

## ✳ SET – 3 (NEW CODING CONTEST QUESTIONS)

## ✓ EASY – Problem E1: Count Vowels in a String

 Input

hello world

 Output

3

 Logic Explanation

string లోని ప్రతి character ని scan చేయాలి

vowels: a, e, i, o, u (uppercase కూడా)

count చేసి ప్రింట్ చేయాలి

 Pseudocode

```
read S
count = 0
for each char c in S:
    if c in [a,e,i,o,u,A,E,I,O,U]:
        count++
print count
```

---


## ✓ EASY – Problem E2: Count Even Numbers in Array

 Input6  
1 2 3 4 5 6 Output

3


 Logic Explanation

array లోని ప్రతి ఎలిమెంట్ even అయితే count చేయాలి


 Pseudocode

```
read N
read array A
count = 0
for x in A:
    if x % 2 == 0:
        count++
print count
```

---

 MODERATE – Problem M1: Reverse Words in a Sentence Input

welcome to coding contest

 Output

emotion tsetnoc gnidoc ot emoclew

(Note: ప్రతి word reverse చేయాలి కానీ word order మారకూడదు)

 Logic Explanation

sentence ని split చేయాలి

ప్రతి word ని individually reverse చేయాలి

spacing same గా ఉంచాలి

 Pseudocode

```
read S
split words by space
result = ""
for each word in words:
    reversed_word = reverse(word)
    append reversed_word to result
print result
```

---

 MODERATE – Problem M2: Frequency of Each Element (Sorted Output) Input

```
7
2 3 2 5 3 2 5
```

 Output

```
2 3
3 2
5 2
```

 Logic Explanation

array లో repeats ఉంటాయి

element → frequency ఉపయోగించి map లో save చేయాలి

ascending order లో element ని print చేయాలి

 Pseudocode

```
read N
read array A
freq_map = empty map
```

```
for x in A:
    freq_map[x]++

sort keys of freq_map

for key in sorted keys:
    print key, freq_map[key]
```

---

 HARD – Problem H1: Longest Increasing Subarray Input

9  
1 2 2 3 4 1 2 3 4

 Output

4

(Example: longest increasing contiguous sequence → 1,2,3,4, length = 4)

 Logic Explanation

contiguous increasing sequence find చేయాలి

$arr[i] > arr[i-1]$  అయితే  $length++$

else → new sequence ప్రారంభించాలి

max length store చేయాలి

 Pseudocode


```
read N
read array A
maxLen = 1
```

```
currLen = 1
```

```
for i in 1 to N-1:  
    if A[i] > A[i-1]:  
        currLen++  
        maxLen = max(maxLen, currLen)  
    else:  
        currLen = 1
```

```
print maxLen
```

---

 HARD – Problem H2: Number of Subarrays Whose Sum Equals K

 Input

```
5 5  
1 2 3 2 1
```

 Output

```
2
```

(valid subarrays: [2,3], [3,2])

 Logic Explanation

subarray sum = K

brute force  $O(N^2)$  slow అవుతుంది

prefix sum + hashmap technique  $\rightarrow O(N)$

 Pseudocode

```
read N, K  
read array A
```

```
map = {0:1}    # prefix-sum=0 count = 1
```

```
prefix = 0
count = 0
```

```
for i in 0..N-1:
    prefix += A[i]
    if (prefix - K) in map:
        count += map[prefix - K]
    map[prefix]++
print count
```

.....

.....

.....

### ✓ EASY – Problem E1: Count Vowels in a String

📥 Input

hello world

📤 Output

3

🧠 Logic

ప్రతి character vowel (a,e,i,o,u — uppercase కూడా) అయితే count++

📝 Pseudocode

```
read S
count = 0
for c in S:
    if c is vowel:
        count++
print count
```

 Java Code

```
import java.io.*;

public class CountVowels {
    public static void main(String[] args) throws Exception {
        BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
        String s = br.readLine();
        int count = 0;
        s = s.toLowerCase();
        for (char c : s.toCharArray()) {
            if (c=='a' || c=='e' || c=='i' || c=='o' || c=='u')
                count++;
        }
        System.out.println(count);
    }
}
```

---

 EASY – Problem E2: Count Even Numbers in Array Input

6  
1 2 3 4 5 6

 Output

3


 Logic

array  $\mathcal{S}$  even elements count++

 Pseudocode

read N  
read array A

```
count = 0
for x in A:
    if x % 2 == 0:
        count++
print count
```

 Java Code

```
import java.io.*;

public class CountEvenNumbers {
    public static void main(String[] args) throws Exception {
        BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
        int n = Integer.parseInt(br.readLine());
        String[] parts = br.readLine().split(" ");
        int count = 0;
        for (String p : parts) {
            int x = Integer.parseInt(p);
            if (x % 2 == 0) count++;
        }
        System.out.println(count);
    }
}
```

