**Programming with LLM’s**

**Using a language model to write a program to analyze sentiments**

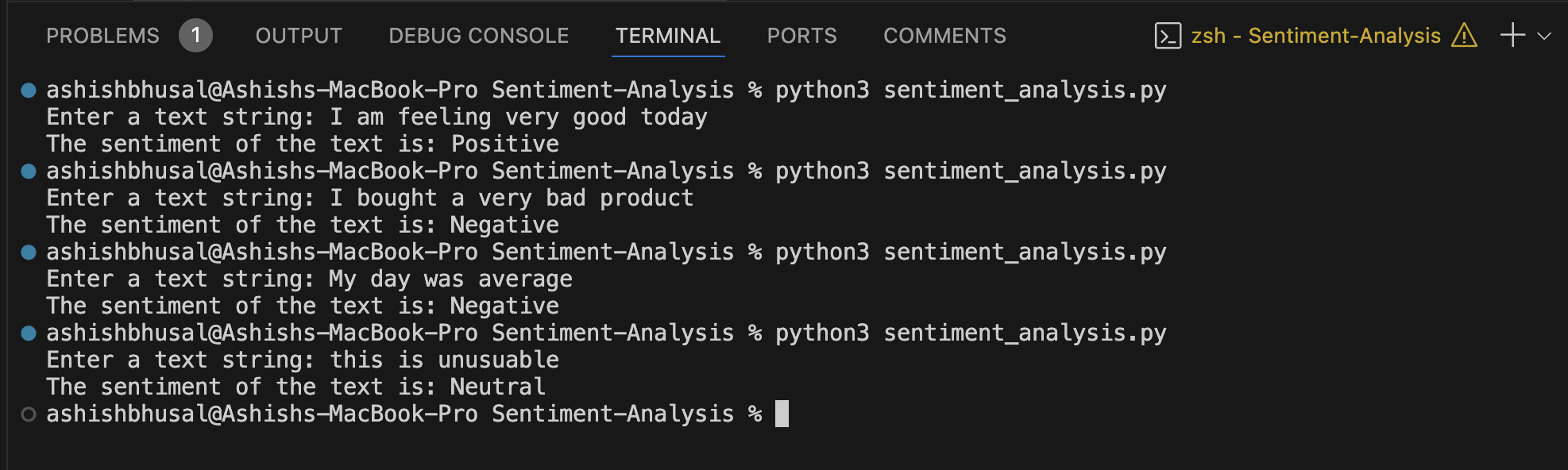
### **Sentiment Analysis Code Generated by ChatGPT**



## **Test Code**



### **Local Testing**



## **Links**

* [**ChatGPT Prompts link**](https://chatgpt.com/share/67a198ed-a860-8008-a36f-f717a1854804)
* [**GitHub Repo Link**](https://github.com/bhusalashish/Sentiment-Analysis)

