



Worksheet- 6

Student Name: Harsh

UID: 24MCA20045

Branch: MCA

Section/Group: 1(A)

Semester: 2

Date of Performance: 04-04-2025

Subject Name: Design and Analysis of Algorithms Lab **Subject Code:** 24CAP-612

Q. a. AIM:

Find a subset of a given set $S = \{s_1, s_2, \dots, s_n\}$ of n positive integers whose sum is equal to a given positive integer d . For example, if $S = \{1, 2, 5, 6, 8\}$ and $d = 9$ there are two solutions $\{1, 2, 6\}$ and $\{1, 8\}$. A suitable message is to be displayed if the given problem instance doesn't have a solution.

1. Task to be done

- **Understand the Problem Requirements:**
 - Analyse the subset sum problem, which involves finding subsets of a given set that sum up to a target integer.
 - Determine and handle cases where no subset matches the desired sum by displaying an appropriate message.
- **Implement the Solution in Java:**
 - Create a Java program that recursively finds all subsets of a set SSS that sum to a given integer d .
 - Use recursion to explore both choices (including and excluding elements) for each element in the set.
 - Print each subset that sums to the target value.
 - Include base cases to handle when the subset sum matches the target, the target becomes negative, or the subset is empty.

2. Code for experiment/practical:

```
def find_subsets_with_sum(S, d):  
    result = []
```

```
def backtrack(start, path, current_sum):  
    if current_sum == d:  
        result.append(path[:])  
    return
```

```
    if current_sum > d:
```

```
return
for i in range(start, len(S)):
    path.append(S[i])
    backtrack(i + 1, path, current_sum + S[i])
    path.pop()
```

```
backtrack(0, [], 0)
return result
```

```
# Example usage:
S = [1, 2, 5, 6, 8]
d = 9
subsets = find_subsets_with_sum(S, d)
```

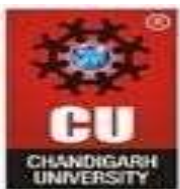
```
if subsets:
    print(f'Subsets of {S} with sum {d}:')
    for subset in subsets:
        print(subset)
    else:
        print(f'No subset of {S} adds up to {d}.')
```

3. Result/Output/Writing Summary:

```
[Running] python -u "c:\Users\Deepa\OneDrive\Desktop\New folder\worksheet 6.py"
Subsets of [1, 2, 5, 6, 8] with sum 9:
[1, 2, 6]
[1, 8]
```

4. Learning outcomes (What I have learnt):

- Understanding Recursion:
 - Gain a deeper understanding of recursive programming techniques, especially for problems that involve exploring multiple choices and paths.
 - Learn to apply recursion for subset generation and backtracking, which is useful in many algorithmic problems.
- Subset Sum Problem Solving:
 - Develop problem-solving skills for the subset sum problem, a classic problem in computer science, and recognize its applications in other fields like combinatorial optimization.
 - Learn to implement and understand the logic behind exploring all subsets of a set and identifying specific conditions for successful subset matches.



Evaluation Grid:

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Demonstration and Performance (Pre Lab Quiz)		5
2.	Worksheet		10
3.	Post Lab Quiz		5