

ASSIGNMENT-2

NAME: SHIVAM SINGH

1.

```
def wordBreak(s, wordDict):
    n = len(s)

    dp = [False] * (n + 1)
    dp[0] = True

    for i in range(1, n + 1):
        for j in range(i):

            if dp[j] and s[j:i] in wordDict:
                dp[i] = True
                break

    return dp[n]

# Example:
wordDict = {"i", "like", "sam", "sung", "samsung", "mobile", "ice", "cream",
            "icecream", "man", "go", "mango"}

input_str1 = "ilike"
output1 = wordBreak(input_str1, wordDict)
print(f"Input: {input_str1}\nOutput: {'Yes' if output1 else 'No'}\n")
```

2.

```
def minSquares(n):

    dp = [float('inf')] * (n + 1)

    dp[0] = 0

    for i in range(1, n + 1):
        j = 1
        while j * j <= i:
            dp[i] = min(dp[i], 1 + dp[i - j * j])
            j += 1

    return dp[n]

# Example
input1 = 100
output1 = minSquares(input1)
print(f"Input: {input1}\nOutput: {output1}\n")
```

3.

```
def isDivisibleBy7(n):
    if n < 0:
        return isDivisibleBy7(-n)

    if n == 0 or n == 7:
        return True

    if n < 10:
        return False

    return isDivisibleBy7((n // 10) - 2 * (n % 10))

# Example:
number1 = 371
result1 = isDivisibleBy7(number1)
print(f"Number {number1} is {'divisible' if result1 else 'not divisible'} by 7.\n")
```

4.

```
def countAndSay(n):
    if n == 1:
        return "1"

    prev_term = countAndSay(n - 1)

    result = ""
    count = 1

    for i in range(1, len(prev_term)):
        if prev_term[i] == prev_term[i - 1]:
            count += 1
        else:
            result += str(count) + prev_term[i - 1]
            count = 1
    result += str(count) + prev_term[-1]
    return result

# Example:
input1 = 3
output1 = countAndSay(input1)
print(f"Input: {input1}\nOutput: {output1}\n")
```