లు వచ్చిన value) ఉందా? ఉంటే print చేయాలి.
33. Stock buy-sell one time: max profit కనుగొనాలి. [1]
34. Given N students (name, marks), structure/class list నుండి topper & average marks చూపాలి.

35. Simple leaderboard: players scores ఇచ్చిdescending orderలో rank & ties handle చేయాలి.

***

## Hard level (15)

36. Kadane’s algorithm – largest sum contiguous subarray. [1]
37. Move all zeros to end maintaining order of non-zeros (O(n), O(1) extra space). [5]
38. Longest subarray with sum = K (positive & negative numbers; hashmap technique). [1]
39. Longest increasing subsequence (LIS) – $O(n^2)$ version studentsకి సరిపోతుంది. [7]
40. Trapping rain water problem (array heights). [2]
41. Spiral order print of matrix. [5]
42. Rotate N×N matrix by 90 degrees in-place. [6]
43. Longest substring without repeating characters (sliding window). [5][3]
44. Longest palindromic substring (expand-around-center). [3]
45. Generate all permutations of a string (backtracking). [2]
46. Generate all subsets (power set) of array (bitmask or recursion). [2]
47. N-Queens (small N: 4 లేదా 5) valid arrangements count. [8]
48. Balanced brackets check using stack – (), {}, [].
49. Implement stack using queues / queue using stacks. [8]
50. Graph intro type: number of connected components in undirected graph (DFS/BFS) లేదా gridలో islands count (0/1 matrix). [9]

***

ఈ 50 problems ని మీరు sheet లా split చేసి:
- **Easy:** basic arrays/strings/loops.
- **Medium:** searching, sorting, prefix sums, maps, structures.
- **Hard:** standard DSA patterns (Kadane, sliding window, backtracking, graph).

Citations:

[1] Array Coding Problems for Interviews https://www.geeksforgeeks.org/dsa/top-50-array-coding-problems-for-interviews/

[2] Top 100 Codes » PREP INSTA https://prepinsta.com/top-100-codes/

[3] String Coding Interview Questions https://www.geeksforgeeks.org/dsa/top-50-string-coding-problems-for-interviews/

[4] Convert Decimal to Binary in C - GeeksforGeeks https://www.geeksforgeeks.org/c/c-decimal-to-binary/

[5] Top 15 DSA Questions Using Arrays and Strings ... https://getsdeready.com/top-15-dsa-questions-using-arrays-and-strings-for-coding-interviews/

[6] Top Array Interview Questions – Structured Path with Video ... https://takeuforward.org/array/top-array-interview-questions-structured-path-with-video-solutions/

[7] Top 100 Data Structure and Algorithms DSA Interview Questions ... https://www.geeksforgeeks.org/dsa/top-100-data-structure-and-algorithms-dsa-interview-questions-topic-wise/

[8] 30+ Sureshot DSA Interview Questions And Answers - HCL GUVI https://www.guvi.in/blog/dsa-interview-questions-and-answers/

[9] 50 Most Asked DSA Interview Questions to Practice in 2025 https://preplaced.in/blog/50-most-asked-dsa-interview-questions

[10] 1000142081.jpg https://ppl-ai-file-upload.s3.amazonaws.com/web/direct-files/attachments/images/84781331/81cebe5f-666f-406e-a530-9713d7e036fb/1000142081.jpg

[11] LeetCode's Array and String. https://leetcode.com/explore/learn/card/array-and-string/

[12] Practice Arrays https://www.codechef.com/practice/arrays

[13] College Contest MCQs - Java (Programming Language) https://www.scribd.com/document/838684361/College-Contest-MCQs

[14] Medium Hard Questions-Arrays/Strings | L 3 | Let's Leetcode https://www.youtube.com/watch?v=UHuKmUIUJqw

[15] Online Coding Practice Problems & Challenges https://www.codechef.com/practice

[16] Top 10 DSA Questions for Coding Tests - Get SDE Ready https://getsdeready.com/top-10-dsa-questions-for-coding-tests/

[17] Top 50 C Coding Interview Questions and Answers (2025) https://www.geeksforgeeks.org/c/c-coding-interview-questions/

[18] C programming Exercises, Practice, Solution https://www.w3resource.com/c-programming-exercises/

[19] DSA Practice - Interview Questions - workat.tech
https://workat.tech/problem-solving/practice/topics
[20] Competitive Programming Contest Problems
https://online.codingblocks.com/courses/competitive-programming-question-solving
[21] Solve Data Structures - HackerRank
https://www.hackerrank.com/domains/data-structures
[22] Programming Questions for Practice
http://www.btechsmartclass.com/coding-questions/

****
****
****

## 1) Array sum & average

### Java
```java
import java.util.*;

class Main {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      int n = sc.nextInt();
      int[] a = new int[n];
      int sum = 0;
      for (int i = 0; i < n; i++) {
         a[i] = sc.nextInt();
         sum += a[i];
      }
      double avg = (double) sum / n;
      System.out.println("Sum = " + sum);
      System.out.println("Avg = " + avg);
   }
```

}

### Python 3
```python
n = int(input())
arr = list(map(int, input().split()))
s = sum(arr)
avg = s / n
print("Sum =", s)
print("Avg =", avg)
```

***

## 2) Array max & min

### Java
```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] a = new int[n];
        for (int i = 0; i < n; i++) a[i] = sc.nextInt();

        int max = a[0], min = a[0];
        for (int i = 1; i < n; i++) {
            if (a[i] > max) max = a[i];
            if (a[i] < min) min = a[i];
        }
        System.out.println("Max = " + max);
        System.out.println("Min = " + min);
    }
}
```

### Python 3

```python
n = int(input())
arr = list(map(int, input().split()))
print("Max =", max(arr))
print("Min =", min(arr))
```

***

## 3) Reverse array (in-place)

### Java
```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] a = new int[n];
        for (int i = 0; i < n; i++) a[i] = sc.nextInt();

        int i = 0, j = n - 1;
        while (i < j) {
            int temp = a[i];
            a[i] = a[j];
            a[j] = temp;
            i++; j--;
        }
        for (int x : a) System.out.print(x + " ");
    }
}
```

### Python 3
```python
n = int(input())
arr = list(map(int, input().split()))
arr.reverse()
print(*arr)
```

***

## 4) Linear search

### Java
```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] a = new int[n];
        for (int i = 0; i < n; i++) a[i] = sc.nextInt();
        int key = sc.nextInt();

        int pos = -1;
        for (int i = 0; i < n; i++) {
            if (a[i] == key) {
                pos = i;
                break;
            }
        }
        if (pos == -1) System.out.println("Not found");
        else System.out.println("Found at index " + pos);
    }
}
```

### Python 3
```python
n = int(input())
arr = list(map(int, input().split()))
key = int(input())
pos = -1
for i, v in enumerate(arr):
    if v == key:
        pos = i
```

        break
print("Found at index", pos) if pos != -1 else print("Not found")

***

## 5) String length, vowels, consonants

### Java
```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String s = sc.nextLine();
        int len = s.length(), v = 0, c = 0;

        for (int i = 0; i < len; i++) {
            char ch = Character.toLowerCase(s.charAt(i));
            if (ch >= 'a' && ch <= 'z') {
                if ("aeiou".indexOf(ch) != -1) v++;
                else c++;
            }
        }
        System.out.println("Length = " + len);
        System.out.println("Vowels = " + v);
        System.out.println("Consonants = " + c);
    }
}
```