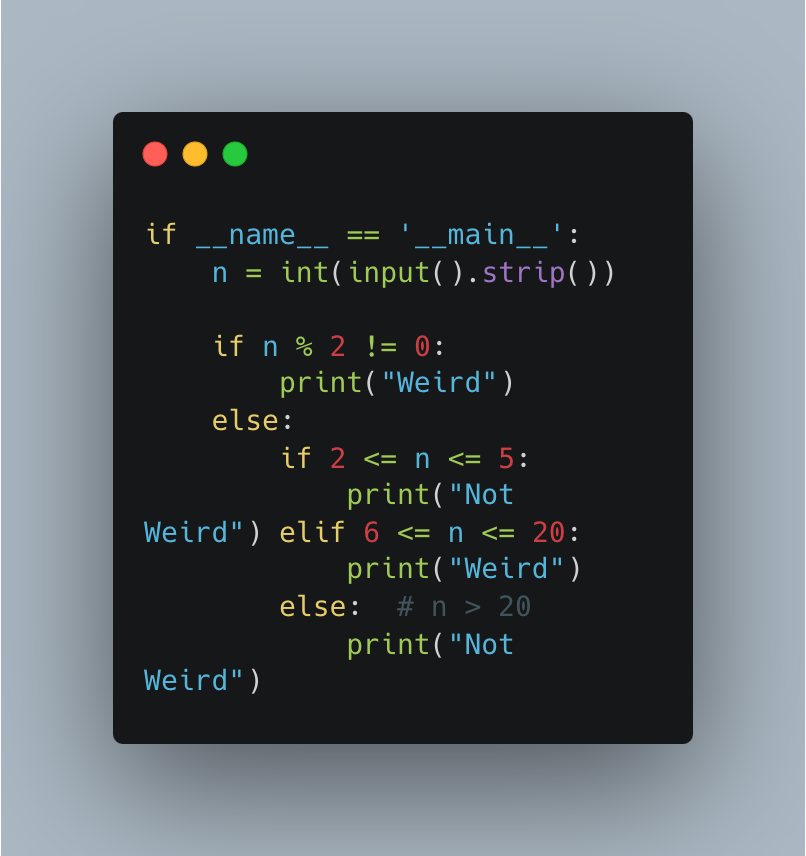
### **HackerRank Problems Promts**

### **Hackerrank Easy**

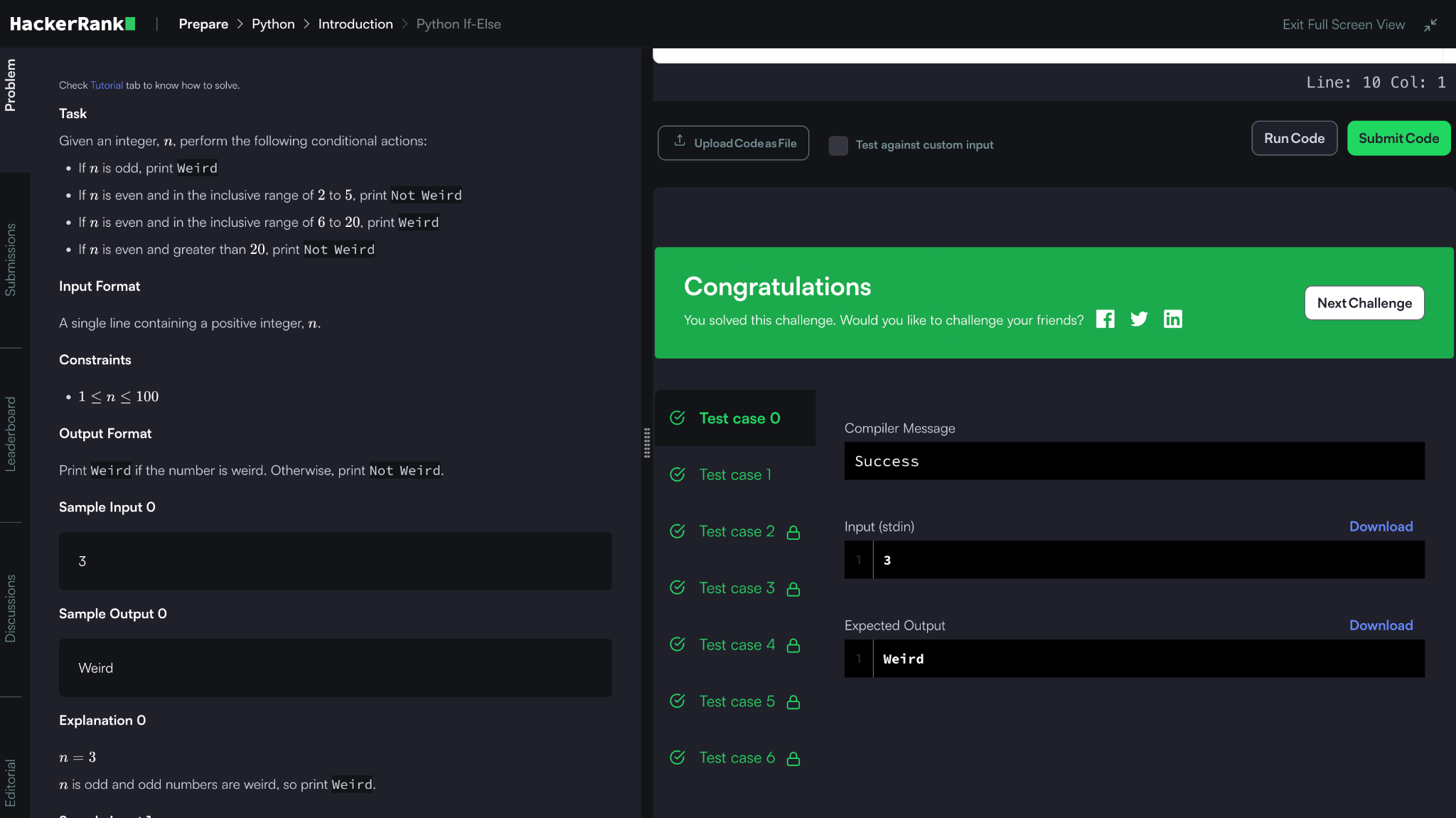
[https://www.hackerrank.com/challenges/py-if-else/problem](https://chatgpt.com/share/67a19d0b-2c40-8008-a1e1-9fde2edcaa9d)

