* **Parameter settings**

gene\_length = 30 #二進制的長度

population\_size= 500 #每一代染色體的數量

mutation\_rate = 0.5 #變異機率

tolerance = 1e-11 # 停止條件

aus = 1e6

target = float(input("請輸入您要計算的根號值："))

* **def Binary(chromosome)**

這個部分主要是將我的數值轉為二進制的定義

* **def decimal\_number(chromosome)**

這個部分主要是將我的數值轉為十進制的定義

* **def generate\_population(population\_size, gene\_length)**

生成初始族群的函數

* **def fitness\_function(individual)**

# 評估染色體適應度的函數

chromosome = decimal\_number(individual) / (10\*\*11)

score = aus - abs(chromosome\*\*2 - target)

return max(0, score) #確保適應度值大於等於零

**def crossover(parent1, parent2)**

# 交叉遺傳的函數

# 隨機選擇交叉點，避免在邊界處交叉

point = random.randint(1, len(parent1) - 2)

# 生成子代1，將父母1的前段和父母2的後段結合

child1 = parent1[:point] + parent2[point:]

# 生成子代2，將父母2的前段和父母1的後段結合

child2 = parent2[:point] + parent1[point:]

return child1, child2

* **def mutate(individual)**

# 變異的函數

# 隨機選擇染色體中的一個位置

mutated\_index = random.randint(0, len(individual) - 1)

# 將選定位置的基因取反，即0變為1，1變為0

mutated\_gene = '0' if individual[mutated\_index] == '1' else '1'

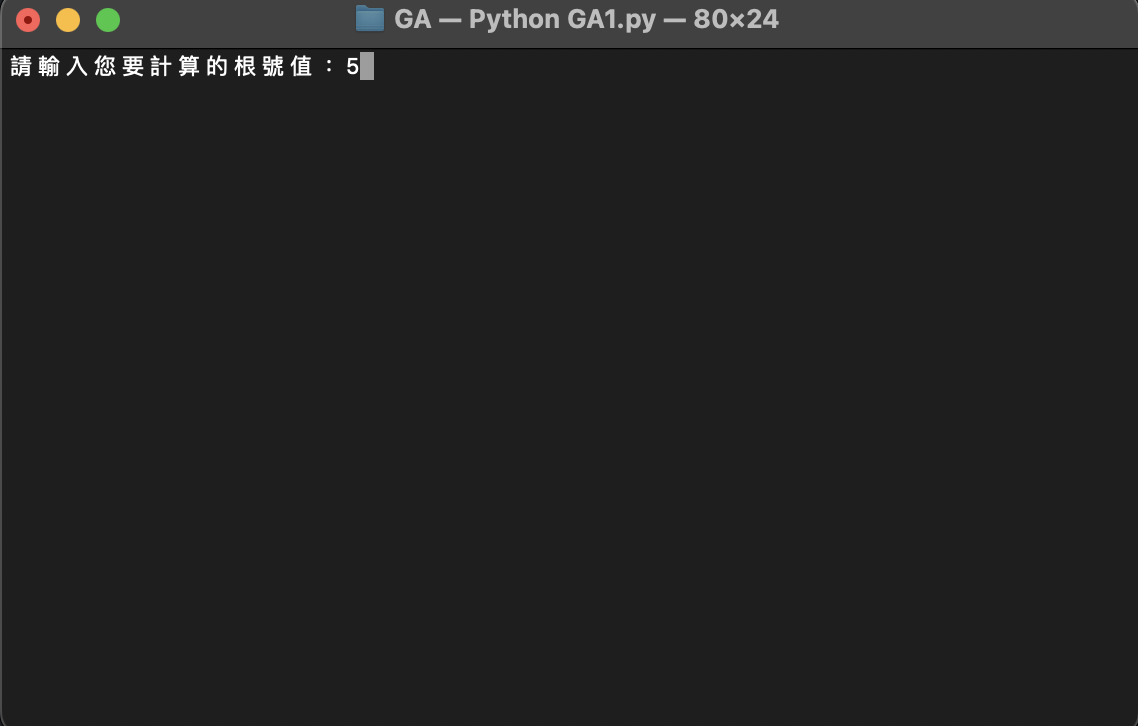
# 產生變異後的染色體，將選定位置的基因替換為變異後的基因

return individual[:mutated\_index] + mutated\_gene + individual[mutated\_index + 1:]

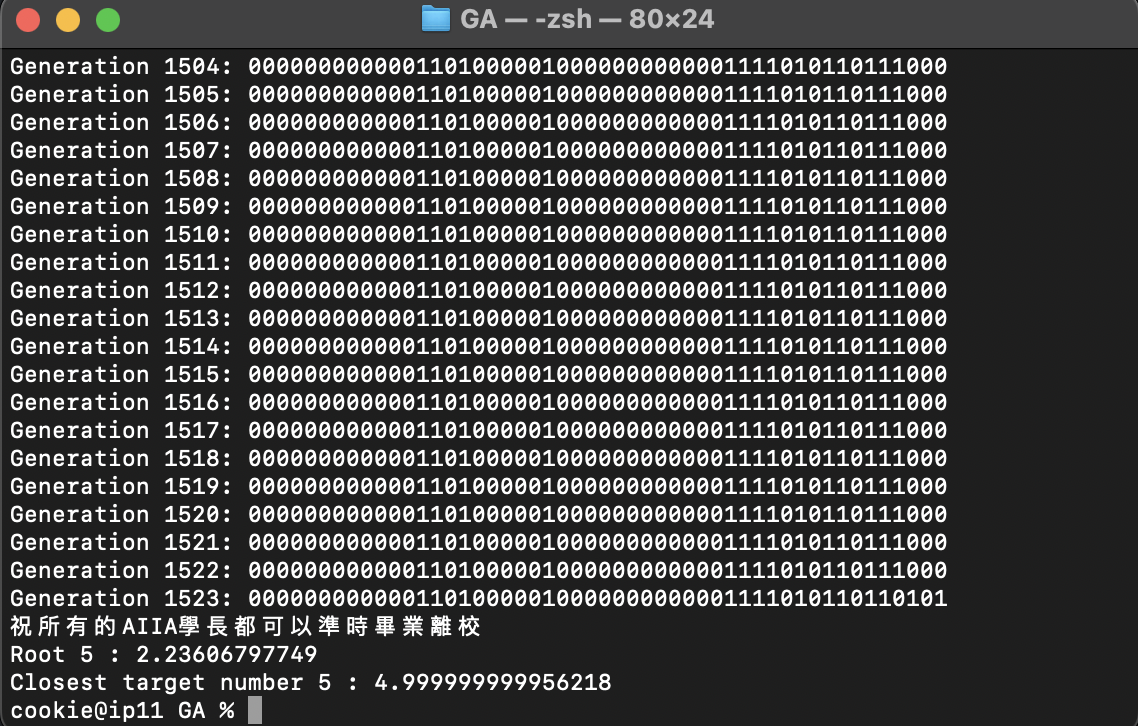
* **def genetic\_algorithm()**

遺傳演算法主要的函數，會先生成一個族群，然後將族群按照適應的程度去排序，產生最佳的染色體，這邊也會去選擇父母的染色體，進行交叉遺傳，然後變異，將好的子# 檢查是否達到停止條件，子代留下來，產生新的族群

（以下為執行結果）



（圖一）可以輸入需要計算的根號數字



（圖二）為輸入(根號)5所得到最終的結果

（以下為程式碼）

