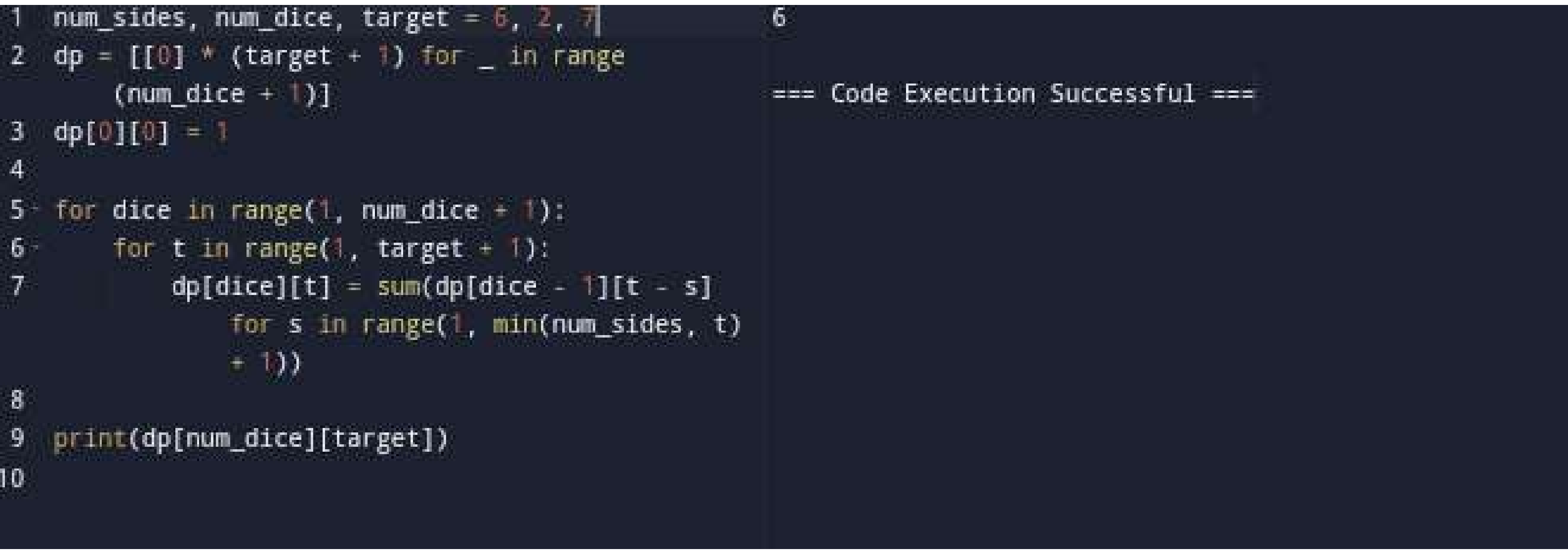
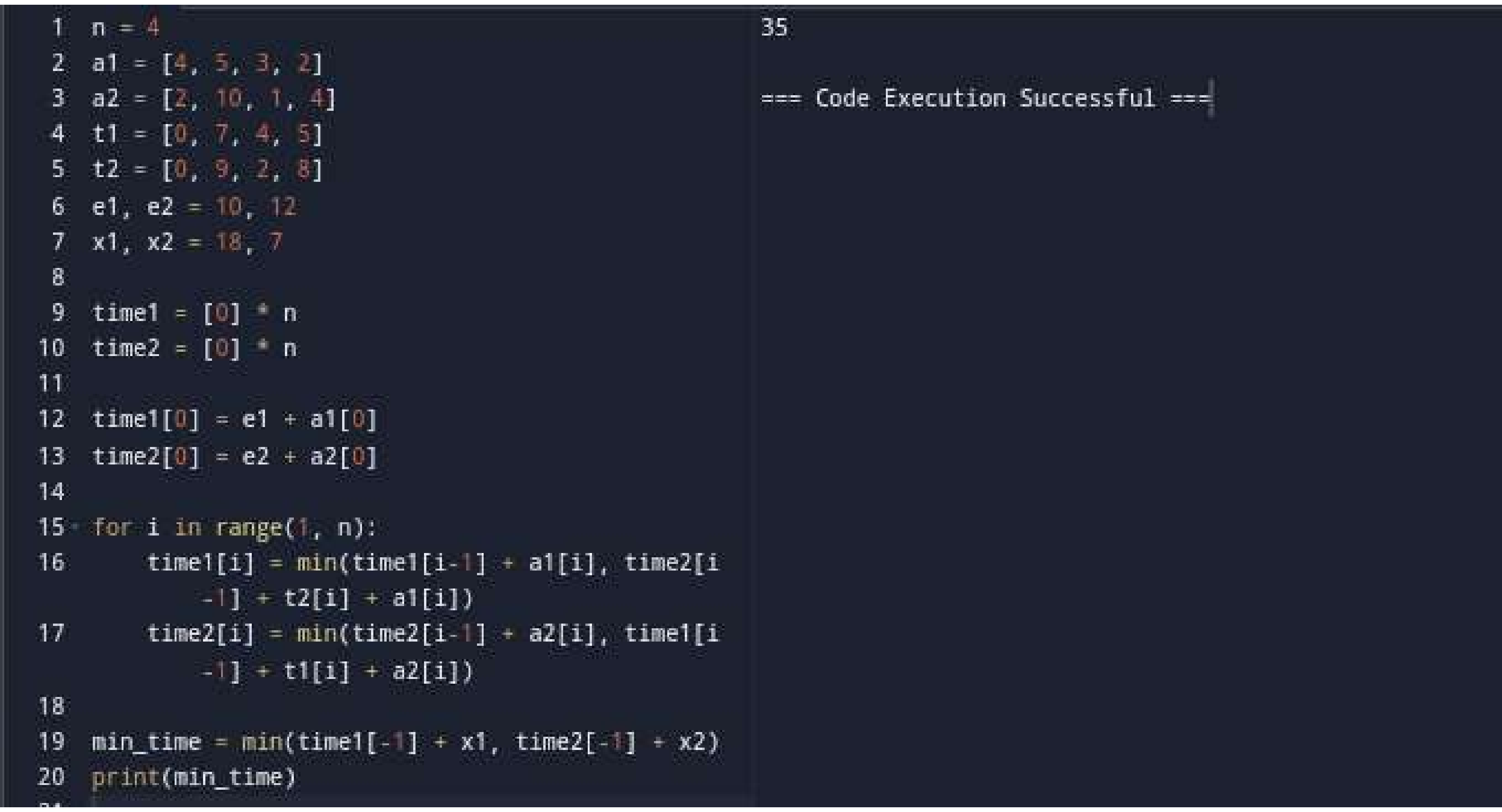