---

◆ MODERATE – Problem M1: Reverse Words in a Sentence

 Input

welcome to coding contest

 Output

emoclew ot gnidoc tsetnoc

 Logic

sentence ని split చేయాలి

ప్రతి word ని reverse చేసి join చేయాలి

 Pseudocode

```
read S
words = split(S)
result = ""
for w in words:
    reverse w
    add to result
print result
```

 Java Code

```
import java.io.*;

public class ReverseWords {
    public static void main(String[] args) throws Exception {
        BufferedReader br = new BufferedReader(new
        InputStreamReader(System.in));
        String s = br.readLine();

        String[] words = s.split(" ");
        StringBuilder result = new StringBuilder();

        for (String w : words) {
            StringBuilder sb = new StringBuilder(w);
            result.append(sb.reverse().toString()).append(" ");
        }

        System.out.println(result.toString().trim());
    }
}
```

---

◆ MODERATE – Problem M2: Frequency of Each Element (Sorted Output)

 Input

7  
2 3 2 5 3 2 5


 Output

2 3  
3 2  
5 2


 Logic

map ద్వారా frequency count

keys ని ascending విధంగా sort చేసి print చేయాలి

 Pseudocode

```
read N
read array A
freq_map = {}
for x in A:
    freq_map[x]++
sort keys
for key in sorted keys:
    print key, freq[key]
```

 Java Code

```
import java.io.*;
import java.util.*;

public class FrequencySorted {
    public static void main(String[] args) throws Exception {
        BufferedReader br = new BufferedReader(new
        InputStreamReader(System.in));
        int n = Integer.parseInt(br.readLine());
        String[] parts = br.readLine().split(" ");

        TreeMap<Integer, Integer> map = new TreeMap<>();

        for (String p : parts) {
```

```
int x = Integer.parseInt(p);
map.put(x, map.getOrDefault(x, 0) + 1);
}

for (int key : map.keySet()) {
    System.out.println(key + " " + map.get(key));
}
}
```

---

 HARD – Problem H1: Longest Increasing Subarray Input

9  
1 2 2 3 4 1 2 3 4

 Output


4

 Logic

If  $A[i] > A[i-1] \rightarrow$  current length++

Else  $\rightarrow$  reset to 1

max length track చేయాలి

 Pseudocode

```
read N
read array A
maxLen = 1
curr = 1
for i = 1 to N-1:
    if  $A[i] > A[i-1]$ :
        curr++
```

```
        maxLen = max(maxLen, curr)
    else:
        curr = 1
print maxLen
```

 Java Code

```
import java.io.*;


public class LongestIncreasingSubarray {
    public static void main(String[] args) throws Exception {
        BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
        int n = Integer.parseInt(br.readLine());
        String[] parts = br.readLine().split(" ");

        int[] arr = new int[n];
        for (int i = 0; i < n; i++) arr[i] = Integer.parseInt(parts[i]);

        int maxLen = 1, curr = 1;

        for (int i = 1; i < n; i++) {
            if (arr[i] > arr[i - 1]) {
                curr++;
                maxLen = Math.max(maxLen, curr);
            } else {
                curr = 1;
            }
        }
        System.out.println(maxLen);
    }
}
```

---

 HARD – Problem H2: Subarrays Whose Sum Equals K

 Input

```
5 5
1 2 3 2 1
```


 Output

2

 Logic

prefix sum + hashmap  $\rightarrow O(N)$

prefix-k already  $\rightarrow$  subarray exists

 Pseudocode

```

read N, K
read array A
map = {0:1}
prefix = 0
count = 0
for each x in A:
    prefix += x
    if prefix-K in map:
        count += map[prefix-K]
    map[prefix]++
print count

```

 Java Code

```

import java.io.*;
import java.util.*;

public class SubarraySumEqualsK {
    public static void main(String[] args) throws Exception {
        BufferedReader br = new BufferedReader(new
        InputStreamReader(System.in));
        String[] nk = br.readLine().split(" ");
        int n = Integer.parseInt(nk[0]);
        long k = Long.parseLong(nk[1]);

        String[] parts = br.readLine().split(" ");
        long[] arr = new long[n];
        for (int i = 0; i < n; i++) arr[i] = Long.parseLong(parts[i]);
    }
}

```

```
HashMap<Long, Long> map = new HashMap<>();
map.put(0L, 1L);

long prefix = 0, count = 0;

for (long x : arr) {
    prefix += x;
    if (map.containsKey(prefix - k))
        count += map.get(prefix - k);
    map.put(prefix, map.getDefault(prefix, 0L) + 1);
}

System.out.println(count);
}
```