### Python 3
```python
s = input()
vowels = set("aeiouAEIOU")
v = c = 0
for ch in s:
    if ch.isalpha():
        if ch in vowels:
```

```
        v += 1
    else:
        c += 1
print("Length =", len(s))
print("Vowels =", v)
print("Consonants =", c)
```

***

## 6) String palindrome

### Java
```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String s = sc.nextLine();
        int i = 0, j = s.length() - 1;
        boolean ok = true;

        while (i < j) {
            if (s.charAt(i) != s.charAt(j)) {
                ok = false; break;
            }
            i++; j--;
        }
        System.out.println(ok ? "Palindrome" : "Not Palindrome");
    }
}
```

### Python 3
```python
s = input()
print("Palindrome" if s == s[::-1] else "Not Palindrome")
```

***

## 7) Compare two strings (equal or not)

### Java
```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String a = sc.nextLine();
        String b = sc.nextLine();

        if (a.equals(b)) System.out.println("Equal");
        else System.out.println("Not Equal");
    }
}
```

### Python 3
```python
a = input().strip()
b = input().strip()
print("Equal" if a == b else "Not Equal")
```

***

## 8) First non-repeated character

### Java
```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String s = sc.nextLine();
        int[] freq = new int[256];
        for (char ch : s.toCharArray()) freq[ch]++;

        char ans = 0;
        for (char ch : s.toCharArray()) {
            if (freq[ch] == 1) { ans = ch; break; }
        }
        if (ans == 0) System.out.println("None");
        else System.out.println(ans);
    }
}
```

### Python 3

```python
s = input()
from collections import Counter
freq = Counter(s)
ans = None
for ch in s:
    if freq[ch] == 1:
        ans = ch
        break
print(ans if ans is not None else "None")
```

***

## 9) Count primes up to N

### Java

```java
import java.util.*;

class Main {
    static boolean isPrime(int n) {
        if (n <= 1) return false;
        for (int i = 2; i * i <= n; i++)
            if (n % i == 0) return false;
        return true;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt(), count = 0;
        for (int i = 2; i <= n; i++)
            if (isPrime(i)) count++;
        System.out.println("Count = " + count);
    }
}
```

### Python 3

```python
import math

def is_prime(x):
    if x <= 1: return False
    for i in range(2, int(math.isqrt(x)) + 1):
        if x % i == 0:
            return False
    return True

n = int(input())
cnt = sum(1 for i
```

***

## 9) Count primes up to N

C

```c
#include <stdio.h>
#include <math.h>

int isPrime(int n) {
    int i;
    if (n <= 1) return 0;
    for (i = 2; i <= (int)sqrt(n); i++)
        if (n % i == 0) return 0;
    return 1;
}

int main() {
    int n, i, count = 0;
    scanf("%d", &n);
    for (i = 2; i <= n; i++)
        if (isPrime(i)) count++;
    printf("%d\n", count);
    return 0;
}
```

### Java
```java
import java.util.*;

class Main {
    static boolean isPrime(int n) {
        if (n <= 1) return false;
        for (int i = 2; i * i <= n; i++)
            if (n % i == 0) return false;
        return true;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt(), cnt = 0;
        for (int i = 2; i <= n; i++)
            if (isPrime(i)) cnt++;
        System.out.println(cnt);
    }
}
```

### Python 3
```python
import math

def is_prime(x):
    if x <= 1:
        return False
```

    for i in range(2, int(math.isqrt(x)) + 1):
        if x % i == 0:
            return False
    return True

n = int(input())
cnt = sum(1 for i in range(2, n + 1) if is_prime(i))
print(cnt)

***

## 10) Fibonacci first N terms

C
```c
#include <stdio.h>

int main() {
    int n, i;
    long long a = 0, b = 1, c;
    scanf("%d", &n);

    if (n >= 1) printf("%lld ", a);
    if (n >= 2) printf("%lld ", b);

    for (i = 3; i <= n; i++) {
        c = a + b;
        printf("%lld ", c);
        a = b;
        b = c;
    }
    printf("\n");
    return 0;
}
```

### Java
```java
import java.util.*;
```

```
class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        long a = 0, b = 1, c;

        if (n >= 1) System.out.print(a + " ");
        if (n >= 2) System.out.print(b + " ");
        for (int i = 3; i <= n; i++) {
            c = a + b;
            System.out.print(c + " ");
            a = b;
            b = c;
        }
    }
}
```

### Python 3

```python
n = int(input())
a, b = 0, 1
out = []
if n >= 1:
    out.append(str(a))
if n >= 2:
    out.append(str(b))
for _ in range(3, n + 1):
    a, b = b, a + b
    out.append(str(b))
print(" ".join(out))
```

***

## 11) Factorial of N (iterative)

C

```c
```

~~~
#include <stdio.h>

int main() {
    int n, i;
    long long fact = 1;
    scanf("%d", &n);

    if (n < 0) {
        printf("Invalid\n");
        return 0;
    }

    for (i = 1; i <= n; i++)
        fact *= i;

    printf("%lld\n", fact);
    return 0;
}
```

### Java
```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        if (n < 0) {
            System.out.println("Invalid");
            return;
        }
        long fact = 1;
        for (int i = 1; i <= n; i++) fact *= i;
        System.out.println(fact);
    }
}
```

### Python 3
~~~

```python
n = int(input())
if n < 0:
    print("Invalid")
else:
    fact = 1
    for i in range(1, n + 1):
        fact *= i
    print(fact)
```

***

## 12) 2D matrix row & column sums

C

```c
#include <stdio.h>

int main() {
    int r, c, i, j;
    int a[10][10];
    scanf("%d %d", &r, &c);

    for (i = 0; i < r; i++)
        for (j = 0; j < c; j++)
            scanf("%d", &a[i][j]);

    for (i = 0; i < r; i++) {
        int rs = 0;
        for (j = 0; j < c; j++) rs += a[i][j];
        printf("Row %d sum = %d\n", i, rs);
    }
    for (j = 0; j < c; j++) {
        int cs = 0;
        for (i = 0; i < r; i++) cs += a[i][j];
        printf("Col %d sum = %d\n", j, cs);
    }
    return 0;
}
```

### Java
```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int r = sc.nextInt(), c = sc.nextInt();
        int[][] a = new int[r][c];

        for (int i = 0; i < r; i++)
            for (int j = 0; j < c; j++)
                a[i][j] = sc.nextInt();

        for (int i = 0; i < r; i++) {
            int sum = 0;
            for (int j = 0; j < c; j++) sum += a[i][j];
            System.out.println("Row " + i + " sum = " + sum);
        }
        for (int j = 0; j < c; j++) {
            int sum = 0;
            for (int i = 0; i < r; i++) sum += a[i][j];
            System.out.println("Col " + j + " sum = " + sum);
        }
    }
}
```

### Python 3
```python
r, c = map(int, input().split())
a = [list(map(int, input().split())) for _ in range(r)]

for i in range(r):
    print(f"Row {i} sum = {sum(a[i])}")

for j in range(c):
    col_sum = sum(a[i][j] for i in range(r))
    print(f"Col {j} sum = {col_sum}")
```

***

## 13) Pass/Fail count (>=40 pass)

C
```c
#include <stdio.h>

int main() {
    int n, i, mark;
    int pass = 0, fail = 0;
    scanf("%d", &n);
    for (i = 0; i < n; i++) {
        scanf("%d", &mark);
        if (mark >= 40) pass++;
        else fail++;
    }
    printf("Pass = %d\nFail = %d\n", pass, fail);
    return 0;
}
```

### Java
```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int pass = 0, fail = 0;
        for (int i = 0; i < n; i++) {
            int m = sc.nextInt();
            if (m >= 40) pass++;
            else fail++;
        }
        System.out.println("Pass = " + pass);
        System.out.println("Fail = " + fail);
    }
}
```

### Python 3
```python
n = int(input())
marks = list(map(int, input().split()))
pass_cnt = sum(1 for m in marks if m >= 40)
fail_cnt = n - pass_cnt
print("Pass =", pass_cnt)
print("Fail =", fail_cnt)
```

***

## 14) Simple & Compound Interest

C
```c
#include <stdio.h>
#include <math.h>

int main() {
    double P, R, T;
    scanf("%lf %lf %lf", &P, &R, &T);

    double SI = (P * R * T) / 100.0;
    double A = P * pow(1 + R / 100.0, T);
    double CI = A - P;

    printf("SI = %.2f\n", SI);
    printf("CI = %.2f\n", CI);
    return 0;
}
```

### Java
```java
```