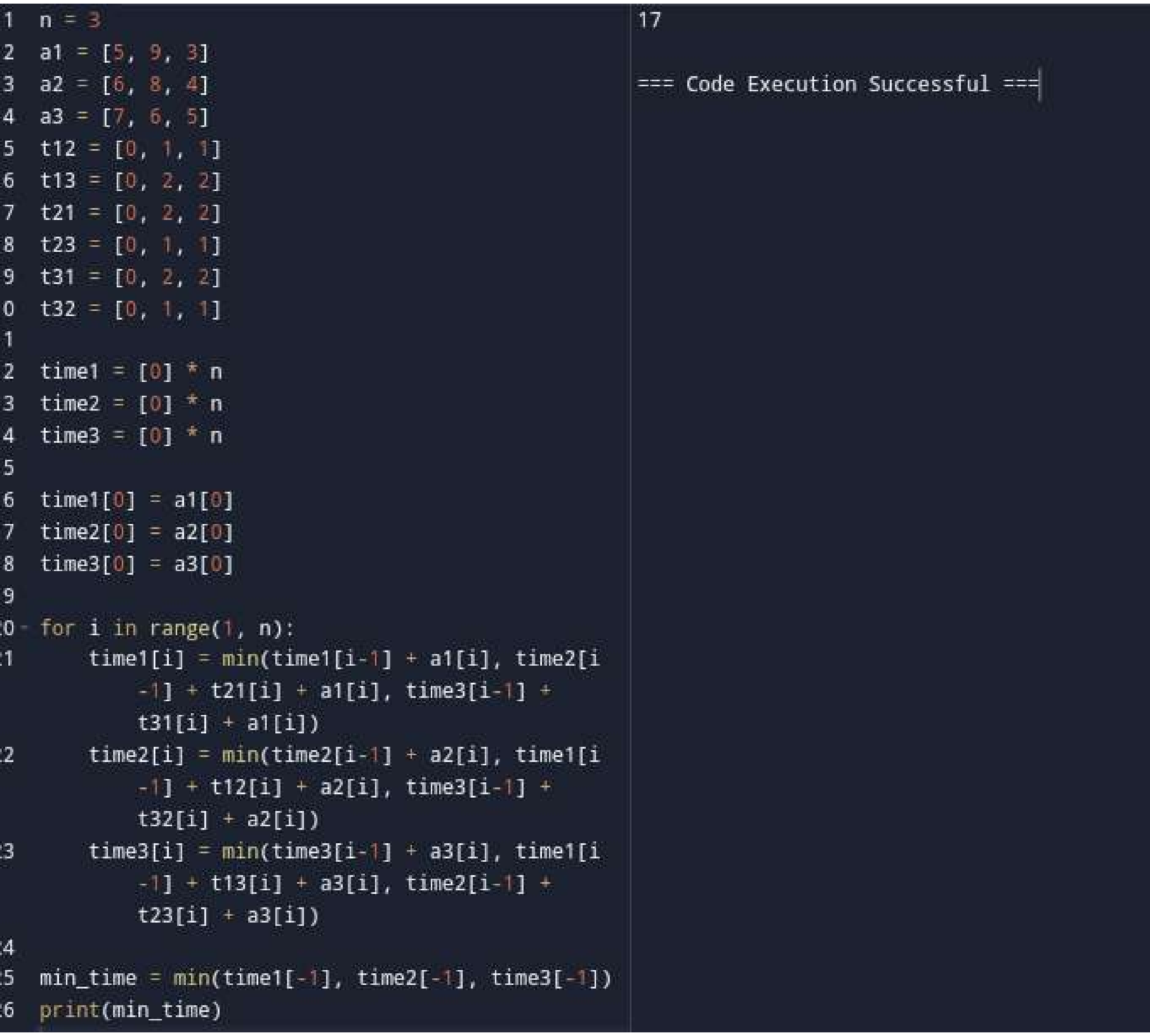
```python
num_sides, num_dice, target = 6, 2, 7
dp = [[0] * (target + 1) for _ in range
    (num_dice + 1)]
dp[0][0] = 1

for dice in range(1, num_dice + 1):
    for t in range(1, target + 1):
        dp[dice][t] = sum(dp[dice - 1][t - s]