[ChatGPT Prompt](https://chatgpt.com/share/67a19d0b-2c40-8008-a1e1-9fde2edcaa9d)

**Code Generated by ChatGPT**



**HackerRank Submission**



### **2. Medium**

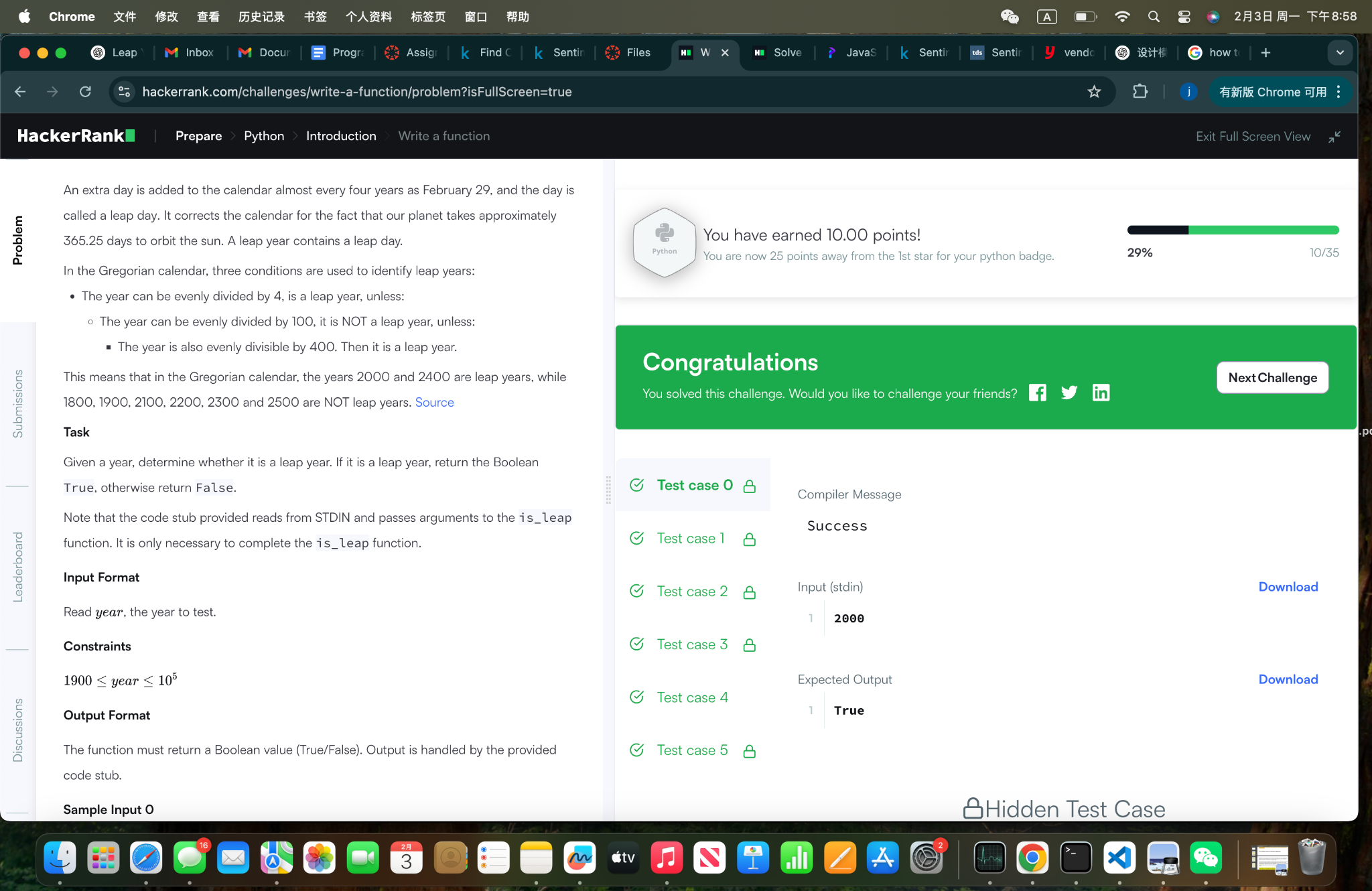
[https://www.hackerrank.com/challenges/py-if-else/problem](https://chatgpt.com/share/67a19d0b-2c40-8008-a1e1-9fde2edcaa9d)

