Bahria University,
Karachi Campus

LAB EXPERIMENT NO.
07

LIST OF TASKS

TASK NO	OBJECTIVE
1.	**Using a genetic Algorithm Create a GUI based Guess a password given the number of correct letters in the guess. Build a mutation Engine**
2.	
3.	
4.	
5.	

Submitted On:
4/5/2024

TASK NO 1: Using a genetic Algorithm Create a GUI based Guess a password given the number of correct letters in the guess. Build a mutation Engine.

```
import tkinter as tk
import random
import string
POPULATION_SIZE = 100
MUTATION_RATE = 0.1
TARGET_PASSWORD = "AILAB7970"
class GeneticAlgorithm:
    def __init__(self):
        self.population = []
    def generate_population(self):
        for _ in range(POPULATION_SIZE):
            password
="".join(random.choice(string.ascii_lowercase) for _ in
range(len(TARGET_PASSWORD)))
            self.population.append(password)
    def fitness(self, password):
        return sum(c1 == c2 for c1, c2 in
zip(password,TARGET_PASSWORD))
    def mutate(self, password):
        mutated_password = ""
        for char in password:
            if random.random() < MUTATION_RATE:
                mutated_password +=
random.choice(string.ascii_lowercase)
            else:
                mutated_password += char
        return mutated_password
    def evolve(self):
        new_generation = []
        for _ in range(POPULATION_SIZE):
            parent1 = random.choice(self.population)
            parent2 = random.choice(self.population)
            midpoint =
random.randint(0,len(TARGET_PASSWORD))
            child = parent1[:midpoint] +
parent2[midpoint:]
            child = self.mutate(child)
            new_generation.append(child)
        self.population = new_generation
class PasswordGuesserApp:
    def __init__(self, master):
        self.master = master
        self.master.title("Password Guesser")
        self.genetic_algorithm = GeneticAlgorithm()
        self.genetic_algorithm.generate_population()
        self.label = tk.Label(master, text="Enter your
guess:")
        self.label.pack()
        self.entry = tk.Entry(master)
        self.entry.pack()
        self.button = tk.Button(master,
text="Submit",command=self.guess_password)
        self.button.pack()
        self.result_label = tk.Label(master, text="")
        self.result_label.pack()
    def guess_password(self):
        guess = self.entry.get()
        fitness = self.genetic_algorithm.fitness(guess)
        if guess == TARGET_PASSWORD:
            self.result_label.config(text="Congratulations!
You guessed the password.")
        else:
            self.result_label.config(text=f"Password
Fittness: {fitness}")
        self.genetic_algorithm.evolve()
def main():
    root = tk.Tk()
    app = PasswordGuesserApp(root)
    root.mainloop()
if __name__ == "__main__":
    main()
```

OUTPUT: