



OnlineGDB beta

online compiler and debugger for c/c++

Welcome, **Maithreyee Jagadeesan** ▲

Create New Project

My Projects

Classroom **new**

Learn Programming

Programming Questions

Jobs **new**

Upgrade

Logout ▼

Learn Python with
KodeKloud

[About](#) • [FAQ](#) • [Blog](#) • [Terms of Use](#) • [Contact Us](#) • [GDB](#)

[Tutorial](#) • [Credits](#) • [Privacy](#)

© 2016 - 2024 GDB Online

Run Debug Stop Share Save {} Beautify

Language Python 3

main.py

```
1 def is_valid(board, row, col, num):
2     for x in range(9):
3         if board[row][x] == num:
4             return False
5
6     for x in range(9):
7         if board[x][col] == num:
8             return False
9
10    start_row, start_col = 3 * (row // 3), 3 * (col // 3)
11    for i in range(3):
12        for j in range(3):
13            if board[i + start_row][j + start_col] == num:
14                return False
15
16    return True
17
18 def solve_sudoku(board):
19     empty = find_empty_location(board)
20     if not empty:
21         return True
22
23     row, col = empty
24
25     for num in range(1, 10):
26         if is_valid(board, row, col, num):
27             board[row][col] = num
28
29         if solve_sudoku(board):
```

input

Sudoku solved successfully!

```
5 3 4 6 7 8 9 1 2
6 7 2 1 9 5 3 4 8
1 9 8 3 4 2 5 6 7
8 5 9 7 6 1 4 2 3
4 2 6 8 5 3 7 9 1
7 1 3 9 2 4 8 5 6
```



OnlineGDB beta

online compiler and debugger for c/c++

Welcome, **Maithreyee Jagadeesan** 📌

Create New Project

My Projects

Classroom new

Learn Programming

Programming Questions

Jobs now

Upgrade

Logout ▾

📄 📁 ▶ Run ⚙ Debug ■ Stop 🔄 Share 💾 Save {} Beautify ⬇️ ▾

Language Python 3 ⓘ ⚙

main.py

```
1 def removeElement(nums, val):
2     nums[:] = [x for x in nums if x != val]
3     return len(nums)
4
5 nums = [3, 2, 2, 3]
6 val = 3
7 print(removeElement(nums, val))
```

📄 📁 🖨 ⚙ 🗑

input

```
2
...Program finished with exit code 0
Press ENTER to exit console.
```

[About](#) • [FAQ](#) • [Blog](#) • [Terms of Use](#) • [Contact Us](#) • [GDB](#)

[Tutorial](#) • [Credits](#) • [Privacy](#)

© 2016 - 2024 GDB Online



OnlineGDB beta

online compiler and debugger for c/c++

Welcome, **Maithreyee Jagadeesan**

Create New Project

My Projects

Classroom new

Learn Programming

Programming Questions

Jobs new

Upgrade

Logout

Learn Python with
KodeKloud

[About](#) • [FAQ](#) • [Blog](#) • [Terms of Use](#) • [Contact Us](#) • [GDB](#)

[Tutorial](#) • [Credits](#) • [Privacy](#)

© 2016 - 2024 GDB Online



Language Python 3

main.py

```
1 def count_and_say(n):
2     if n == 1:
3         return "1"
4
5     previous_term = "1"
6     for _ in range(2, n + 1):
7         current_term = ""
8         count = 1
9         for j in range(1, len(previous_term)):
10            if previous_term[j] == previous_term[j - 1]:
11                count += 1
12            else:
13                current_term += str(count) + previous_term[j - 1]
14                count = 1
15        current_term += str(count) + previous_term[-1]
16        previous_term = current_term
17
18    return previous_term
19
20 n = 5
21 print(f"The {n}-th term of the Count and Say sequence is: {count_and_say(n)}")
22
```

input

The 5-th term of the Count and Say sequence is: 111221

...Program finished with exit code 0

Press ENTER to exit console.



