

```
import random

def binary_to_decimal(binary_str):
    return int(binary_str, 2)

def binaryConversion(chromosome):
    return bin(chromosome)[2:].zfill(5)

def crossover(population, crossoverRate):
    offspring = []

    for i in range(0, len(population), 2):
        parent1 = population[i]
        parent2 = population[i + 1]

        if random.random() < crossoverRate * 100:
            crossoverPoint = random.randint(1, len(
parent1) - 1)
            child1 = parent1[:crossoverPoint] + parent2
[crossoverPoint:]
            child2 = parent2[:crossoverPoint] + parent1
[crossoverPoint:]

            offspring.extend([child1, child2])
        else:
            offspring.extend([parent1, parent2])

    return offspring

pop = ['13', '10', '7', '9', '5', '16']
population = [bin(int(decimal))[2:].zfill(4) for
decimal in pop]
print(population)

crossoverRate = 0.25

print("Initial Population: ")
for ind, binary in enumerate(population):
    print(f"Binary: {binary}, Decimal: {
binary_to_decimal(binary)}")
```

```
offspringPopulation = crossover(population,  
crossoverRate)  
  
print("\nOffspring Population after crossover: ")  
for ind, binary in enumerate(offspringPopulation):  
    print(f"Binary: {binary}, Decimal: {  
binary_to_decimal(binary)}")
```