            for s in range(1, min(num_sides, t)
            + 1))

print(dp[num_dice][target])
```

```
6

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



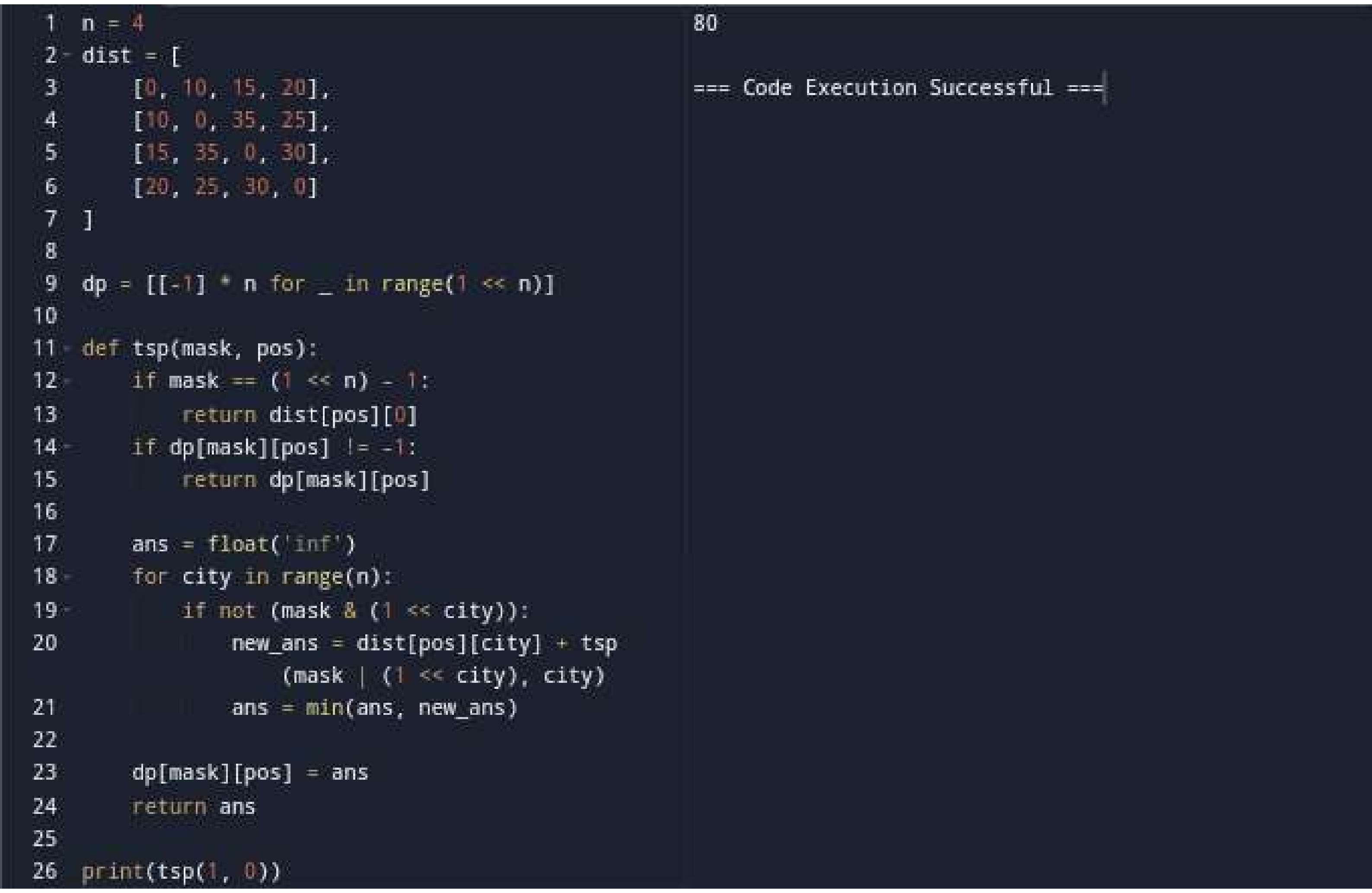
```python
n = 4
a1 = [4, 5, 3, 2]
a2 = [2, 10, 1, 4]
t1 = [0, 7, 4, 5]
t2 = [0, 9, 2, 8]
e1, e2 = 10, 12
x1, x2 = 18, 7