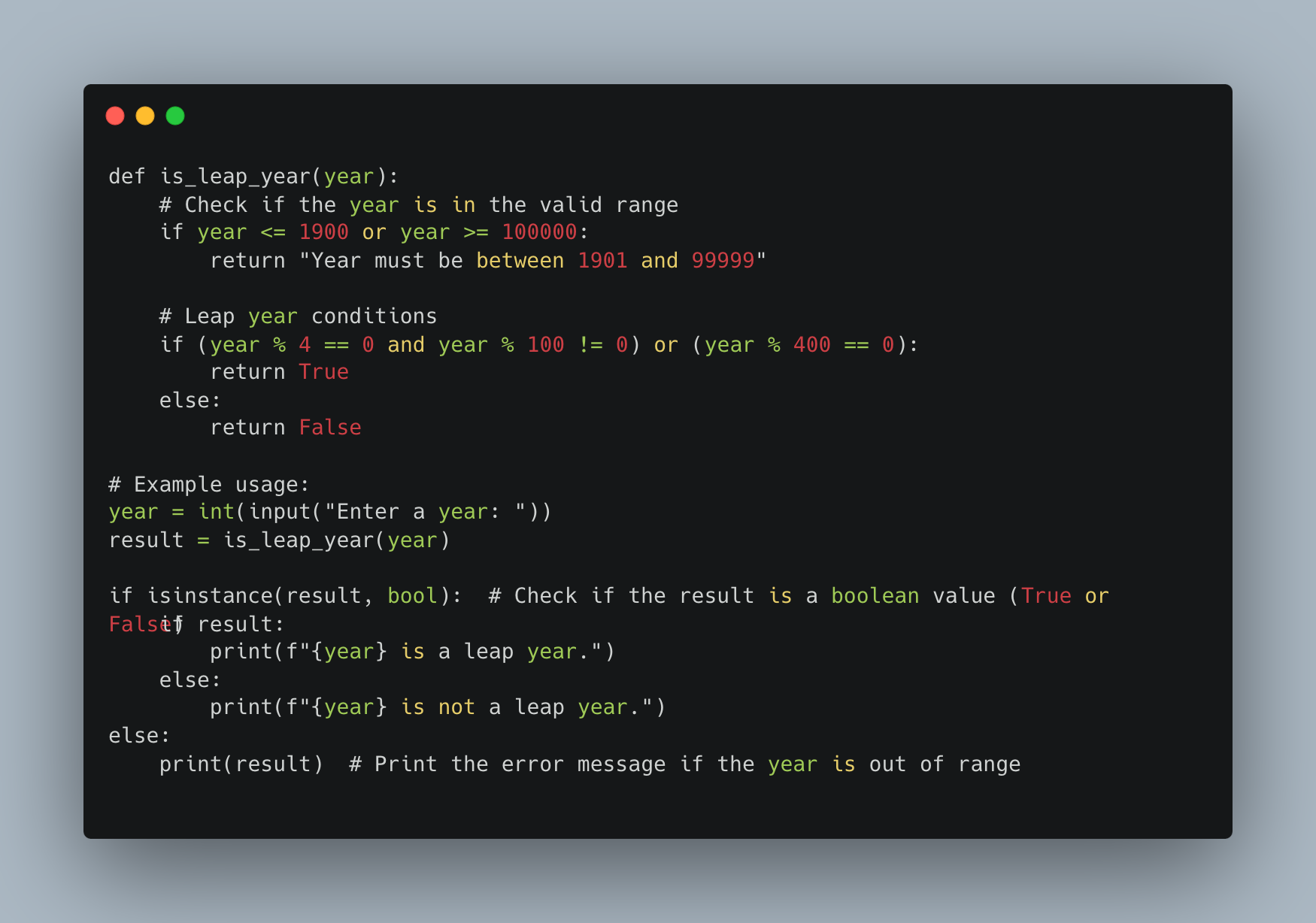
[ChatGPT Prompt link](https://chatgpt.com/share/67a19e2f-06e4-8004-a887-e94d982309d0)