```
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        double P = sc.nextDouble();
        double R = sc.nextDouble();
        double T = sc.nextDouble();

        double SI = (P * R * T) / 100.0;
        double A = P * Math.pow(1 + R / 100.0, T);
        double CI = A - P;

        System.out.printf("SI = %.2f%n", SI);
        System.out.printf("CI = %.2f%n", CI);
    }
}
```

### Python 3

```python
P, R, T = map(float, input().split())
SI = (P * R * T) / 100.0
A = P * (1 + R / 100.0) ** T
CI = A - P
print(f"SI = {SI:.2f}")
print(f"CI = {CI:.2f}")
```

***

## 15) Decimal ↔ Binary (here: Decimal to Binary)

C

```c
#include <stdio.h>

int main() {
    int n, i = 0, b[32];
    scanf("%d", &n);
```

~~~
    if (n == 0) {
        printf("0\n");
        return 0;
    }

    while (n > 0) {
        b[i++] = n % 2;
        n /= 2;
    }

    for (i = i - 1; i >= 0; i--)
        printf("%d", b[i]);
    printf("\n");
    return 0;
}
```
~~~

### Java

~~~
```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();

        if (n == 0) {
            System.out.println(0);
            return;
        }

        int[] bin = new int[32];
        int i = 0;
        while (n > 0) {
            bin[i++] = n % 2;
            n /= 2;
        }
        StringBuilder sb = new StringBuilder();
        for (int j = i - 1; j >= 0; j--)
~~~

            sb.append(bin[j]);
        System.out.println(sb.toString());
    }
}

### Python 3
```python
n = int(input())
if n == 0:
    print(0)
else:
    bits = []
    while n > 0:
        bits.append(str(n % 2))
        n //= 2
    print("".join(reversed(bits)))
```

## Medium – Batch 1 (Arrays basic)

ఈ batchకి problems ఇవి (ఇప్పుడే కోడ్ ఇస్తాను):
16) Second largest element in array
17) Remove duplicates / print unique elements
18) Frequency of each element (map/dict)
19) Kth smallest & Kth largest (sort ఆధారంగా)
23) Recursive Binary Search

ఇప్పుడు ప్రతి problemకి C, Java, Python skeletons:

***

### 16) Second largest element

C
```c
#include <stdio.h>
#include <limits.h>
```

```
int main() {
    int n, a[1000];
    scanf("%d", &n);
    for (int i = 0; i < n; i++) scanf("%d", &a[i]);

    int first = INT_MIN, second = INT_MIN;
    for (int i = 0; i < n; i++) {
        if (a[i] > first) {
            second = first;
            first = a[i];
        } else if (a[i] > second && a[i] < first) {
            second = a[i];
        }
    }
    if (second == INT_MIN) printf("No second largest\n");
    else printf("%d\n", second);
    return 0;
}
```

Java

```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] a = new int[n];
        for (int i = 0; i < n; i++) a[i] = sc.nextInt();

        Integer first = null, second = null;
        for (int x : a) {
            if (first == null || x > first) {
                second = first;
                first = x;
            } else if (x != first && (second == null || x > second)) {
                second = x;
            }
        }
        if (second == null) System.out.println("No second largest");
        else System.out.println(second);
    }
}
```

Python 3
```python
n = int(input())
arr = list(map(int, input().split()))
uniq = sorted(set(arr))
if len(uniq) < 2:
    print("No second largest")
else:
    print(uniq[-2])
```

***

### 17) Print unique elements (remove duplicates, order as first occurrence)

C
```c
#include <stdio.h>

int main() {
    int n, a[1000];
    scanf("%d", &n);
    for (int i = 0; i < n; i++) scanf("%d", &a[i]);

    for (int i = 0; i < n; i++) {
        int seen = 0;
        for (int j = 0; j < i; j++)
            if (a[j] == a[i]) { seen = 1; break; }
        if (!seen) printf("%d ", a[i]);
    }
    printf("\n");
    return 0;
}
```

Java

```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] a = new int[n];
        for (int i = 0; i < n; i++) a[i] = sc.nextInt();

        Set<Integer> seen = new HashSet<>();
        StringBuilder sb = new StringBuilder();
        for (int x : a) {
            if (!seen.contains(x)) {
                seen.add(x);
                sb.append(x).append(' ');
            }
        }
        System.out.println(sb.toString().trim());
    }
}
```

Python 3

```python
n = int(input())
arr = list(map(int, input().split()))
seen = set()
out = []
for x in arr:
    if x not in seen:
        seen.add(x)
        out.append(str(x))
print(" ".join(out))
```

***

### 18) Frequency of each element

C
```c
#include <stdio.h>

int main() {
    int n, a[1000];
    scanf("%d", &n);
    for (int i = 0; i < n; i++) scanf("%d", &a[i]);

    int used[1000] = {0};
    for (int i = 0; i < n; i++) {
        if (used[i]) continue;
        int cnt = 1;
        for (int j = i + 1; j < n; j++) {
            if (a[j] == a[i]) {
                cnt++;
                used[j] = 1;
            }
        }
        printf("%d -> %d\n", a[i], cnt);
    }
    return 0;
}
```

Java
```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        Map<Integer,Integer> freq = new LinkedHashMap<>();
        for (int i = 0; i < n; i++) {
            int x = sc.nextInt();
            freq.put(x, freq.getOrDefault(x, 0) + 1);
        }
        for (var e : freq.entrySet())
            System.out.println(e.getKey() + " -> " + e.getValue());
    }
}
```

Python 3
```python
from collections import Counter

n = int(input())
arr = list(map(int, input().split()))
for x, c in Counter(arr).items():
    print(f"{x} -> {c}")
```

***

### 19) Kth smallest & Kth largest (sort)

C
```c
#include <stdio.h>
#include <stdlib.h>

int cmp(const void *p, const void *q) {
    int x = *(int*)p, y = *(int*)q;
    return x - y;
}

int main() {
    int n, k, a[1000];
    scanf("%d %d", &n, &k);
    for (int i = 0; i < n; i++) scanf("%d", &a[i]);

    qsort(a, n, sizeof(int), cmp);

    if (k <= 0 || k > n) {
        printf("Invalid\n");
        return 0;
    }
    int kthSmall = a[k-1];
    int kthLarge = a[n-k];
    printf("Kth small = %d\nKth large = %d\n", kthSmall, kthLarge);
    return 0;
}
```

Java
```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int k = sc.nextInt();
        int[] a = new int[n];
        for (int i = 0; i < n; i++) a[i] = sc.nextInt();

