**Ramdeobaba University, Nagpur**

**Department of Computer Science and Engineering**

**Session: 2025-26**

**DAA LAB III Semester**

**Name: Vardhan Ingole**

**Section: A4**

**Batch: B3**

**Roll No: 43**

**Practical - 4**

**Aim:** Implement maximum sum of subarray for the given scenario of resource allocation using

the divide and conquer approach.

**Problem Statement:**

A project requires allocating resources to various tasks over a period of time. Each task requires a certain amount of resources, and you want to maximize the overall efficiency of resource usage. You're given an array of resources where resources[i] represents the amount of resources required for the task. Your goal is to find the contiguous subarray of tasks that maximizes the total resources utilized without exceeding a given resource constraint. Handle cases where the total resources exceed the constraint by adjusting the subarray window

accordingly. Your implementation should handle various cases, including scenarios where there's no feasible subarray given the constraint and scenarios where multiple subarrays yield the same maximum resource utilization.

**Code:**

#include <stdio.h>

int SumSubarray(int \*nums, int n, int con)

{

for (int i = 0; i < n; i++)

{

int sum = 0;

for (int j = i; j < n; j++)

{

sum += nums[j];

if (sum == con)

{

printf("Subarray found: [");

for (int k = i; k <= j; k++)

{

printf("%d", nums[k]);

if (k < j)

printf(", ");

}

printf("]\n");

return 1;

}

else if (sum == con - 1)

{

printf("The Next Best Subarray found: [");

for (int k = i; k <= j; k++)

{

printf("%d", nums[k]);

if (k < j)

printf(", ");

}

printf("] For Constraint = %d\n", con - 1);

return 1;

}

}

}

return 0;

}

int main()

{

int n;

printf("Enter number of elements: ");

scanf("%d", &n);

if (n >= 10000)

{

printf("I cannot Give input for these many elements");

return 0;

}

int arr[n];

for (int i = 0; i < n; i++)

{

printf("Enter %d elements: ", i);

scanf("%d", &arr[i]);

}

int con;

printf("Enter constraint: ");

scanf("%d", &con);

if (!SumSubarray(arr, n, con))

{

printf("No subarray with sum = %d found\n", con);

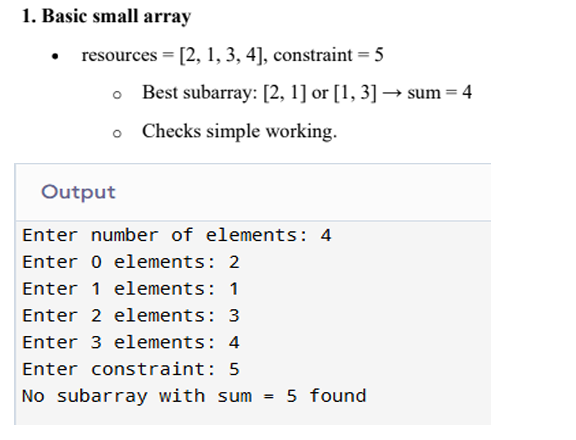
}

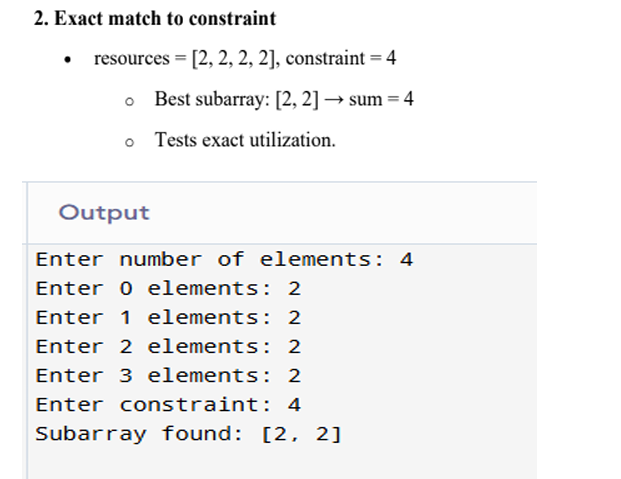
return 0;

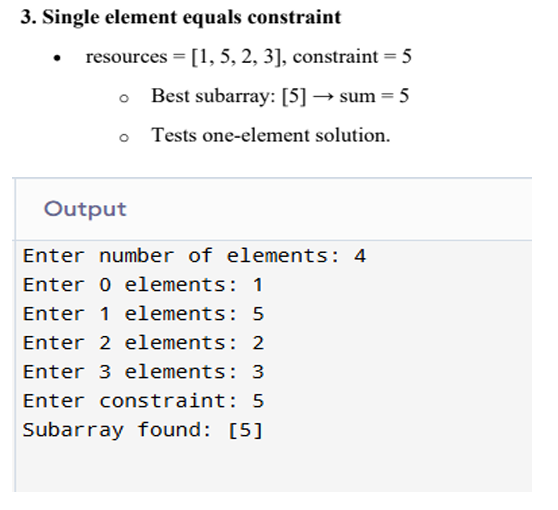
}

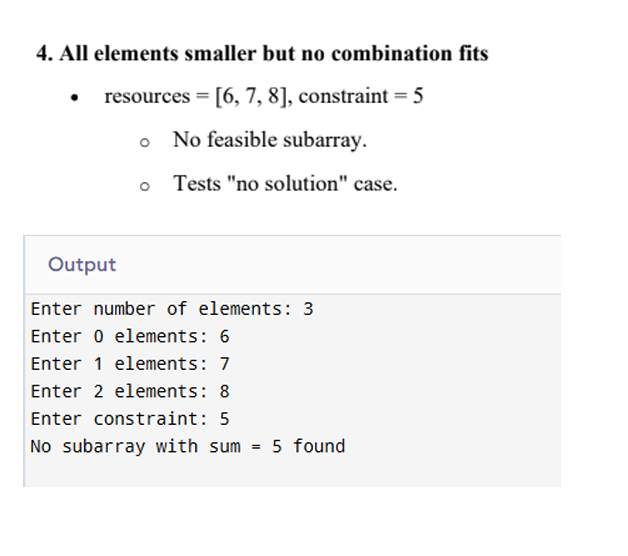
**Output:**

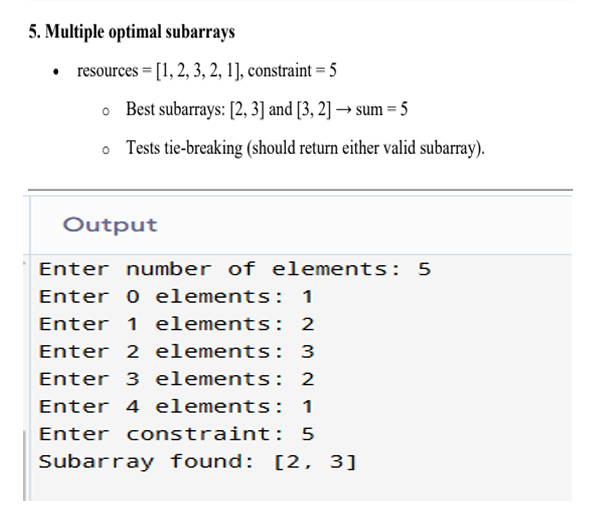
**Test Cases:**

****

****

****

****

****

