```python
def partition(arr, pivot):
    left = [x for x in arr if x < pivot]
    right = [x for x in arr if x > pivot]
    return left, pivot, right

def select(arr, k):
    if len(arr) <= 5:
        return sorted(arr)[k]

    sublists = [arr[i:i + 5] for i in range(0, len(arr), 5)]
    medians = [sorted(sublist)[len(sublist) // 2] for sublist in sublists]
    median_of_medians = select(medians, len(medians) // 2)

    left, pivot, right = partition(arr, median_of_medians)
    L, P = len(left), len(left) + 1

    if k < L:
        return select(left, k)
    elif k > L:
        return select(right, k - P)
    else:
        return pivot

# Test Cases
print(select([12, 3, 5, 7, 19], 2))  # Expected Output: 5
print(select([12, 3, 5, 7, 4, 19, 26], 3))  # Expected Output: 5
print(select([1, 2, 3, 4, 5, 6, 7, 8, 9, 10], 6))  # Expected Output: 6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
7
7
7

[Done] exited with code=0 in 0.137 seconds
```

```python
def partition(arr, pivot):
    left = [x for x in arr if x < pivot]
    right = [x for x in arr if x > pivot]
    return left, pivot, right

def select(arr, k):
    if len(arr) <= 5:
        return sorted(arr)[k]
    sublists = [arr[i:i + 5] for i in range(0, len(arr), 5)]
    medians = [sorted(sublist)[len(sublist) // 2] for sublist in sublists]
    median_of_medians = select(medians, len(medians) // 2)
    left, pivot, right = partition(arr, median_of_medians)
    L = len(left)
    if k < L:
        return select(left, k)
    elif k > L:
        return select(right, k - L - 1)
    else:
        return pivot

def median_of_medians(arr, k):
    return select(arr, k - 1)
arr1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
k1 = 6
print(median_of_medians(arr1, k1))

arr2 = [23, 17, 31, 44, 55, 21, 20, 18, 19, 27]
k2 = 5
print(median_of_medians(arr2, k2))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
6
21

[Done] exited with code=0 in 0.156 seconds
```

```python
from itertools import combinations

def closest_subset_sum(arr, target):
    n = len(arr)
    left_subsets = [sum(comb) for i in range(n//2+1) for comb in combination
    right_subsets = [sum(comb) for i in range(n-n//2+1) for comb in combinat
    right_subsets.sort()

    closest_sum = float('inf')
    for s in left_subsets:
        pos = binary_search_closest(right_subsets, target - s)
        if abs(target - (s + pos)) < abs(target - closest_sum):
            closest_sum = s + pos
    return closest_sum

def binary_search_closest(arr, target):
    low, high = 0, len(arr) - 1
    while low < high:
        mid = (low + high) // 2
        if arr[mid] < target:
            low = mid + 1
        else:
            high = mid
    return arr[low] if abs(arr[low] - target) < abs(arr[low - 1] - target) e

print(closest_subset_sum([45, 34, 4, 12, 5, 2], 42))  # Expected Output: 42
print(closest_subset_sum([1, 3, 2, 7, 4, 6], 10))  # Expected Output: 10

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python
41
10

[Done] exited with code=0 in 0.149 seconds
```

```python
from itertools import combinations

def subset_sum(arr, target_sum):
    # Split the array into two halves
    mid = len(arr) // 2
    left_half = arr[:mid]
    right_half = arr[mid:]

    # Generate all possible subset sums for each half
    left_sums = {sum(subset) for i in range(len(left_half) + 1) for subset i
    right_sums = {sum(subset) for i in range(len(right_half) + 1) for subset

    # Check if there is any combination of sums that equals the target
    for left_sum in left_sums:
        if (target_sum - left_sum) in right_sums:
            return True

    return False

# Test Case 1
E1 = [1, 3, 9, 2, 7, 12]
exact_sum1 = 15
print(subset_sum(E1, exact_sum1))  # Expected Output: True

# Test Case 2
E2 = [3, 34, 4, 12, 5, 2]
exact_sum2 = 15
print(subset_sum(E2, exact_sum2))  # Expected Output: True
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
True
True

[Done] exited with code=0 in 0.136 seconds
```

```python
def strassen_multiply(A, B):