time1 = [0] * n
time2 = [0] * n

time1[0] = e1 + a1[0]
time2[0] = e2 + a2[0]

for i in range(1, n):
    time1[i] = min(time1[i-1] + a1[i], time2[i
        -1] + t2[i] + a1[i])
    time2[i] = min(time2[i-1] + a2[i], time1[i
        -1] + t1[i] + a2[i])

min_time = min(time1[-1] + x1, time2[-1] + x2)
print(min_time)
```

```
35

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



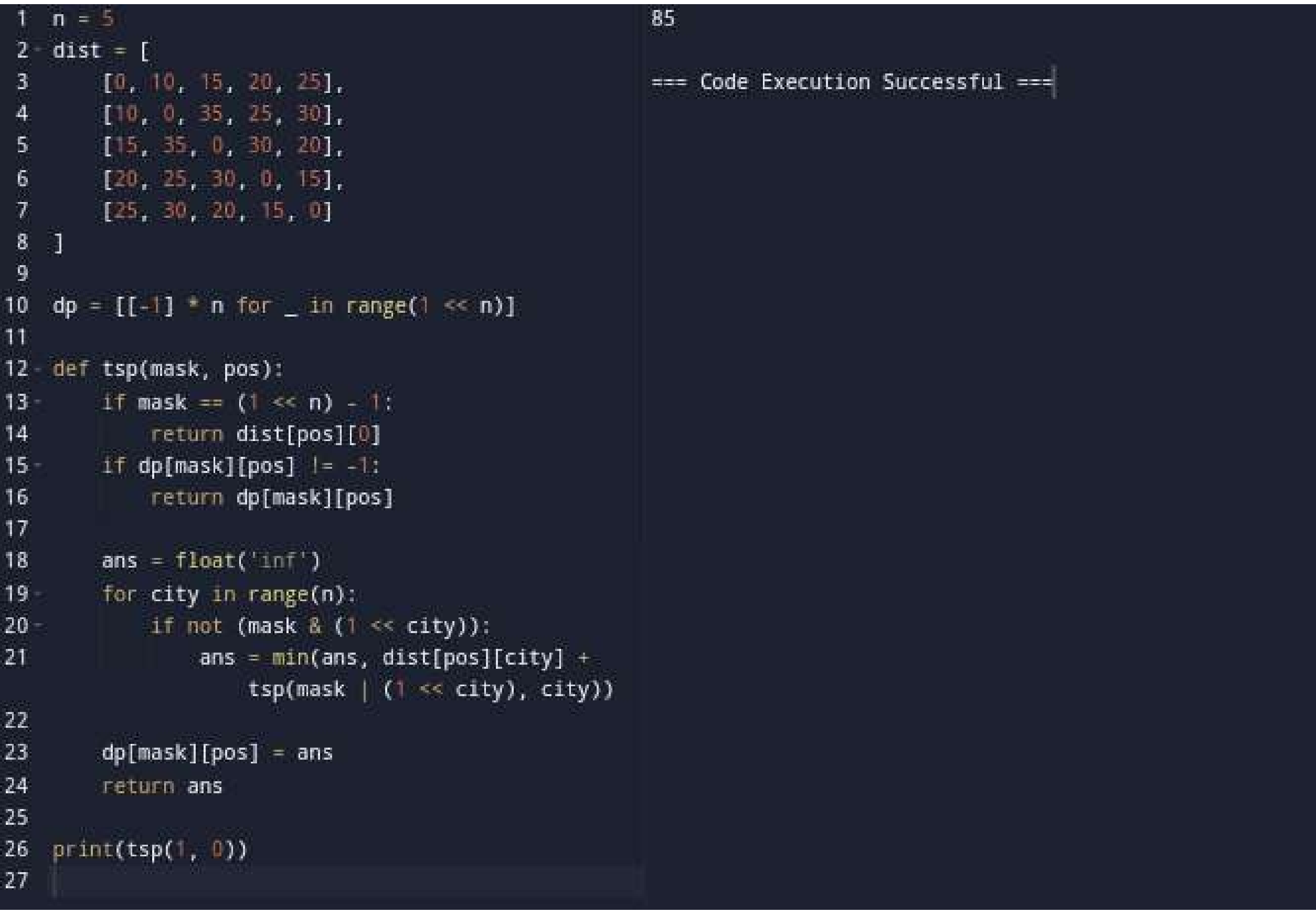
```python
n = 3
a1 = [5, 9, 3]
a2 = [6, 8, 4]
a3 = [7, 6, 5]
t12 = [0, 1, 1]
t13 = [0, 2, 2]
t21 = [0, 2, 2]
t23 = [0, 1, 1]
t31 = [0, 2, 2]
t32 = [0, 1, 1]

time1 = [0] * n
time2 = [0] * n
time3 = [0] * n

time1[0] = a1[0]
time2[0] = a2[0]
time3[0] = a3[0]

for i in range(1, n):
    time1[i] = min(time1[i-1] + a1[i], time2[i
