LAB-05 :-

Stimulated Annealing :-

Algo ~~Step1~~:- objective function :- $x^2 + 5 \sin x$

```
function Stimulated Annealing (intial_state,
initial_temperature, cooling_state iteration)

Current_state = initial_state
best - state = Current_state
best - cost = objective function (current_state)
 temp = initial_temperature

 while temp >1:

     for i ← 1 to iterations
     new_state = Neighbour (current_state)
     curr-cost =objective Function (curr-state)
    new_cost = objective Function (new_state)

 if Ap (curr-cost, new-cost, temp) > Random(0,1)
        current_state = new_state

  if new-cost < best_cost
       best state = new-state
       best - cost = new cost

  tmp* = Cooling_rate

 return (best_state, best-cost)

Function objective Function(state):
       cost = 0
   for x in state

      cost = x^2+5 sin x
```

```
return cost

function Neighbor (state)

    new_state = state_copy()
    index = Random (0, length (state)-1)
    new_state [index] += Random(-1, 1)
    return new state.

function AP( curr state, new_cost, temp)
    if (new_cost < curr_(cost));
        return:
    else
        return (curr_cost - new_cost)/temp
```

Code

```
def main():
    initial_temp = 1000
    cooling_rate = 0.9
    iterations = 1000
    initial_state = [random.uniform(-10,10)
        for _ in range (2)]

best_state, best_cost = since (initial_state,
    initial_temp, cooling_rate, iteration)

print("Best state: {best state}")
print("Best cost: {best_cost}")
```