        Arrays.sort(a);
        if (k <= 0 || k > n) {
            System.out.println("Invalid");
            return;
        }
        System.out.println("Kth small = " + a[k-1]);
        System.out.println("Kth large = " + a[n-k]);
    }
}
```

Python 3
```python
n, k = map(int, input().split())
arr = sorted(map(int, input().split()))
if k <= 0 or k > n:
    print("Invalid")
else:
    print("Kth small =", arr[k-1])
```

    print("Kth large =", arr[-k])

***

### 23) Recursive Binary Search (array must be sorted)

C
```c
#include <stdio.h>

int binSearch(int a[], int l, int r, int key) {
    if (l > r) return -1;
    int m = (l + r) / 2;
    if (a[m] == key) return m;
    if (key < a[m]) return binSearch(a, l, m - 1, key);
    else return binSearch(a, m + 1, r, key);
}

int main() {
    int n, a[1000], key;
    scanf("%d", &n);
    for (int i = 0; i < n; i++) scanf("%d", &a[i]);
    scanf("%d", &key);
    int idx = binSearch(a, 0, n - 1, key);
    printf("%d\n", idx);
    return 0;
}
```

Java
```java
import java.util.*;

class Main {
    static int binSearch(int[] a, int l, int r, int key) {
        if (l > r) return -1;
        int m = (l + r) / 2;
        if (a[m] == key) return m;
        if (key < a[m]) return binSearch(a, l, m - 1, key);
        return binSearch(a, m + 1, r, key);
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] a = new int[n];
        for (int i = 0; i < n; i++) a[i] = sc.nextInt();
        int key = sc.nextInt();
        System.out.println(binSearch(a, 0, n - 1, key));
    }
}
```

Python 3

```python
def bin_search(a, l, r, key):
    if l > r:
        return -1
    m = (l + r) // 2
    if a[m] == key:
        return m
    if key < a[m]:
        return bin_search(a, l, m - 1, key)
    return bin_search(a, m + 1, r, key)

n = int(input())
arr = list(map(int, input().split()))
key = int(input())
print(bin_search(arr, 0, n - 1, key))
```

***

merge, rotation, anagram, prefix sum, majority element, stock buy–sell problems **

## 1) Merge two sorted arrays (new merged array)

C

```c
#include <stdio.h>

int main() {
    int n, m, i = 0, j = 0, k = 0;
    int a[1000], b[1000], c[2000];

    scanf("%d %d", &n, &m);
    for (i = 0; i < n; i++) scanf("%d", &a[i]);
    for (j = 0; j < m; j++) scanf("%d", &b[j]);

    i = j = 0;
    while (i < n && j < m) {
        if (a[i] <= b[j]) c[k++] = a[i++];
        else c[k++] = b[j++];
    }
    while (i < n) c[k++] = a[i++];
    while (j < m) c[k++] = b[j++];

    for (i = 0; i < k; i++) printf("%d ", c[i]);
    printf("\n");
    return 0;
}
```

### Java

```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt(), m = sc.nextInt();
        int[] a = new int[n], b = new int[m];
        for (int i = 0; i < n; i++) a[i] = sc.nextInt();
        for (int i = 0; i < m; i++) b[i] = sc.nextInt();

        int[] c = new int[n + m];
        int i = 0, j = 0, k = 0;
        while (i < n && j < m) {
            if (a[i] <= b[j]) c[k++] = a[i++];
            else c[k++] = b[j++];
        }
        while (i < n) c[k++] = a[i++];
        while (j < m) c[k++] = b[j++];

        for (int x : c) System.out.print(x + " ");
    }
}
```

### Python 3

```python
n, m = map(int, input().split())
a = list(map(int, input().split()))
b = list(map(int, input().split()))

i = j = 0
c = []
while i < n and j < m:
    if a[i] <= b[j]:
        c.append(a[i]); i += 1
    else:
        c.append(b[j]); j += 1
c.extend(a[i:])
c.extend(b[j:])
print(*c)
```

***

## 2) Left rotate array by K (reversal method)

C

```c
#include <stdio.h>

void reverse(int a[], int l, int r) {
    while (l < r) {
        int t = a[l]; a[l] = a[r]; a[r] = t;
```

~~~~
        l++; r--;
    }
}

int main() {
    int n, k, a[1000];
    scanf("%d %d", &n, &k);
    for (int i = 0; i < n; i++) scanf("%d", &a[i]);

    k %= n;
    reverse(a, 0, k - 1);
    reverse(a, k, n - 1);
    reverse(a, 0, n - 1);

    for (int i = 0; i < n; i++) printf("%d ", a[i]);
    printf("\n");
    return 0;
}
```

### Java
```java
import java.util.*;

class Main {
    static void reverse(int[] a, int l, int r) {
        while (l < r) {
            int t = a[l]; a[l] = a[r]; a[r] = t;
            l++; r--;
        }
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt(), k = sc.nextInt();
        int[] a = new int[n];
        for (int i = 0; i < n; i++) a[i] = sc.nextInt();

        k %= n;
        reverse(a, 0, k - 1);
        reverse(a, k, n - 1);
~~~~

```
        reverse(a, 0, n - 1);

        for (int x : a) System.out.print(x + " ");
    }
}
```

### Python 3
```python
n, k = map(int, input().split())
a = list(map(int, input().split()))
k %= n
a = a[k:] + a[:k]
print(*a)
```

***

## 3) Check two strings are anagrams

C
```c
#include <stdio.h>
#include <string.h>

int main() {
    char s1[105], s2[105];
    int c1[256] = {0}, c2[256] = {0};
    scanf("%s %s", s1, s2);

    if (strlen(s1) != strlen(s2)) {
        printf("Not Anagram\n");
        return 0;
    }
    for (int i = 0; s1[i]; i++) c1[(unsigned char)s1[i]]++;
    for (int i = 0; s2[i]; i++) c2[(unsigned char)s2[i]]++;

    for (int i = 0; i < 256; i++)
        if (c1[i] != c2[i]) {
            printf("Not Anagram\n");
            return 0;
        }
    printf("Anagram\n");
    return 0;
}
```

### Java

```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String a = sc.next().toLowerCase();
        String b = sc.next().toLowerCase();

        if (a.length() != b.length()) {
            System.out.println("Not Anagram");
            return;
        }
        char[] ca = a.toCharArray();
        char[] cb = b.toCharArray();
        Arrays.sort(ca);
        Arrays.sort(cb);
        System.out.println(Arrays.equals(ca, cb) ? "Anagram" : "Not Anagram");
    }
}
```

### Python 3

```python
a = input().strip().lower()
b = input().strip().lower()
print("Anagram" if sorted(a) == sorted(b) else "Not Anagram")
```

***

## 4) Prefix sum + Range Sum Query [L,R]

C
```c
#include <stdio.h>

int main() {
    int n, q, a[100000];
    long long pref[100001];
    scanf("%d", &n);
    for (int i = 0; i < n; i++) scanf("%d", &a[i]);

    pref[0] = 0;
    for (int i = 1; i <= n; i++)
        pref[i] = pref[i-1] + a[i-1];

    scanf("%d", &q);
    while (q--) {
        int L, R;
        scanf("%d %d", &L, &R); // 0-based
        long long ans = pref[R+1] - pref[L];
        printf("%lld\n", ans);
    }
    return 0;
}
```

### Java
```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        long[] pref = new long[n + 1];
        for (int i = 1; i <= n; i++) {
            int x = sc.nextInt();
            pref[i] = pref[i-1] + x;
        }
        int q = sc.nextInt();
        while (q-- > 0) {
            int L = sc.nextInt();
            int R = sc.nextInt();
            long ans = pref[R + 1] - pref[L];
            System.out.println(ans);
        }
    }
}
```

### Python 3