        -1] + t21[i] + a1[i], time3[i-1] +
        t31[i] + a1[i])
    time2[i] = min(time2[i-1] + a2[i], time1[i
        -1] + t12[i] + a2[i], time3[i-1] +
        t32[i] + a2[i])
    time3[i] = min(time3[i-1] + a3[i], time1[i
        -1] + t13[i] + a3[i], time2[i-1] +
        t23[i] + a3[i])

min_time = min(time1[-1], time2[-1], time3[-1])
print(min_time)
```

```
17

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



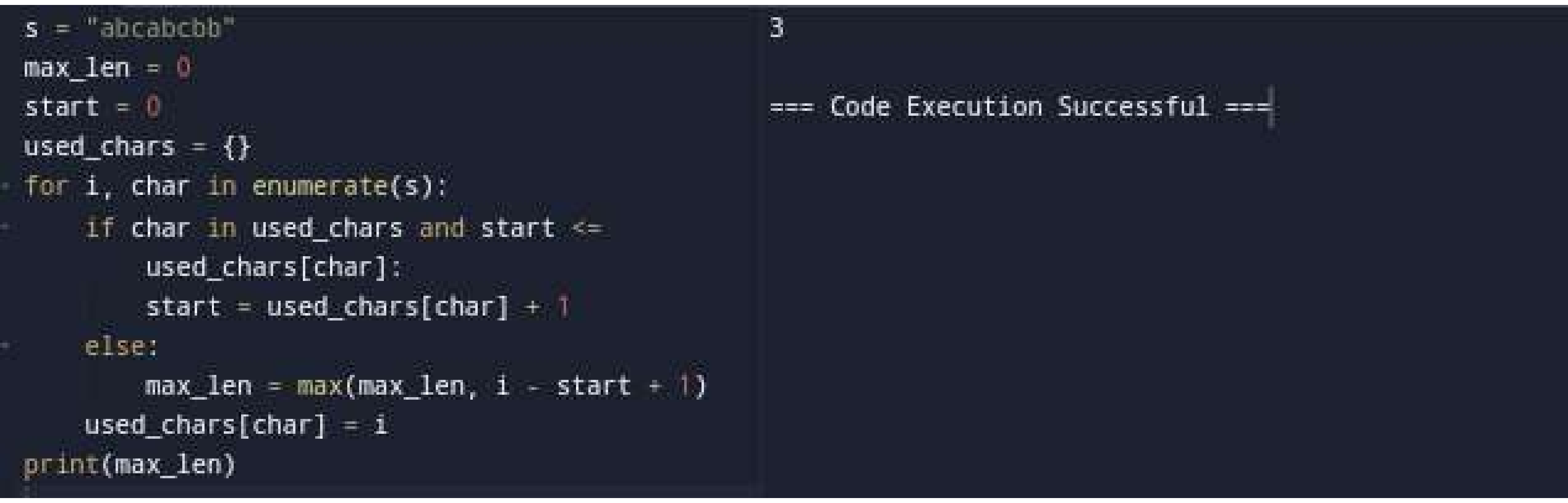
```python
n = 4
dist = [
    [0, 10, 15, 20],
    [10, 0, 35, 25],
    [15, 35, 0, 30],
    [20, 25, 30, 0]
]

dp = [[-1] * n for _ in range(1 << n)]

def tsp(mask, pos):
    if mask == (1 << n) - 1:
        return dist[pos][0]
    if dp[mask][pos] != -1:
        return dp[mask][pos]

