LAB 5 (22|10|24)

Simmulation of Annealing:

Objective fun: $x^2 + 5\sin x$

Step1: define function called "simmulation Annealing"

```
def simulation_annealing(initial_state, initialtemp, coolingstate, i)
        current = initialstate
        best = current
        best = objective(current)
        temp = initialTemp
        while temp >1:
            for i<-1 to it:
                new = neighbour(current)
                curr = objective(current)
               [new_cost = object(new)
               if Function(curr, new_cost, temp) -> Range(0,1):
                    current = new
               if new < best:
                    best = new
               temp = why
        return(best, best_cost)
```

Step2: how define is objective function to change the state

```
   def. objective(state):
        cost = 0
        for ele in state:
            cost + = ele + sin(ele)
        return cost
```

Step 3: next function is to check|search for neighbour.

```
  def neighbour(state):
        new = state.copy()
        ind = Rand(0, curr(state) - 1))
        new[ind] + = Rand(-1,1)
        return new
```

Step 4: a fxn for acceptance probability

```
def function( curr-cost, new-cost , temp)
    if (new-cost < curr cost):
        return 1

    else:
        return e^((new-cost - current-cost)/T)
```

Code:

```
def main():
    initialTemp = 1000
    coolingRate = 0.9
    iterations = 1000
    initialState = [rd.uniform (10.10) for _ in range(1)]
    bestState, bestCost = Simulation.analog (initialState, initialTemp,
        coolingRate, iteration)
    print ("Best State : ", bestState)
    print ("BestCost : ", best cost)

if __name__ == "__main__":
        main()
```

Output:

```
Best state : [-0.52671, -0.39996, -0.51576, -0.30521, -0.45250]

Best Cost : -1.127671
```