# PROGRAM-19

## SIMULATED ANNEALING ALGORITHM PROBLEM

## AIM:-

To write and execute the python program for the Simulated annealing algorithm program.

## PROCEDURE:-

**Imports and Setup:**

- Import the required libraries: math and random.

**Simulated Annealing Function:**

- Define the simulated annealing algorithm function.

**Define Cost Function:**

- Define the cost function. In this case, it's the Rastrigin function simplified to 1D.

**Set Initial Parameters:**

- Set the initial temperature and cooling rate.

**Execution and Output:**

- Execute the simulated annealing function with the defined parameters.
- Print the optimum state found.

.

## CODING:-

```
import math, random

def simulated_annealing(cost_func, start_temp, cooling_rate):

    state = current = random.uniform(-10, 10)

    temp = start_temp

    while temp > 0.001:

        new_state = current + random.uniform(-1, 1)

        cost_diff = cost_func(new_state) - cost_func(current)

        if cost_diff < 0 or math.exp(-cost_diff / temp) > random.random():
```

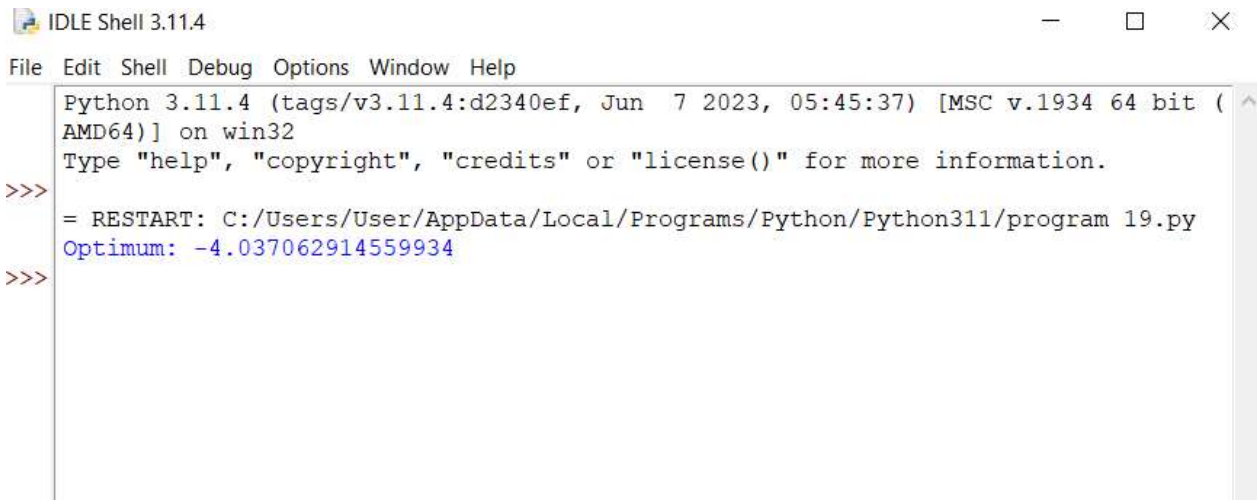
```
            state = new_state
        current = new_state
        temp *= cooling_rate
    return state

cost_func = lambda x: (x ** 2) - (10 * math.cos(2 * math.pi * x)) + 10  # Rastrigin function simplified to 1D

start_temp, cooling_rate = 1000, 0.98

print("Optimum:", simulated_annealing(cost_func, start_temp, cooling_rate))
```

## OUTPUT:-




## RESULT:-

Hence the program has been successfully executed and verified.