    ans = float('inf')
    for city in range(n):
        if not (mask & (1 << city)):
            new_ans = dist[pos][city] + tsp
                (mask | (1 << city), city)
            ans = min(ans, new_ans)

    dp[mask][pos] = ans
    return ans

print(tsp(1, 0))
```

```
80

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

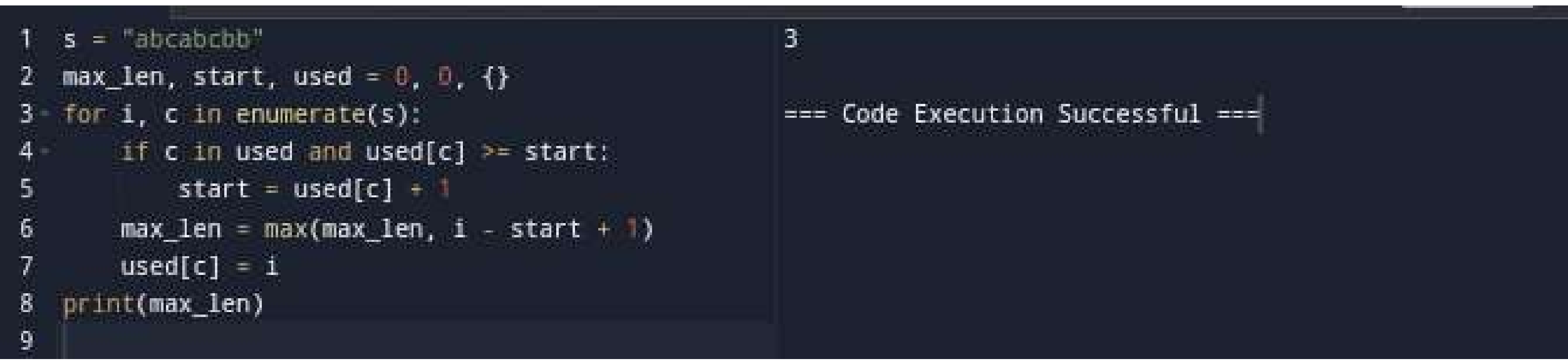


```python
n = 5
dist = [
    [0, 10, 15, 20, 25],
    [10, 0, 35, 25, 30],
    [15, 35, 0, 30, 20],
    [20, 25, 30, 0, 15],
    [25, 30, 20, 15, 0]
]

dp = [[-1] * n for _ in range(1 << n)]

def tsp(mask, pos):
    if mask == (1 << n) - 1:
        return dist[pos][0]
    if dp[mask][pos] != -1:
        return dp[mask][pos]

    ans = float('inf')
    for city in range(n):
        if not (mask & (1 << city)):
            ans = min(ans, dist[pos][city] +
                tsp(mask | (1 << city), city))

    dp[mask][pos] = ans
    return ans

print(tsp(1, 0))
```

```
85

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

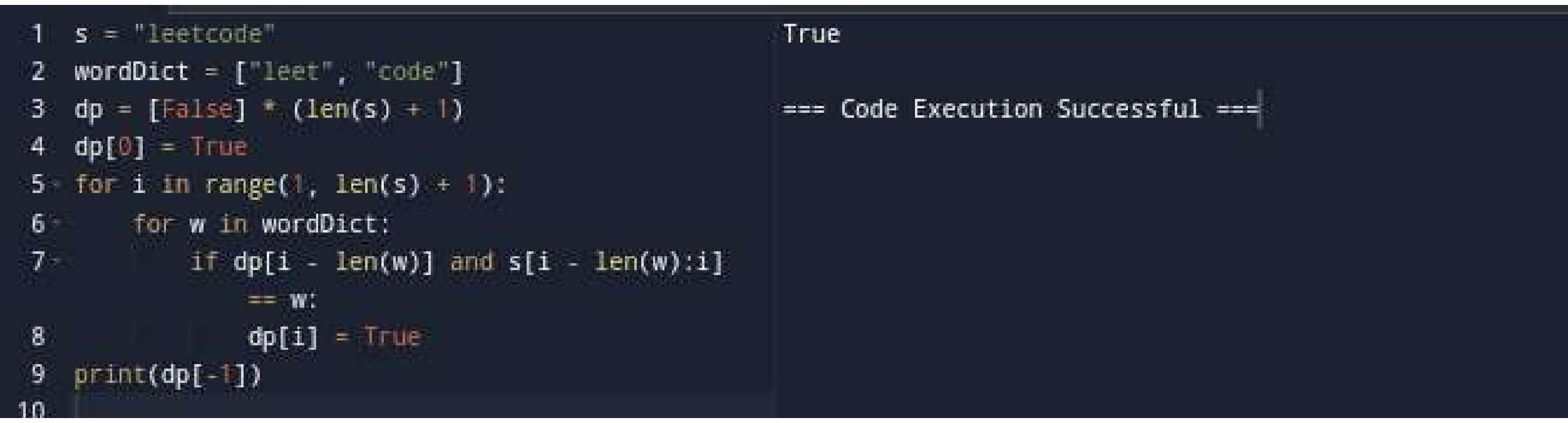