Code Generated by ChatGPT:



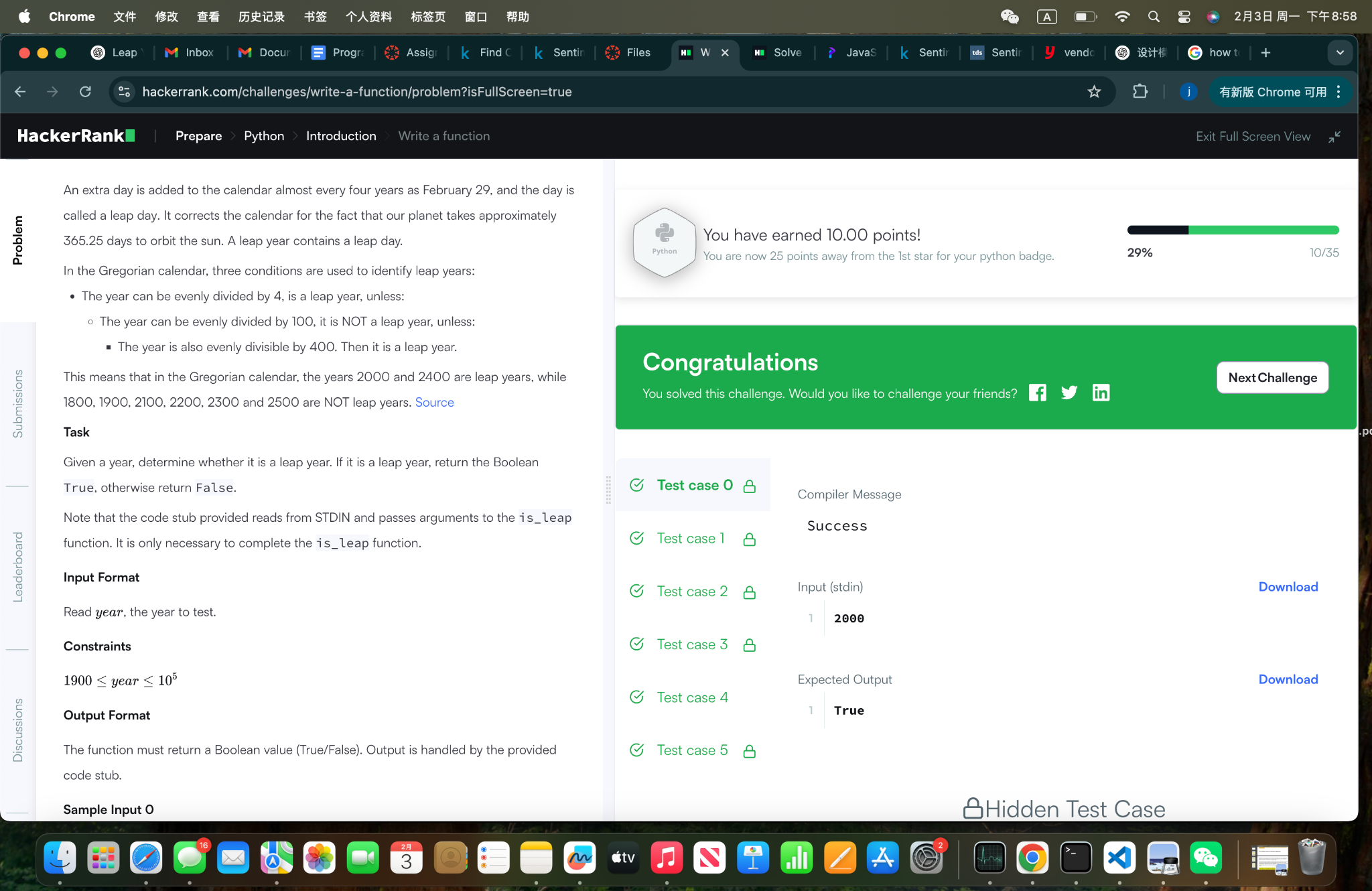
**Result**



Detailed prompt version by ChatGPT

Chatgpt link :