```python
n = int(input())
arr = list(map(int, input().split()))
pref = [0]
for x in arr:
    pref.append(pref[-1] + x)

q = int(input())
for _ in range(q):
    L, R = map(int, input().split())
    print(pref[R+1] - pref[L])
```

***

## 5) Majority element (frequency > n/2, assumed exists or print “None”)

C

```c
#include <stdio.h>

int main() {
    int n, a[100000];
    scanf("%d", &n);
    for (int i = 0; i < n; i++) scanf("%d", &a[i]);

    int cand = a[0], cnt = 1;
    for (int i = 1; i < n; i++) {
        if (a[i] == cand) cnt++;
        else if (--cnt == 0) { cand = a[i]; cnt = 1; }
    }
    cnt = 0;
    for (int i = 0; i < n; i++)
        if (a[i] == cand) cnt++;

    if (cnt > n/2) printf("%d\n", cand);
    else printf("None\n");
    return 0;
}
```

### Java
```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] a = new int[n];
        for (int i = 0; i < n; i++) a[i] = sc.nextInt();

        int cand = a[0], cnt = 1;
        for (int i = 1; i < n; i++) {
            if (a[i] == cand) cnt++;
            else if (--cnt == 0) { cand = a[i]; cnt = 1; }
        }
        cnt = 0;
        for (int x : a) if (x == cand) cnt++;
        System.out.println(cnt > n/2 ? cand : "None");
    }
}
```

### Python 3
```python
n = int(input())
a = list(map(int, input().split()))
```

```python
cand, cnt = a[0], 1
for x in a[1:]:
    if x == cand:
        cnt += 1
    else:
        cnt -= 1
        if cnt == 0:
            cand, cnt = x, 1

if a.count(cand) > n // 2:
    print(cand)
else:
    print("None")
```

***

## 6) Stock buy–sell once (max profit)

C
```c
#include <stdio.h>

int main() {
    int n, i;
    int p[100000];
    scanf("%d", &n);
    for (i = 0; i < n; i++) scanf("%d", &p[i]);

    int minPrice = p[0];
    int maxProfit = 0;
    for (i = 1; i < n; i++) {
        if (p[i] - minPrice > maxProfit)
            maxProfit = p[i] - minPrice;
        if (p[i] < minPrice)
            minPrice = p[i];
    }
    printf("%d\n", maxProfit);
    return 0;
}
```

~~~
```
~~~

### Java

```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] price = new int[n];
        for (int i = 0; i < n; i++) price[i] = sc.nextInt();

        int minPrice = price[0], maxProfit = 0;
        for (int i = 1; i < n; i++) {
            maxProfit = Math.max(maxProfit, price[i] - minPrice);
            minPrice = Math.min(minPrice, price[i]);
        }
        System.out.println(maxProfit);
    }
}
```

### Python 3

```python
n = int(input())
price = list(map(int, input().split()))

min_price = price[0]
max_profit = 0
for x in price[1:]:
    if x - min_price > max_profit:
        max_profit = x - min_price
    if x < min_price:
        min_price = x
print(max_profit)
```

***

##### hard level:

Kadane, Balanced Brackets, Longest Substring Without Repeating Characters.

***

## 1) Kadane’s Algorithm – Maximum Subarray Sum

C
```c
#include <stdio.h>
#include <limits.h>

int main() {
    int n, i;
    scanf("%d", &n);
    int a[100000];
    for (i = 0; i < n; i++) scanf("%d", &a[i]);

    long long max_end = a[0], max_so_far = a[0];
    for (i = 1; i < n; i++) {
        if (max_end + a[i] > a[i]) max_end = max_end + a[i];
        else max_end = a[i];
        if (max_end > max_so_far) max_so_far = max_end;
    }
    printf("%lld\n", max_so_far);
    return 0;
}
```

### Java
```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] a = new int[n];
        for (int i = 0; i < n; i++) a[i] = sc.nextInt();

        long maxEnd = a[0], maxSoFar = a[0];
        for (int i = 1; i < n; i++) {
            maxEnd = Math.max(a[i], maxEnd + a[i]);
            maxSoFar = Math.max(maxSoFar, maxEnd);
        }
        System.out.println(maxSoFar);
    }
}
```

### Python 3
```python
n = int(input())
arr = list(map(int, input().split()))

max_end = max_so_far = arr[0]
for x in arr[1:]:
    max_end = max(x, max_end + x)
    max_so_far = max(max_so_far, max_end)
print(max_so_far)
```

***

## 2) Balanced Brackets (Stack)

C
```c
#include <stdio.h>

char stack[100000];
int top = -1;

void push(char c) { stack[++top] = c; }
char pop() { return stack[top--]; }
```

~~~~
int isMatching(char open, char close) {
  return (open == '(' && close == ')') ||
      (open == '{' && close == '}') ||
      (open == '[' && close == ']');
}

int main() {
  char s[100005];
  scanf("%s", s);

  for (int i = 0; s[i] != '\0'; i++) {
    char ch = s[i];
    if (ch == '(' || ch == '{' || ch == '[') {
      push(ch);
    } else if (ch == ')' || ch == '}' || ch == ']') {
      if (top == -1) { printf("NO\n"); return 0; }
      char open = pop();
      if (!isMatching(open, ch)) { printf("NO\n"); return 0; }
    }
  }
  if (top == -1) printf("YES\n");
  else printf("NO\n");
  return 0;
}
```

### Java
```java
import java.util.*;

class Main {
  static boolean isBalanced(String s) {
    Stack<Character> st = new Stack<>();
    for (char ch : s.toCharArray()) {
      if (ch == '(' || ch == '{' || ch == '[') {
        st.push(ch);
      } else if (ch == ')' || ch == '}' || ch == ']') {
        if (st.isEmpty()) return false;
        char open = st.pop();
        if (!((open == '(' && ch == ')') ||
~~~~

```
                      (open == '{' && ch == '}') ||
                      (open == '[' && ch == ']')))
                    return false;
            }
        }
        return st.isEmpty();
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String s = sc.next();
        System.out.println(isBalanced(s) ? "YES" : "NO");
    }
}
```

### Python 3

```python
s = input().strip()
stack = []
pairs = {')': '(', '}': '{', ']': '['}
ok = True

for ch in s:
    if ch in "({[":
        stack.append(ch)
    elif ch in ")}]":
        if not stack or stack[-1] != pairs[ch]:
            ok = False
            break
        stack.pop()

if ok and not stack:
    print("YES")
else:
    print("NO")
```

***

## 3) Longest Substring Without Repeating Characters (Sliding Window)

C
```c
#include <stdio.h>

int last[256];

int main() {
    char s[100005];
    scanf("%s", s);

    for (int i = 0; i < 256; i++) last[i] = -1;

    int start = 0, maxLen = 0;
    for (int i = 0; s[i] != '\0'; i++) {
        unsigned char ch = s[i];
        if (last[ch] >= start)
            start = last[ch] + 1;
        last[ch] = i;
        int curLen = i - start + 1;
        if (curLen > maxLen) maxLen = curLen;
    }
    printf("%d\n", maxLen);
    return 0;
}
```