```python
s = "abcabcbb"
max_len = 0
start = 0
used_chars = {}
for i, char in enumerate(s):
    if char in used_chars and start <=
        used_chars[char]:
        start = used_chars[char] + 1
    else:
        max_len = max(max_len, i - start + 1)
    used_chars[char] = i
print(max_len)
```

```
3

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



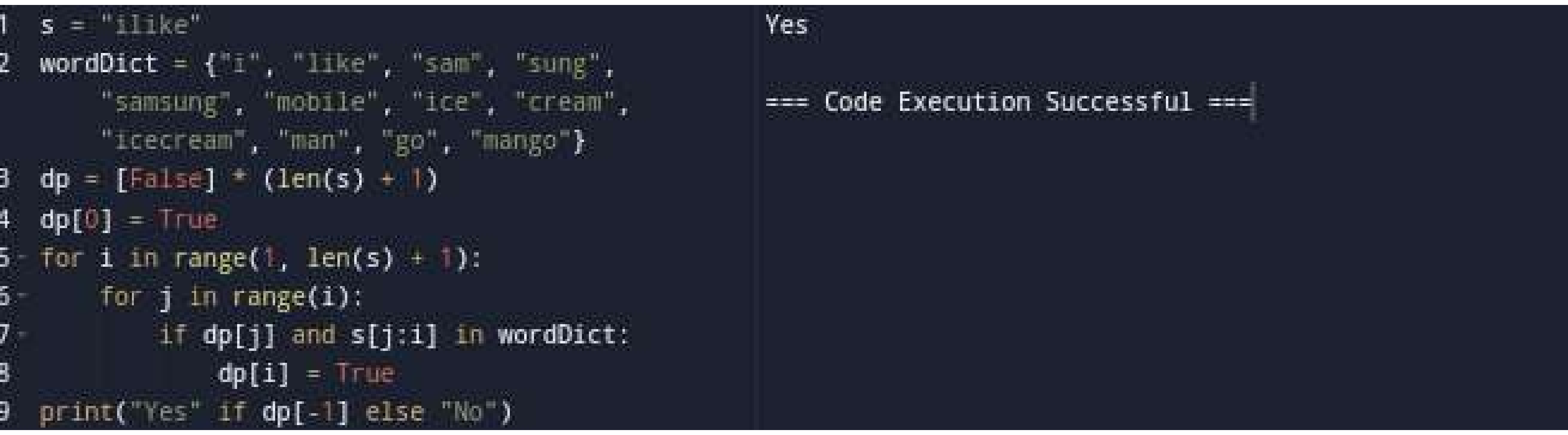
```python
s = "abcabcbb"
max_len, start, used = 0, 0, {}
for i, c in enumerate(s):
    if c in used and used[c] >= start:
        start = used[c] + 1
    max_len = max(max_len, i - start + 1)
    used[c] = i
print(max_len)

