MidSem-A Lab Exam

Problem 1: Pingala Storing Student Name

Write a **C** program that allows Pingala server to store a fixed number of names in an array. The program should let Pingala server perform the following operations:

- Add a name.
- Display all names.
- Exit the program.

Constraints

- The maximum number of names is fixed at **10**.
- Each name can be up to **30** characters long.
- Use dynamic memory allocation for storing names.

Input Format

Commands, where:

- **1** indicates adding a name.
- **2** indicates displaying all names.
- **0** indicates exiting the program.

and Names.

Sample Input 1

```
1
Alice
1
Bob
2
0
```

Sample Output 1

```
Names:
Alice
Bob
```

Sample Input 2

```
1
Alice
1
Bob
1
Charlie
1
David
1
Eve
1
Frank
1
Grace
1
Hannah
1
Ivy
1
Jack
1
Kate
2
0
```

Sample Output 2

```
Cannot add more names. Maximum limit reached.
Names:
Alice
Bob
Charlie
David
Eve
Frank
Grace
Hannah
Ivy
Jack
```

Additional Notes:

Ensure to handle memory allocation properly and free any allocated memory before the program exits to avoid memory leak errors.

If you encounter a **Time Limit Exceeded (TLE)** error, check the given conditions and feel free to modify approach.

Problem 2: Automated Robot Sorting

A little robot boy , Valera, studies an algorithm of sorting an integer array. After studying the theory, he wrote a program that sorts an array of `n` integers `a1, a2, ..., an` in non-decreasing order. The pseudocode of the program, written by Valera, is given below:

```
loop integer variable i from 1 to n - 1
    loop integer variable j from i to n - 1
        if (aj > aj + 1), then swap the values of elements aj and aj + 1
```

But Valera could have made a mistake because he hasn't fully learned the sorting algorithm. If Valera made a mistake in his program, you need to give a counter-example that makes his program work improperly (i.e., the example that makes the program sort the array incorrectly). If such an example for the given value of `n` doesn't exist, print `-1`.

Input format:

- The input consists of a single integer `n (1 ≤ n ≤ 50)` — the size of the array.

Output format:

- Print `n` space-separated integers `a1, a2, ..., an (1 ≤ ai ≤ 100)` — a counter-example for which Valera's algorithm will not work correctly. If a counter-example that meets the conditions is impossible, print `-1`.
- If there are several counter-examples, you may print any of them.

Examples:

Sample Input 1:

```
1
```

Sample Output 1:

```
-1
```

Explanation:

When `n = 1`, there's only one element in the array, so sorting is trivial, and no counter-example exists.

Sample Input 2:

```
3
```

Sample Output 2:

```
3 2 1
```

Explanation:

For `n = 3`, an input array such as `[3, 2, 1]` could be a counter-example because Valera's sorting algorithm may not handle certain conditions correctly.

Problem 3: Alien Buffet Feast

After successfully completing a space mission, astronaut Yash is invited to an alien buffet where an assortment of alien dishes is arranged in a straight line. Each dish gives Yash a different level of satisfaction, but the catch is that Yash can only take consecutive dishes. Your task is to help Yash maximize the total satisfaction he gains by selecting a consecutive subarray of dishes.

Input:

- The first line contains an integer `n`, the number of dishes in the buffet.
- The second line contains `n` integers, where each integer represents the happiness value that each dish gives Yash.

Output:

- Output a single integer, the maximum happiness Yash can achieve by selecting consecutive dishes.

Constraints:

- `1 ≤ n ≤ 10^5`
- `-1000 ≤ a[i] ≤ 1000`, where `a[i]` is the happiness value of the `i-th` dish

Example:

Input 1:

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

Output 1:

```
3
```

Input 2:

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

Output 2:

```
7
```

In the first example, Yash selects the dish with happiness value `3`, which maximizes his happiness. In the second example, Yash selects the consecutive dishes with happiness values `1 2 3 -2 3` which gives a total of `7` happiness points.

Submission Guidelines

- Do not rename any files given in the handout.
- Only write the code in the specified C files in the respective directories.

- Good luck!