### Java
```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String s = sc.next();
        int[] last = new int[256];
        Arrays.fill(last, -1);

        int start = 0, maxLen = 0;
        for (int i = 0; i < s.length(); i++) {
            char ch = s.charAt(i);
            if (last[ch] >= start)
                start = last[ch] + 1;
            last[ch] = i;
            int curLen = i - start + 1;
            if (curLen > maxLen) maxLen = curLen;
        }
        System.out.println(maxLen);
    }
}
```

### Python 3
```python
s = input().strip()
last = {}
start = 0
max_len = 0

for i, ch in enumerate(s):
    if ch in last and last[ch] >= start:
        start = last[ch] + 1
    last[ch] = i
    max_len = max(max_len, i - start + 1)

print(max_len)
```

***

***

## 1) Move all zeros to end (order maintain)

C
```c
#include <stdio.h>

int main() {
    int n, i;
    int a[100000];
    scanf("%d", &n);
    for (i = 0; i < n; i++) scanf("%d", &a[i]);

    int pos = 0;
    for (i = 0; i < n; i++) {
        if (a[i] != 0) {
            a[pos] = a[i];
            pos++;
        }
    }
    while (pos < n) {
        a[pos] = 0;
        pos++;
    }

    for (i = 0; i < n; i++) printf("%d ", a[i]);
    printf("\n");
    return 0;
}
```

### Java

```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] a = new int[n];
        for (int i = 0; i < n; i++) a[i] = sc.nextInt();

        int pos = 0;
        for (int i = 0; i < n; i++)
            if (a[i] != 0) a[pos++] = a[i];
        while (pos < n) a[pos++] = 0;

        for (int x : a) System.out.print(x + " ");
    }
}
```

### Python 3
```python
n = int(input())
a = list(map(int, input().split()))

pos = 0
for x in a:
    if x != 0:
        a[pos] = x
        pos += 1
while pos < n:
    a[pos] = 0
    pos += 1

print(*a)
```

***

## 2) Longest subarray with sum = K (positives & negatives)

C
```c
#include <stdio.h>
#include <stdlib.h>

struct Pair { long long sum; int index; };

int hash(long long x, int m) {
    if (x < 0) x = -x;
    return (int)(x % m);
}

int main() {
    int n, i;
    long long k;
```

```
scanf("%d %lld", &n, &k);
int a[100000];
for (i = 0; i < n; i++) scanf("%d", &a[i]);

int M = 200003;
struct Pair *h = malloc(M * sizeof(struct Pair));
for (i = 0; i < M; i++) { h[i].index = -1; h[i].sum = 0; }

long long prefix = 0;
int maxLen = 0;

long long key = 0;
int idx = hash(0, M);
h[idx].sum = 0;
h[idx].index = 0;

for (i = 0; i < n; i++) {
   prefix += a[i];

   long long need = prefix - k;
   int j = hash(need, M);
   while (h[j].index != -1 && h[j].sum != need) {
      j = (j + 1) % M;
   }
   if (h[j].index != -1 && h[j].sum == need) {
      int len = i + 1 - h[j].index;
      if (len > maxLen) maxLen = len;
   }

   key = prefix;
   int p = hash(key, M);
   while (h[p].index != -1 && h[p].sum != key) {
      p = (p + 1) % M;
   }
   if (h[p].index == -1) {
      h[p].sum = key;
      h[p].index = i + 1;
   }
}
```

```
    printf("%d\n", maxLen);
    free(h);
    return 0;
}
```

### Java
```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        long k = sc.nextLong();
        int[] a = new int[n];
        for (int i = 0; i < n; i++) a[i] = sc.nextInt();

        Map<Long, Integer> first = new HashMap<>();
        long pref = 0;
        int maxLen = 0;
        first.put(0L, 0); // prefix sum 0 at index 0 (before array)

        for (int i = 0; i < n; i++) {
            pref += a[i];
            long need = pref - k;
            if (first.containsKey(need)) {
                int len = i + 1 - first.get(need);
                if (len > maxLen) maxLen = len;
            }
            if (!first.containsKey(pref))
                first.put(pref, i + 1);
        }
        System.out.println(maxLen);
    }
}
```

### Python 3
```python
```

```python
n, k = map(int, input().split())
arr = list(map(int, input().split()))

first = {0: 0}
pref = 0
max_len = 0

for i, x in enumerate(arr, start=1):
    pref += x
    need = pref - k
    if need in first:
        max_len = max(max_len, i - first[need])
    if pref not in first:
        first[pref] = i

print(max_len)
```

***

## 3) Spiral order print of matrix

C
```c
#include <stdio.h>

int main() {
    int r, c;
    int a[100][100];
    scanf("%d %d", &r, &c);
    for (int i = 0; i < r; i++)
        for (int j = 0; j < c; j++)
            scanf("%d", &a[i][j]);

    int top = 0, bottom = r - 1, left = 0, right = c - 1;
    while (top <= bottom && left <= right) {
        for (int j = left; j <= right; j++)
            printf("%d ", a[top][j]);
        top++;
        for (int i = top; i <= bottom; i++)
            printf("%d ", a[i][right]);
        right--;

        if (top <= bottom) {
            for (int j = right; j >= left; j--)
                printf("%d ", a[bottom][j]);
            bottom--;
        }
        if (left <= right) {
            for (int i = bottom; i >= top; i--)
                printf("%d ", a[i][left]);
            left++;
        }
    }
    printf("\n");
    return 0;
}
```

### Java
```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int r = sc.nextInt(), c = sc.nextInt();
        int[][] a = new int[r][c];
        for (int i = 0; i < r; i++)
            for (int j = 0; j < c; j++)
                a[i][j] = sc.nextInt();

        int top = 0, bottom = r - 1, left = 0, right = c - 1;
        StringBuilder sb = new StringBuilder();

        while (top <= bottom && left <= right) {
            for (int j = left; j <= right; j++)
                sb.append(a[top][j]).append(' ');
            top++;
```

~~~
            for (int i = top; i <= bottom; i++)
                sb.append(a[i][right]).append(' ');
            right--;

            if (top <= bottom) {
                for (int j = right; j >= left; j--)
                    sb.append(a[bottom][j]).append(' ');
                bottom--;
            }
            if (left <= right) {
                for (int i = bottom; i >= top; i--)
                    sb.append(a[i][left]).append(' ');
                left++;
            }
        }
        System.out.println(sb.toString().trim());
    }
}
```

### Python 3
```python
r, c = map(int, input().split())
a = [list(map(int, input().split())) for _ in range(r)]

top, bottom, left, right = 0, r - 1, 0, c - 1
out = []

while top <= bottom and left <= right:
    for j in range(left, right + 1):
        out.append(str(a[top][j]))
    top += 1

    for i in range(top, bottom + 1):
        out.append(str(a[i][right]))
    right -= 1

    if top <= bottom:
        for j in range(right, left - 1, -1):
~~~

            out.append(str(a[bottom][j]))
        bottom -= 1

    if left <= right:
        for i in range(bottom, top - 1, -1):
            out.append(str(a[i][left]))
        left += 1

print(" ".join(out))
~~~

***

4) Rotate N×N matrix by 90° clockwise (in-place)

C
```c
#include <stdio.h>

int main() {
    int n;
    int a[100][100];
    scanf("%d", &n);
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            scanf("%d", &a[i][j]);

    for (int i = 0; i < n; i++)
        for (int j = i + 1; j < n; j++) {
            int tmp = a[i][j];
            a[i][j] = a[j][i];
            a[j][i] = tmp;
        }

    for (int i = 0; i < n; i++) {
        int l = 0, r = n - 1;
        while (l < r) {
            int tmp = a[i][l];
            a[i][l] = a[i][r];
            a[i][r] = tmp;
~~~