```

```
3

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



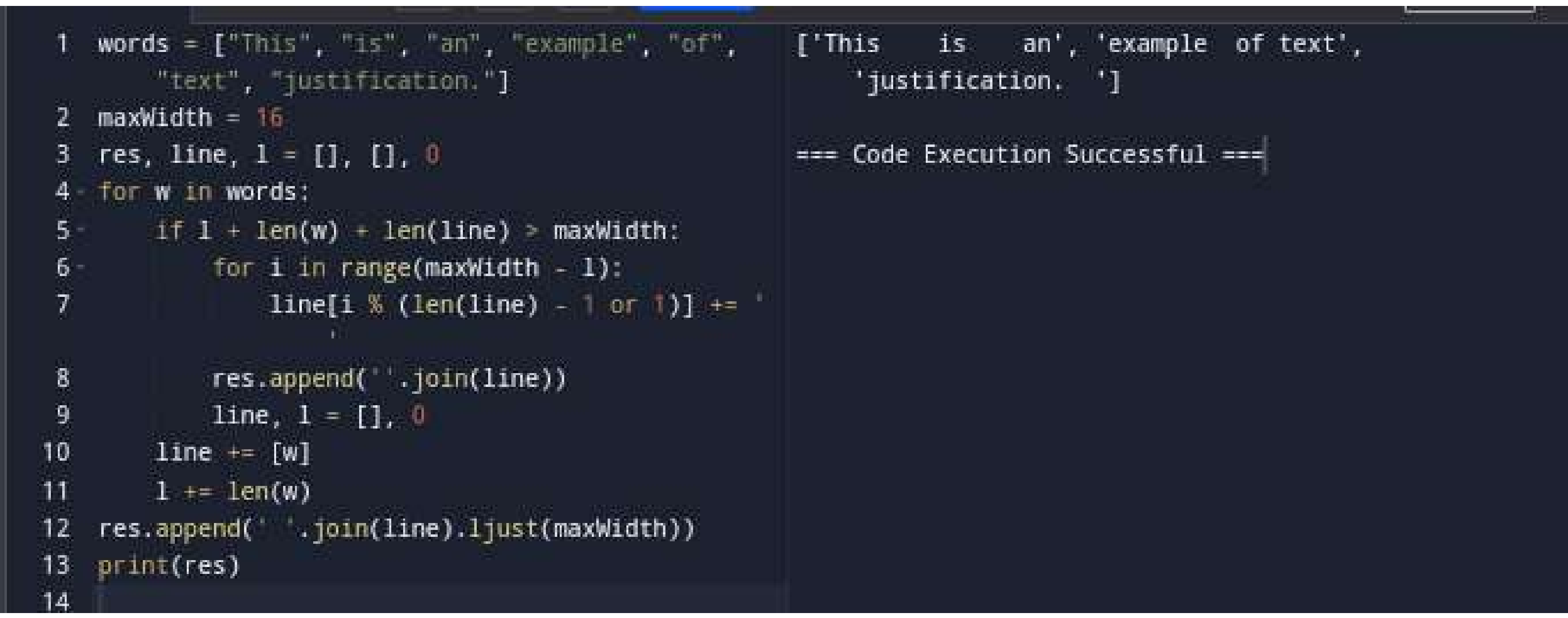
```python
s = "leetcode"
wordDict = ["leet", "code"]
dp = [False] * (len(s) + 1)
dp[0] = True
for i in range(1, len(s) + 1):
    for w in wordDict:
        if dp[i - len(w)] and s[i - len(w):i]
            == w:
            dp[i] = True
print(dp[-1])
```

```
True

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



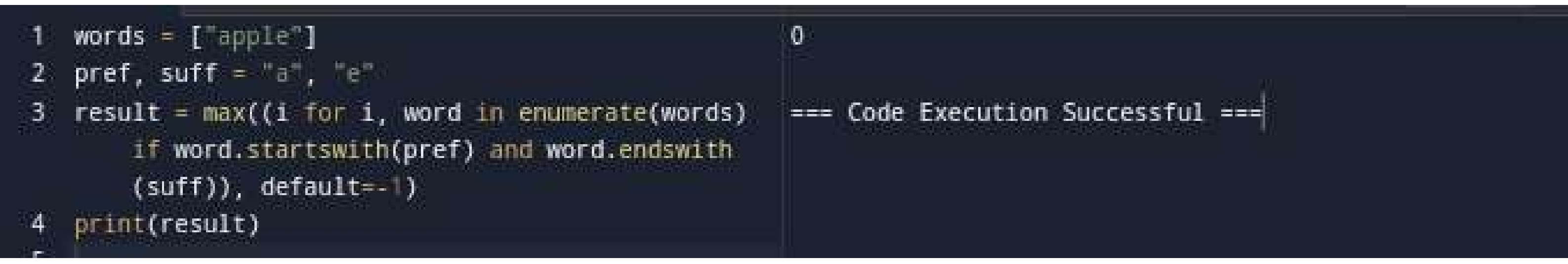
```python
s = "ilike"
wordDict = {"i", "like", "sam", "sung",
    "samsung", "mobile", "ice", "cream",
    "icecream", "man", "go", "mango"}
dp = [False] * (len(s) + 1)
dp[0] = True
for i in range(1, len(s) + 1):
    for j in range(i):
        if dp[j] and s[j:i] in wordDict:
            dp[i] = True
print("Yes" if dp[-1] else "No")
```

```
Yes

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



```python
words = ["This", "is", "an", "example", "of", "text", "justification."]
maxWidth = 16
res, line, l = [], [], 0
for w in words:
    if l + len(w) + len(line) > maxWidth:
        for i in range(maxWidth - l):
            line[i % (len(line) - 1 or 1)] += ' '
        res.append(''.join(line))
        line, l = [], 0
    line += [w]
    l += len(w)
res.append(' '.join(line).ljust(maxWidth))
print(res)
```

```
['This    is    an', 'example  of text',
    'justification.  ']

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



```python
words = ["apple"]
pref, suff = "a", "e"
result = max((i for i, word in enumerate(words)
    if word.startswith(pref) and word.endswith
    (suff)), default=-1)
print(result)
```

```
0

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