13. Write C program that demonstrates the usage of these notations by analyzing the time complexity of some example algorithms.

Program:

```
def constant_time(n):
    """
    Example algorithm with O(1) time complexity.
    No matter how large n is, the time taken remains constant.
    """
    print("This is an example of O(1) time complexity.")
    print("No matter how large n is, the time taken remains constant.")

def linear_time(n):
    """
    Example algorithm with O(n) time complexity.
    The time taken increases linearly with the size of n.
    """
    print("This is an example of O(n) time complexity.")
    print("The time taken increases linearly with the size of n.")
    for i in range(n):
        print(i, end=' ')
    print()

def quadratic_time(n):
    """
    Example algorithm with O(n^2) time complexity.
    The time taken increases quadratically with the size of n.
```

```
    """
    print("This is an example of O(n^2) time complexity.")
    print("The time taken increases quadratically with the
size of n.")
    for i in range(n):
        for j in range(n):
            print(f"({i},{j})", end=' ')
        print()

# Example usage
n = int(input("Enter the value of n: "))

constant_time(n)  # O(1)
linear_time(n)    # O(n)
quadratic_time(n) # O(n^2)
```

Output:

```
"C:\Program Files\Python312\python.exe" "C:\Work Space\DAA COADS.PYTHON\python 13.py"
Enter the value of n: 2
This is an example of O(1) time complexity.
No matter how large n is, the time taken remains constant.
This is an example of O(n) time complexity.
The time taken increases linearly with the size of n.
0 1
This is an example of O(n^2) time complexity.
The time taken increases quadratically with the size of n.
(0,0) (0,1)
(1,0) (1,1)

Process finished with exit code 0
```

Time complexity:

O(√n)