    a, b, c, d = A[0][0], A[0][1], A[1][0], A[1][1]
    e, f, g, h = B[0][0], B[0][1], B[1][0], B[1][1]

    p1 = a * (f - h)
    p2 = (a + b) * h
    p3 = (c + d) * e
    p4 = d * (g - e)
    p5 = (a + d) * (e + h)
    p6 = (b - d) * (g + h)
    p7 = (a - c) * (e + f)

    C = [[p5 + p4 - p2 + p6, p1 + p2],
         [p3 + p4, p1 + p5 - p3 - p7]]
    return C

# Test Case
A = [[1, 7], [3, 5]]
B = [[6, 8], [4, 2]]
print(strassen_multiply(A, B))  # Expected Output: [[18, 14], [62, 66]]
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
[[34, 22], [38, 34]]

[Done] exited with code=0 in 0.157 seconds
```

```python
def karatsuba(x, y):
    if x < 10 or y < 10:
        return x * y
    n = max(len(str(x)), len(str(y)))
    m = n // 2
    high1, low1 = divmod(x, 10**m)
    high2, low2 = divmod(y, 10**m)
    z0 = karatsuba(low1, low2)
    z1 = karatsuba((low1 + high1), (low2 + high2))
    z2 = karatsuba(high1, high2)
    return (z2 * 10**(2*m)) + ((z1 - z2 - z0) * 10**m) + z0

# Test Case
print(karatsuba(1234, 5678))  # Expected Output: 7016652

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
7006652

[Done] exited with code=0 in 0.126 seconds
```

```python
def dice_throw(num_sides, num_dice, target):
    dp = [[0] * (target + 1) for _ in range(num_dice + 1)]
    dp[0][0] = 1
    for dice in range(1, num_dice + 1):
        for sum_val in range(1, target + 1):
            for face in range(1, num_sides + 1):
                if sum_val >= face:
                    dp[dice][sum_val] += dp[dice - 1][sum_val - face]
    return dp[num_dice][target]

print(dice_throw(6, 2, 7))  # Expected Output: 6
print(dice_throw(4, 3, 10))  # Expected Output: 27
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python
6
6

[Done] exited with code=0 in 0.147 seconds
```

```python
def assembly_line(n, a1, a2, t1, t2, e1, e2, x1, x2):
    dp1, dp2 = [0] * n, [0] * n
    dp1[0] = e1 + a1[0]
    dp2[0] = e2 + a2[0]
    for i in range(1, n):
        dp1[i] = min(dp1[i-1] + a1[i], dp2[i-1] + t2[i-1] + a1[i])
        dp2[i] = min(dp2[i-1] + a2[i], dp1[i-1] + t1[i-1] + a2[i])
    return min(dp1[-1] + x1, dp2[-1] + x2)

print(assembly_line(3, [4,5,3], [2,10,1], [7,4], [9,2], 10, 12, 18, 7))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
31

[Done] exited with code=0 in 0.131 seconds
```

```python
def three_line_schedule(station_times, transfer_times, dependencies):
    dp = [[0]*len(station_times[0]) for _ in range(3)]
    dp[0][0], dp[1][0], dp[2][0] = station_times[0][0], station_times[1][0], station_times[2][0]
    for i in range(1, len(station_times[0])):
        for j in range(3):
            dp[j][i] = station_times[j][i] + min(dp[k][i-1] + transfer_times[k][j] for k in range(3))
    return min(dp[j][-1] for j in range(3))

print(three_line_schedule([[5,9,3],[6,8,4],[7,6,5]], [[0,2,3],[2,0,4],[3,4,0]], [(0, 1), (1, 2)]))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
17

[Done] exited with code=0 in 0.138 seconds
```

```c
#include <stdio.h>
#define INF 999999

int minPath(int graph[][4], int n) {
    int minDist = INF;
    for (int i = 0; i < n; i++) {
        int dist = 0;
        for (int j = 0; j < n; j++) {
            dist += graph[i][j];
        }
        if (dist < minDist) minDist = dist;
    }
    return minDist;
}

int main() {
    int graph1[4][4] = {{0,10,15,20}, {10,0,35,25}, {15,35,0,30}, {20,25,30,0}};
    printf("Output: %d\n", minPath(graph1, 4));
    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
[Running] cd "c:\Users\hp\OneDrive\Desktop\" && gcc tempCodeRunnerFile.c -o tempCodeRunnerFile &&
"c:\Users\hp\OneDrive\Desktop\"tempCodeRunnerFile
Output: 45

[Done] exited with code=0 in 3.526 seconds
```

```python
import itertools

def tsp(cities, distances):
    shortest_distance = float('inf')
    shortest_path = None
    for path in itertools.permutations(cities):
        dist = sum(distances[path[i]][path[i+1]] for i in range(len(path) - 1))
        if dist < shortest_distance:
            shortest_distance = dist
            shortest_path = path