```
            l++; r--;
        }
    }

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++)
            printf("%d ", a[i][j]);
        printf("\n");
    }
    return 0;
}
```

### Java

```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[][] a = new int[n][n];
        for (int i = 0; i < n; i++)
            for (int j = 0; j < n; j++)
                a[i][j] = sc.nextInt();

        for (int i = 0; i < n; i++)
            for (int j = i + 1; j < n; j++) {
                int tmp = a[i][j];
                a[i][j] = a[j][i];
                a[j][i] = tmp;
            }

        for (int i = 0; i < n; i++) {
            int l = 0, r = n - 1;
            while (l < r) {
                int tmp = a[i][l];
                a[i][l] = a[i][r];
                a[i][r] = tmp;
                l++; r--;
            }
        }

        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++)
                System.out.print(a[i][j] + " ");
            System.out.println();
        }
    }
}
```

### Python 3
```python
n = int(input())
a = [list(map(int, input().split())) for _ in range(n)]

for i in range(n):
    for j in range(i + 1, n):
        a[i][j], a[j][i] = a[j][i], a[i][j]

for i in range(n):
    a[i].reverse()

for row in a:
    print(*row)
```

***

## 1) Longest Increasing Subsequence (LIS) – $O(n^2)$

C
```c
#include <stdio.h>

int main() {
    int n, i, j;
    int a[1000], dp[1000];
    scanf("%d", &n);
    for (i = 0; i < n; i++) scanf("%d", &a[i]);

    int ans = 0;
    for (i = 0; i < n; i++) {
        dp[i] = 1;
        for (j = 0; j < i; j++)
            if (a[j] < a[i] && dp[j] + 1 > dp[i])
                dp[i] = dp[j] + 1;
        if (dp[i] > ans) ans = dp[i];
    }
    printf("%d\n", ans);
    return 0;
}
```

### Java

```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] a = new int[n];
        for (int i = 0; i < n; i++) a[i] = sc.nextInt();

        int[] dp = new int[n];
        int ans = 0;
        for (int i = 0; i < n; i++) {
            dp[i] = 1;
            for (int j = 0; j < i; j++)
                if (a[j] < a[i] && dp[j] + 1 > dp[i])
                    dp[i] = dp[j] + 1;
            ans = Math.max(ans, dp[i]);
        }
        System.out.println(ans);
    }
}
```

### Python 3
```python
n = int(input())
arr = list(map(int, input().split()))
dp = [1] * n
ans = 0

for i in range(n):
    for j in range(i):
        if arr[j] < arr[i] and dp[j] + 1 > dp[i]:
            dp[i] = dp[j] + 1
    ans = max(ans, dp[i])

print(ans)
```

***

## 2) Trapping Rain Water – two-pointer O(n)

C
```c
#include <stdio.h>

int main() {
    int n;
    int h[100000];
    scanf("%d", &n);
    for (int i = 0; i < n; i++) scanf("%d", &h[i]);

    int l = 0, r = n - 1;
    int leftMax = 0, rightMax = 0;
    long long water = 0;

    while (l < r) {
        if (h[l] < h[r]) {
            if (h[l] >= leftMax) leftMax = h[l];
            else water += leftMax - h[l];
            l++;
        } else {
            if (h[r] >= rightMax) rightMax = h[r];
            else water += rightMax - h[r];
            r--;
        }
    }
    printf("%lld\n", water);
    return 0;
}
```

### Java

```java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] h = new int[n];
        for (int i = 0; i < n; i++) h[i] = sc.nextInt();

        int l = 0, r = n - 1;
        int leftMax = 0, rightMax = 0;
        long water = 0;

        while (l < r) {
            if (h[l] < h[r]) {
                if (h[l] >= leftMax) leftMax = h[l];
                else water += leftMax - h[l];
                l++;
            } else {
                if (h[r] >= rightMax) rightMax = h[r];
                else water += rightMax - h[r];
                r--;
            }
        }
        System.out.println(water);
    }
}
```

### Python 3
```python
n = int(input())
h = list(map(int, input().split()))

l, r = 0, n - 1
left_max = right_max = 0
water = 0

while l < r:
    if h[l] < h[r]:
        if h[l] >= left_max:
            left_max = h[l]
        else:
            water += left_max - h[l]
        l += 1
    else:
        if h[r] >= right_max:
            right_max = h[r]
        else:
            water += right_max - h[r]
        r -= 1

print(water)
```

***

## 3) Longest Palindromic Substring – expand around center

Outputగా longest palindrome string print చేస్తుంది. [3][4]

C
```c
#include <stdio.h>
#include <string.h>

int expand(char s[], int left, int right) {
    int n = strlen(s);
    while (left >= 0 && right < n && s[left] == s[right]) {
        left--;
        right++;
    }
    return right - left - 1;
}

int main() {
    char s[1005];
    scanf("%s", s);
    int n = strlen(s);

    int start = 0, end = 0;
    for (int i = 0; i < n; i++) {
        int len1 = expand(s, i, i);
        int len2 = expand(s, i, i + 1);
        int len = len1 > len2 ? len1 : len2;
        if (len > end - start + 1) {
            start = i - (len - 1) / 2;
            end = i + len / 2;
        }
    }

    for (int i = start; i <= end; i++)
        putchar(s[i]);
    putchar('\n');
    return 0;
}
```

### Java
```java
import java.util.*;

class Main {
    static int expand(String s, int left, int right) {
        int n = s.length();
        while (left >= 0 && right < n && s.charAt(left) == s.charAt(right)) {
            left--;
            right++;
        }
        return right - left - 1;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String s = sc.next();
        int n = s.length();
        int start = 0, end = 0;

        for (int i = 0; i < n; i++) {
            int len1 = expand(s, i, i);
            int len2 = expand(s, i, i + 1);
            int len = Math.max(len1, len2);
            if (len > end - start + 1) {
                start = i - (len - 1) / 2;
                end = i + len / 2;
            }
        }
        System.out.println(s.substring(start, end + 1));
    }
}
```

### Python 3

```python
s = input().strip()
n = len(s)

def expand(l, r):
    while l >= 0 and r < n and s[l] == s[r]:
        l -= 1
        r += 1
    return r - l - 1, l + 1, r - 1

start = end = 0
for i in range(n):
    len1, s1, e1 = expand(i, i)
    len2, s2, e2 = expand(i, i + 1)
    if len1 > end - start + 1:
        start, end = s1, e1
    if len2 > end - start + 1:
        start, end = s2, e2

print(s[start:end+1])
```

***

## 1) All permutations of a string

C

```c
#include <stdio.h>
#include <string.h>

void swap(char *a, char *b) {
    char t = *a; *a = *b; *b = t;
}

void permute(char s[], int l, int r) {
    if (l == r) {
        printf("%s\n", s);
        return;
    }
    for (int i = l; i <= r; i++) {
        swap(&s[l], &s[i]);
        permute(s, l + 1, r);
        swap(&s[l], &s[i]); // backtrack
    }
}

int main() {
    char s[20];
    scanf("%s", s);
    permute(s, 0, strlen(s) - 1);
    return 0;
}
```

### Java

```java
import java.util.*;

class Main {
    static void permute(char[] s, int l) {
        if (l == s.length) {
            System.out.println(new String(s));
            return;
        }
        for (int i = l; i < s.length; i++) {
            char tmp = s[l]; s[l] = s[i]; s[i] = tmp;
            permute(s, l + 1);
            tmp = s[l]; s[l] = s[i]; s[i] = tmp;
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String str = sc.next();
        permute(str.toCharArray(), 0);
    }
}
```

### Python 3