<https://chatgpt.com/share/67a19e2f-06e4-8004-a887-e94d982309d0>

result:

### 

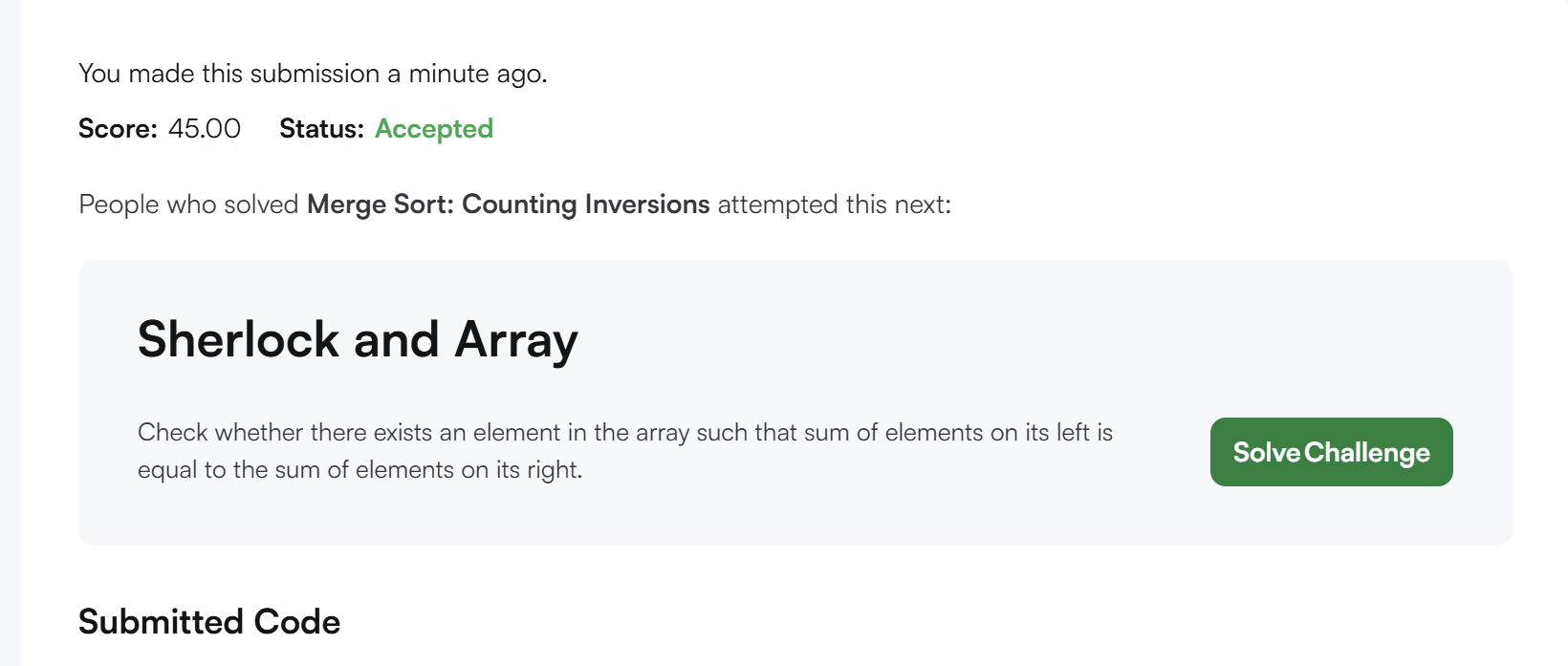
### **3. Hard (Merge Sort: Counting Inversions)**