    return shortest_path, shortest_distance

distances = {
    'A': {'B': 10, 'C': 15, 'D': 20, 'E': 25},
    'B': {'A': 10, 'C': 35, 'D': 25, 'E': 30},
    'C': {'A': 15, 'B': 35, 'D': 30, 'E': 20},
    'D': {'A': 20, 'B': 25, 'C': 30, 'E': 15},
    'E': {'A': 25, 'B': 30, 'C': 20, 'D': 15},
}

print(tsp(['A', 'B', 'C', 'D', 'E'], distances))

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
(('B', 'A', 'C', 'E', 'D'), 60)

[Done] exited with code=0 in 0.135 seconds
```

```python
def longest_palindrome(s):
    res = ''
    for i in range(len(s)):
        for j in range(i, len(s)):
            if s[i:j+1] == s[i:j+1][::-1] and len(s[i:j+1]) > len(res):
                res = s[i:j+1]
    return res

print(longest_palindrome("babad"))  # Expected Output: "bab" or "aba"
print(longest_palindrome("cbbd"))   # Expected Output: "bb"

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
bab
bb

[Done] exited with code=0 in 0.178 seconds
```

```python
def longest_unique_substring(s):
    start, max_len, seen = 0, 0, {}
    for end, char in enumerate(s):
        if char in seen and seen[char] >= start:
            start = seen[char] + 1
        seen[char] = end
        max_len = max(max_len, end - start + 1)
    return max_len

print(longest_unique_substring("abcabcbb"))  # Expected Output: 3

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Cod

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
3

[Done] exited with code=0 in 0.125 seconds
```

```python
def word_break(s, wordDict):
    dp = [False] * (len(s) + 1)
    dp[0] = True
    for i in range(1, len(s) + 1):
        for word in wordDict:
            if dp[i - len(word)] and s[i-len(word):i] == word:
                dp[i] = True
                break
    return dp[-1]

print(word_break("leetcode", ["leet", "code"]))  # Expected Output: True
print(word_break("applepenapple", ["apple", "pen"]))  # Expected Output: True

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python
True
True

[Done] exited with code=0 in 0.141 seconds
```

```python
def can_segment_string(s, dictionary):
    dp = [False] * (len(s) + 1)
    dp[0] = True
    for i in range(1, len(s) + 1):
        for word in dictionary:
            if dp[i-len(word)] and s[i-len(word):i] == word:
                dp[i] = True
                break
    return dp[-1]

print(can_segment_string("ilikesamsung", ["i", "like", "sam", "sung", "samsung"]))  # Expected Output: T
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
True

[Done] exited with code=0 in 0.134 seconds
```

```python
def full_justify(words, maxWidth):
    res, line, length = [], [], 0
    for word in words:
        if length + len(word) + len(line) > maxWidth:
            for i in range(maxWidth - length):
                line[i % (len(line)-1 or 1)] += ' '
            res.append(''.join(line))
            line, length = [], 0
        line += [word]
        length += len(word)
    return res + [' '.join(line).ljust(maxWidth)]

print(full_justify(["This", "is", "an", "example", "of", "text", "justification."], 16))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Code

```
[Done] exited with code=0 in 0.134 seconds

[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
['This    is    an', 'example  of text', 'justification.  ']

[Done] exited with code=0 in 0.136 seconds
```

```python
class WordFilter:
    def __init__(self, words):
        self.word_map = {}
        for index, word in enumerate(words):
            for i in range(len(word) + 1):
                for j in range(len(word) + 1):
                    self.word_map[word[:i] + '#' + word[-j:]] = index

    def f(self, pref, suff):
        return self.word_map.get(pref + '#' + suff, -1)

# Example Usage
word_filter = WordFilter(["apple"])
print(word_filter.f("a", "e"))  # Expected Output: 0
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
[Done] exited with code=0 in 0.221 seconds

[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
0

[Done] exited with code=0 in 0.162 seconds
```