```
------------------------------ Алгоритмы с возвратом ----------------------

//51. Удалить из заданного числового массива наименьшее число
// элементов так, чтобы оставшиеся составили возрастающую последовательность
// (не меняя их порядок следования).

//Mia Muzamil Ahmad
//Group 91.

#include <iostream>
const int n = 7;
int arr[n] = { 1,100,2,3,1,4,5 };

int current[n];
int best[n];
int best_length = 0;

void Task(int index, int current_length) {
    if (index < n)
    {
        // Включаем текущий элемент, если он сохраняет порядок возрастания

        if (current_length == 0 || arr[index] > current[current_length - 1]) {
            current[current_length] = arr[index];
            Task(index + 1, current_length + 1);
        }

        Task(index + 1, current_length);
    }

    // Обновляем наилучшую подпоследовательность при достижении конца массива
    if (index == n) {
        bool isIncreasing = true;
        for (int i{1}; i < current_length; i++) {
            if (current[i] <= current[i - 1]) {
                isIncreasing = false;
            }
        }
        // Если подпосле-ность возрас-ая и длиннее найденной ранее
        if (isIncreasing && current_length > best_length) {
            best_length = current_length;
            // Сохраняем новую наилучшую подпоследовательность
            for (int i = 0; i < current_length; i++) {
                best[i] = current[i];
            }
        }
    }
}

int main()
{
    Task(0, 0);

    std::cout << "\nLongest Increasing Subsequence: ";
    for (int i = 0; i < best_length; i++) {
        std::cout << best[i] << " ";
    }
    std::cout << "\n";
```

```
    return 0;
}
```