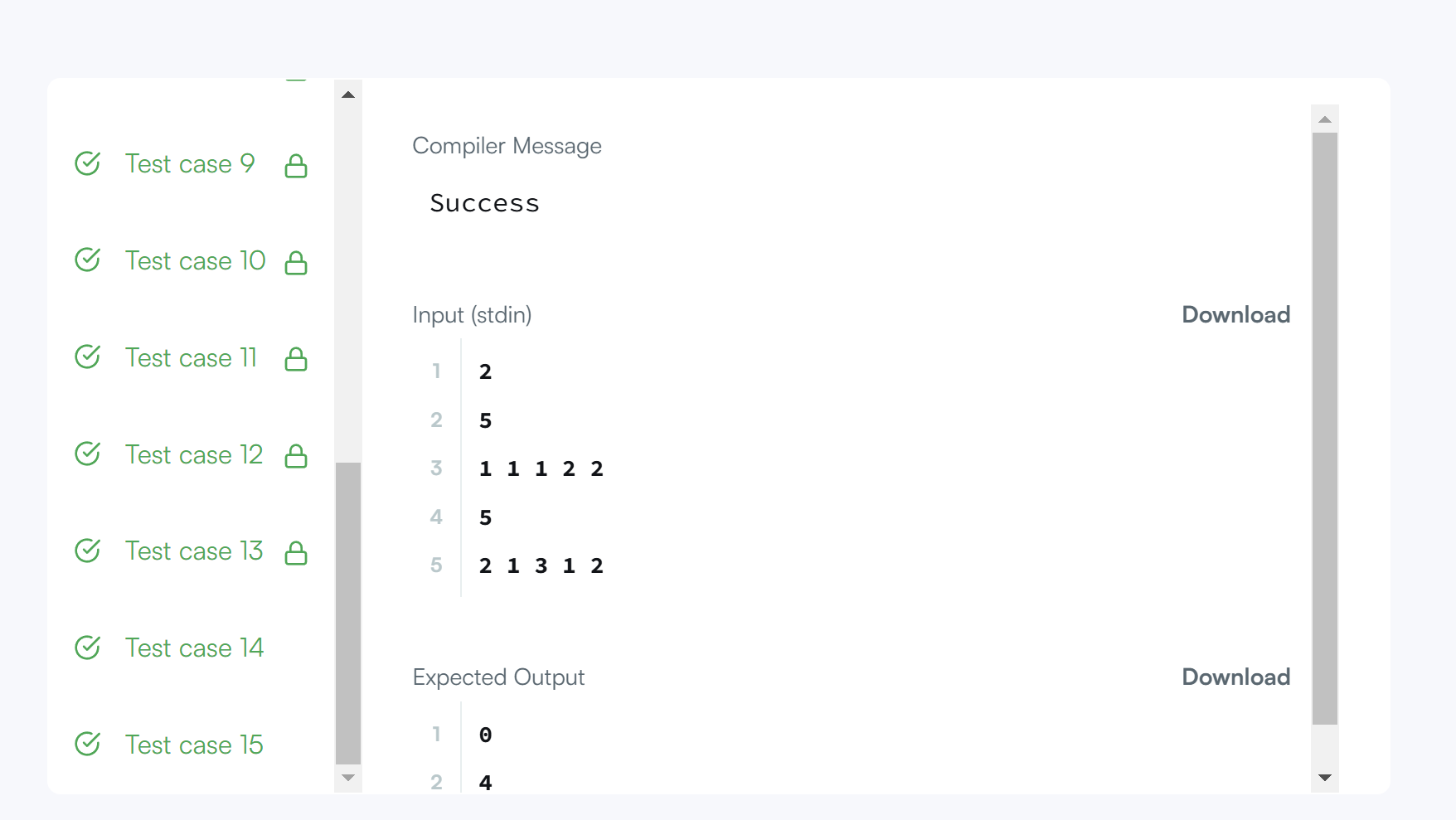
<https://chatgpt.com/share/67a19ced-582c-8012-a32e-500dd19c7581>  
<https://www.hackerrank.com/challenges/ctci-merge-sort/problem>