OnlineGDB beta

online compiler and debugger for c/c++

Welcome, **Maithreyee Jagadeesan** 📌

Create New Project

My Projects

Classroom **new**

Learn Programming

Programming Questions

Jobs **new**

Upgrade

Logout ▾

Run Debug Stop Share Save Beautify

Language Python 3 ⚙️

main.py

```
1 def combination_sum(candidates, target):
2     def backtrack(remaining, start, path, result):
3         if remaining == 0:
4             result.append(list(path))
5             return
6         elif remaining < 0:
7             return
8
9         for i in range(start, len(candidates)):
10            path.append(candidates[i])
11            backtrack(remaining - candidates[i], i, path, result)
12            path.pop()
13
14        result = []
15        backtrack(target, 0, [], result)
16        return result
17
18 candidates = [2, 3, 6, 7]
19 target = 7
20 print(f"Combinations summing to {target} are: {combination_sum(candidates, target)}")
21
```

input

Combinations summing to 7 are: [[2, 2, 3], [7]]

...Program finished with exit code 0
Press ENTER to exit console.



OnlineGDB beta

online compiler and debugger for c/c++

Welcome, **Maithreyee Jagadeesan** 🧑

Create New Project

My Projects

Classroom new

Learn Programming

Programming Questions

Jobs new

Upgrade

Logout ▾

Learn Python with
KodeKloud

Run Debug Stop Share Save Beautyfy

Language Python 3

main.py

```
1 def combination_sum2(candidates, target):
2     def backtrack(remaining, start, path, result):
3         if remaining == 0:
4             result.append(list(path))
5             return
6         elif remaining < 0:
7             return
8
9         for i in range(start, len(candidates)):
10            # Skip duplicates
11            if i > start and candidates[i] == candidates[i - 1]:
12                continue
13            path.append(candidates[i])
14            backtrack(remaining - candidates[i], i + 1, path, result)
15            path.pop()
16
17        candidates.sort()
18        result = []
19        backtrack(target, 0, [], result)
20        return result
21
22    # Example usage
23    candidates = [10, 1, 2, 7, 6, 1, 5]
24    target = 8
25    print(f"Combinations summing to {target} are: {combination_sum2(candidates, target)}")
26
```

input

Combinations summing to 8 are: [[1, 1, 6], [1, 2, 5], [1, 7], [2, 6]]

...Program finished with exit code 0
Press ENTER to exit console.



OnlineGDB beta

online compiler and debugger for c/c++

Welcome, **Maithreyee Jagadeesan** 📌

Create New Project

My Projects

Classroom new

Learn Programming

Programming Questions

Jobs new

Upgrade

Logout ▾

Learn Python with
KodeKloud

[About](#) • [FAQ](#) • [Blog](#) • [Terms of Use](#) • [Contact Us](#) • [GDB](#)

[Tutorial](#) • [Credits](#) • [Privacy](#)

© 2016 - 2024 GDB Online

📄 🏠 ▶ Run ⚙ Debug ■ Stop 🔄 Share 💾 Save {} Beautify ⬇

Language Python 3 ⌵ ⚙

main.py

```
1 def permute_unique(nums):
2     def backtrack(start):
3         if start == len(nums):
4             result.append(nums[:])
5             return
6
7         seen = set()
8         for i in range(start, len(nums)):
9             if nums[i] in seen:
10                continue
11             seen.add(nums[i])
12             nums[start], nums[i] = nums[i], nums[start]
13             backtrack(start + 1)
14             nums[start], nums[i] = nums[i], nums[start]
15
16     nums.sort()
17     result = []
18     backtrack(0)
19     return result
20
21 nums = [1, 1, 2]
22 print(f"Unique permutations of {nums} are: {permute_unique(nums)}")
23
```

input



OnlineGDB beta

online compiler and debugger for c/c++

Welcome, **Maithreyee Jagadeesan**

Create New Project

My Projects

Classroom new

Learn Programming

Programming Questions

Jobs new

Upgrade

Logout

Learn Python with
KodeKloud



Language Python 3

main.py

```
1 import math
2
3 def get_permutation(n, k):
4     numbers = list(range(1, n + 1))
5     permutation = []
6
7     k -= 1
8
9     for i in range(n, 0, -1):
10         f = math.factorial(i - 1)
11
12         index = k // f
13
14         permutation.append(numbers[index])
15
16         numbers.pop(index)
17
18         k %= f
19
20     return ''.join(map(str, permutation))
21
22 n = 3
23 k = 3
24 print(f"The {k}-th permutation sequence of {n} numbers is: {get_permutation(n, k)}")
25
```