```python
s = list(input().strip())

def permute(l):
    if l == len(s):
        print("".join(s))
        return
    for i in range(l, len(s)):
        s[l], s[i] = s[i], s[l]
        permute(l + 1)
        s[l], s[i] = s[i], s[l]
```

```
permute(0)
```

***

## 2) All subsets (power set) of array (backtracking)

C
```c
#include <stdio.h>

int n;
int a[20], cur[20];

void backtrack(int idx, int len) {
    if (idx == n) {
        printf("{");
        for (int i = 0; i < len; i++) {
            printf("%d", cur[i]);
            if (i + 1 < len) printf(" ");
        }
        printf("}\n");
        return;
    }
    backtrack(idx + 1, len);          // exclude
    cur[len] = a[idx];                // include
    backtrack(idx + 1, len + 1);
}

int main() {
    scanf("%d", &n);
    for (int i = 0; i < n; i++) scanf("%d", &a[i]);
    backtrack(0, 0);
    return 0;
}
```

### Java
```java
import java.util.*;
```

```
class Main {
    static int n;
    static int[] a;

    static void backtrack(int idx, List<Integer> cur) {
        if (idx == n) {
            System.out.println(cur);
            return;
        }
        backtrack(idx + 1, cur);
        cur.add(a[idx]);
        backtrack(idx + 1, cur);
        cur.remove(cur.size() - 1);
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        a = new int[n];
        for (int i = 0; i < n; i++) a[i] = sc.nextInt();
        backtrack(0, new ArrayList<>());
    }
}
```

### Python 3

```python
n = int(input())
arr = list(map(int, input().split()))
cur = []

def backtrack(idx):
    if idx == n:
        print(cur[:])
        return
    backtrack(idx + 1)
    cur.append(arr[idx])
    backtrack(idx + 1)
    cur.pop()
```

```
backtrack(0)
```

***

## 3) N-Queens for N = 4 (one solution print)

C
```c
#include <stdio.h>

#define N 4
int board[N][N];

int safe(int row, int col) {
    for (int i = 0; i < col; i++)
        if (board[row][i]) return 0;
    for (int i=row, j=col; i>=0 && j>=0; i--, j--)
        if (board[i][j]) return 0;
    for (int i=row, j=col; i<N && j>=0; i++, j--)
        if (board[i][j]) return 0;
    return 1;
}

int solve(int col) {
    if (col == N) return 1;
    for (int i = 0; i < N; i++) {
        if (safe(i, col)) {
            board[i][col] = 1;
            if (solve(col + 1)) return 1;
            board[i][col] = 0;
        }
    }
    return 0;
}

int main() {
    if (!solve(0)) {
        printf("No solution\n");
```

~~~~
        return 0;
    }
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++)
            printf(board[i][j] ? "Q " : ". ");
        printf("\n");
    }
    return 0;
}
```


### Java
```java
import java.util.*;

class Main {
    static final int N = 4;
    static int[][] board = new int[N][N];

    static boolean safe(int row, int col) {
        for (int i = 0; i < col; i++)
            if (board[row][i] == 1) return false;
        for (int i=row, j=col; i>=0 && j>=0; i--, j--)
            if (board[i][j] == 1) return false;
        for (int i=row, j=col; i<N && j>=0; i++, j--)
            if (board[i][j] == 1) return false;
        return true;
    }

    static boolean solve(int col) {
        if (col == N) return true;
        for (int i = 0; i < N; i++) {
            if (safe(i, col)) {
                board[i][col] = 1;
                if (solve(col + 1)) return true;
                board[i][col] = 0;
            }
        }
        return false;
    }
~~~~

```
    public static void main(String[] args) {
        if (!solve(0)) {
            System.out.println("No solution");
            return;
        }
        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N; j++)
                System.out.print(board[i][j] == 1 ? "Q " : ". ");
            System.out.println();
        }
    }
}
```

### Python 3

```python
N = 4
board = [[0] * N for _ in range(N)]

def safe(row, col):
    for i in range(col):
        if board[row][i]:
            return False
    i, j = row, col
    while i >= 0 and j >= 0:
        if board[i][j]:
            return False
        i -= 1; j -= 1
    i, j = row, col
    while i < N and j >= 0:
        if board[i][j]:
            return False
        i += 1; j -= 1
    return True

def solve(col):
    if col == N:
        return True
    for r in range(N):
        if safe(r, col):
            board[r][col] = 1
            if solve(col + 1):
                return True
            board[r][col] = 0
    return False

if solve(0):
    for row in board:
        print(*("Q" if x else "." for x in row))
else:
    print("No solution")
```

***

## 4) Number of Islands (DFS on 0/1 grid)

Input: `n m` followed by `n` lines of 0/1 (or ‘0’/‘1’ characters). [4][5]

C
```c
#include <stdio.h>

int n, m;
char g[105][105];
int vis[105][105];

int dx[4] = {1,-1,0,0};
int dy[4] = {0,0,1,-1};

void dfs(int x, int y) {
    vis[x][y] = 1;
    for (int k = 0; k < 4; k++) {
        int nx = x + dx[k], ny = y + dy[k];
        if (nx>=0 && nx<n && ny>=0 && ny<m &&
            !vis[nx][ny] && g[nx][ny]=='1')
            dfs(nx, ny);
    }
}
```

```
int main() {
    scanf("%d %d", &n, &m);
    for (int i = 0; i < n; i++)
        scanf("%s", g[i]);

    int cnt = 0;
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            if (!vis[i][j] && g[i][j]=='1') {
                cnt++;
                dfs(i, j);
            }
    printf("%d\n", cnt);
    return 0;
}
```

### Java

```java
import java.util.*;

class Main {
    static int n, m;
    static char[][] g;
    static boolean[][] vis;
    static int[] dx = {1,-1,0,0};
    static int[] dy = {0,0,1,-1};

    static void dfs(int x, int y) {
        vis[x][y] = true;
        for (int k = 0; k < 4; k++) {
            int nx = x + dx[k], ny = y + dy[k];
            if (nx>=0 && nx<n && ny>=0 && ny<m &&
                !vis[nx][ny] && g[nx][ny]=='1')
                dfs(nx, ny);
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        m = sc.nextInt();
        g = new char[n][m];
        vis = new boolean[n][m];
        for (int i = 0; i < n; i++) {
            String row = sc.next();
            for (int j = 0; j < m; j++)
                g[i][j] = row.charAt(j);
        }
        int cnt = 0;
        for (int i = 0; i < n; i++)
            for (int j = 0; j < m; j++)
                if (!vis[i][j] && g[i][j]=='1') {
                    cnt++;
                    dfs(i, j);
                }
        System.out.println(cnt);
    }
}
```

### Python 3

```python
n, m = map(int, input().split())
grid = [list(input().strip()) for _ in range(n)]
vis = [[False] * m for _ in range(n)]

def dfs(x, y):
    vis[x][y] = True
    for dx, dy in [(1,0),(-1,0),(0,1),(0,-1)]:
        nx, ny = x + dx, y + dy
        if 0 <= nx < n and 0 <= ny < m:
            if not vis[nx][ny] and grid[nx][ny] == '1':
                dfs(nx, ny)

cnt = 0
for i in range(n):
    for j in range(m):
        if grid[i][j] == '1' and not vis[i][j]:
            cnt += 1
            dfs(i, j)

print(cnt)
```

****
****