1. **Sum of subsets using backtracking**

**Code:**

#include <stdio.h>

#include <stdbool.h>

#define MAX\_SIZE 100

// Function to print the subset

void printSubset(int subset[], int size) {

printf("{ ");

for (int i = 0; i < size; i++) {

printf("%d ", subset[i]);

}

printf("}\n");

}

// Function to find all subsets with the given sum using backtracking

void findSubsets(int set[], int subset[], int n, int subsetSize, int sum, int targetSum, int k) {

if (sum == targetSum) {

printSubset(subset, subsetSize);

findSubsets(set, subset, n, subsetSize - 1, sum - set[k], targetSum, k + 1);

return;

} else {

for (int i = k; i < n; i++) {

subset[subsetSize] = set[i];

findSubsets(set, subset, n, subsetSize + 1, sum + set[i], targetSum, i + 1);

}

}

}

// Function to initialize the variables and call findSubsets function

void sumOfSubsets(int set[], int n, int targetSum) {

int subset[MAX\_SIZE] = {0};

findSubsets(set, subset, n, 0, 0, targetSum, 0);

}

// Main function

int main() {

int set[MAX\_SIZE], n, targetSum;

printf("Enter the number of elements in the set: ");

scanf("%d", &n);

printf("Enter the elements of the set: ");

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

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

}

printf("Enter the target sum: ");

scanf("%d", &targetSum);

printf("Subsets with sum %d are:\n", targetSum);

sumOfSubsets(set, n, targetSum);

return 0;

}

**Output:**

Enter the number of elements in the set: 7

Enter the elements of the set: 7 14 21 28 35 42 49

Enter the target sum: 56

Subsets with sum 56 are:

{ 7 14 35 }

{ 7 21 28 }

{ 7 49 }

{ 14 42 }

{ 21 35 }

--------------------------------

Process exited after 51.21 seconds with return value 0

Press any key to continue . . .

