

**Programiz**
Python Online
Compiler

 **LEARN PYTHON**
[Learn More](#)

LOOKING TO LEARN PROGRAMMING?
Start your programming journey with Programiz **AT NO COST.**

**Programiz
PRO**

 **main.py**    **Run** **Output** 

```
1 N = 4
2 board = [
3     [1, 0, 0, 0],
4     [0, 0, 1, 0],
5     [0, 1, 0, 0],
6     [0, 0, 0, 1],
7 ]
8
9 for row in board:
10     print(" ".join("Q" if x else "." for x in
11                    row))
```

```
Q . . .
. . Q .
. Q . .
. . . Q

=== Code Execution Successful ===
```

[JS](#)
[Go](#)
[PHP](#)



Programiz
Python Online
Compiler

RADO
SWITZERLAND



CAPTAIN COOK &
TRUE SQUARE

DISCOVER



main.py



Run

Output

```
1 N = 5
2 obstacles = [(1, 2), (2, 3)]
3 board = [-1] * N
4 def is_safe(row, col):
5     if (row, col) in obstacles:
6         return False
7     for prev_row in range(row):
8         prev_col = board[prev_row]
9         if prev_col == col or \
10            prev_col - col == prev_row - row or \
11            prev_col - col == row - prev_row:
12             return False
13     return True
14 def solve(row):
15     if row == N:
16         print(board)
17         return
18     for col in range(N):
19         if is_safe(row, col):
20             board[row] = col
21             solve(row + 1)
22             board[row] = -1
23 solve(0)
24
```

```
[0, 3, 1, 4, 2]
[1, 3, 0, 2, 4]
[1, 4, 2, 0, 3]
[2, 4, 1, 3, 0]
[3, 0, 2, 4, 1]
[3, 1, 4, 2, 0]
```

=== Code Execution Successful ===





Programiz
Python Online
Compiler

E-Book on Lean Six Sigma / Scrum Agile.
Tutorial with CASE study on Scrum Tool

LEARN MORE >

Programiz
PRO >

main.py

Run

Output

Clear

```
1 board = [[["5","3",".", ".", ".", "7",".", ".", ".", "."], ['5', '3', '4', '6', '7', '8', '9', '1', '2']
2           ["6",".", ".", ".", "1","9","5",".", ".", ".", "."], ['6', '7', '2', '1', '9', '5', '3', '4', '8']
3           [".","9","8",".", ".", ".", ".", "6",".", ".", "."], ['1', '9', '8', '3', '4', '2', '5', '6', '7']
4           ["8",".", ".", ".", "6",".", ".", ".", ".", "3"], ['8', '5', '9', '7', '6', '1', '4', '2', '3']
5           ["4",".", ".", ".", "8",".", "3",".", ".", ".", "1"], ['4', '2', '6', '8', '5', '3', '7', '9', '1']
6           ["7",".", ".", ".", "2",".", ".", ".", ".", "6"], ['7', '1', '3', '9', '2', '4', '8', '5', '6']
7           [".","6",".", ".", ".", ".", ".", "2","8",".", "."], ['9', '6', '1', '5', '3', '7', '2', '8', '4']
8           [".",".", ".", ".", "4","1","9",".", ".", ".", "5"], ['2', '8', '7', '4', '1', '9', '6', '3', '5']
9           [".",".", ".", ".", "8",".", ".", ".", "7","9"]], ['3', '4', '5', '2', '8', '6', '1', '7', '9']
10
11 def is_valid(board, row, col, num):
12     for i in range(9):
13         if board[row][i] == num or
14             board[i][col] == num or board[row
15             //3*3 + i//3][col//3*3 + i%3] ==
16             num:
17             return False
18     return True
19
20 def solve_sudoku(board):
21     for row in range(9):
22         for col in range(9):
23             if board[row][col] == '.':
24                 for num in map(str, range(1, 10
25                 )):
26                     if is_valid(board, row, col
27                     , num):
28                         board[row][col] = num
29                         if solve_sudoku(board):
30                             return True
31                         board[row][col] = '.'
32                     return False
33     return True
34
35 solve_sudoku(board)
36 for row in board:
37     print(row)
```

=== Code Execution Successful ===





main.py



Run

Output



```
1 nums = [1, 1, 1, 1, 1]
2 target = 3
3
4 memo = {}
5
6 def dfs(i, current_sum):
7     if i == len(nums):
8         return 1 if current_sum == target else 0
9
10    if (i, current_sum) in memo:
11        return memo[(i, current_sum)]
12
13    add = dfs(i + 1, current_sum + nums[i])
14    subtract = dfs(i + 1, current_sum - nums[i])
15
16    memo[(i, current_sum)] = add + subtract
17    return memo[(i, current_sum)]
18
19 result = dfs(0, 0)
20 print(result) # Output will be 5
```

5
=== Code Execution



Programiz

Python Online
CompilerProgram
PRO >

main.py



Run

Output

Clear

```
1 arr = [3, 1, 2, 4]
2 mod = 10**9 + 7
3 n = len(arr)
4 stack = []
5 prev = [0] * n
6 next = [0] * n
7 for i in range(n):
8     while stack and arr[stack[-1]] >= arr[i]:
9         stack.pop()
10    prev[i] = i - stack[-1] if stack else i + 1
11    stack.append(i)
12    stack = []
13    for i in range(n-1, -1, -1):
14        while stack and arr[stack[-1]] > arr[i]:
15            stack.pop()
16        next[i] = stack[-1] - i if stack else n - i
17        stack.append(i)
18    result = sum(arr[i] * prev[i] * next[i] for i
19                in range(n)) % mod
19    print(result)
20
```

17

=== Code Execution Successful ===





Programiz

Online
Compiler*T&C apply.
Image simulated.

Galaxy A15 5G

Awesome
Freedom
Offer Starting at
₹ 17999
₹ 16499*

SAMSUNG

Buy now

main.py



Run

Output

```
1 def combinationSum(candidates, target):
2     def backtrack(remaining, combination, start):
3         if remaining == 0:
4             result.append(list(combination))
5             return
6         elif remaining < 0:
7             return
8         for i in range(start, len(candidates)):
9             combination.append(candidates[i])
10            backtrack(remaining - candidates[i], combination, i)
11            combination.pop()
12    result = []
13    candidates.sort()
14    backtrack(target, [], 0)
15    return result
16 candidates = [2, 3, 6, 7]
17 target = 7
18 print(combinationSum(candidates, target))
19
```


[[2, 2, 3], [7]]


=== Code Execution Successful ===



5:46

Vol
LTE 34%Online ...
gramiz.com








**Programiz**
Python Online
Compiler



Financial freedom awaits
SPONSORED BY VELTRIX INDIA

LEARN MORE

Programiz
PRO >

 main.py      

```
1 candidates = [10, 1, 2, 7, 6, 1, 5]
2 target = 8
3 candidates.sort()
4 res = []
5
6 def backtrack(start, path, target):
7     if target == 0:
8         res.append(path)
9         return
10    for i in range(start, len(candidates)):
11        if i > start and candidates[i] ==
           candidates[i-1]:
12            continue
13        if candidates[i] > target:
14            break
15        backtrack(i + 1, path + [candidates[i]]
           , target - candidates[i])
16
17 backtrack(0, [], target)
18 print(res)
19
```

[[1, 1, 6], [1, 2, 5], [1, 7], [2, 6]]

=== Code Execution Successful ===



5:49



VoLTE LTE 33%

Online ...
gramiz.com

Programiz
Python Online
Compiler

MilesWeb
50% OFF*

DEDICATED
SERVER

DATA CENTER
IN INDIA

Grab This Deal →

Programiz
PRO >

main.py

Run

Output

Clear

```
1 import itertools
2 nums = [1, 2, 3]
3 result = list(itertools.permutations(nums))
4 print(result)
5
```

```
[(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1)]
=== Code Execution Successful ===
```

Python

JS

GO

PHP





Programiz
Python Online
Compiler

Build, Shape and Unleash the power of data
SPONSORED BY ZOOMINFO

APPLY

Programiz
PRO >

main.py **Run** Output

```
1 from itertools import permutations [(1, 2, 1), (2, 1, 1), (1, 1, 2)]
2 nums = [1, 1, 2]
3 unique_permutations = list(set(permutations(nums) == Code Execution Successful ==
   )))
4 print(unique_permutations)
5
```