**Code Generated by ChatGPT**



Result:





References:

[ChatGPT](https://chatgpt.com/)

[Carbon](https://carbon.now.sh/?bg=rgba%28171%2C+184%2C+195%2C+1%29&t=seti&wt=none&l=auto&width=680&ds=true&dsyoff=20px&dsblur=68px&wc=true&wa=true&pv=56px&ph=56px&ln=false&fl=1&fm=Hack&fs=14px&lh=133%25&si=false&es=2x&wm=false&code=def%2520merge_count%28left%252C%2520right%29%253A%250A%2520%2520%2520%2520%2522%2522%2522%250A%2520%2520%2520%2520Merges%2520two%2520sorted%2520lists%2520%27left%27%2520and%2520%27right%27%2520counting%2520the%2520inversions.%250A%2520%2520%2520%2520An%2520inversion%2520occurs%2520when%2520an%2520element%2520from%2520%27right%27%2520is%2520placed%2520before%250A%2520%2520%2520%2520an%2520element%2520from%2520%27left%27%2520%28since%2520all%2520remaining%2520elements%2520in%2520left%2520are%2520larger%29.%250A%2520%2520%2520%2520%2522%2522%2522%250A%2520%2520%2520%2520i%2520%253D%2520j%2520%253D%25200%250A%2520%2520%2520%2520merged%2520%253D%2520%255B%255D%250A%2520%2520%2520%2520inversions%2520%253D%25200%250A%2520%2520%2520%2520while%2520i%2520%253C%2520len%28left%29%2520and%2520j%2520%253C%2520len%28right%29%253A%250A%2520%2520%2520%2520%2520%2520%2520%2520%2523%2520No%2520inversion%252C%2520because%2520left%255Bi%255D%2520is%2520smaller%2520or%2520equal%250A%2520%2520%2520%2520%2520%2520%2520%2520if%2520left%255Bi%255D%2520%253C%253D%2520right%255Bj%255D%253A%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520merged.append%28left%255Bi%255D%29%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520i%2520%252B%253D%25201%250A%2520%2520%2520%2520%2520%2520%2520%2520else%253A%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2523%2520All%2520remaining%2520elements%2520in%2520left%2520%28from%2520i%2520onward%29%2520are%2520greater%2520than%2520right%255Bj%255D%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520merged.append%28right%255Bj%255D%29%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520inversions%2520%252B%253D%2520%28len%28left%29%2520-%2520i%29%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520j%2520%252B%253D%25201%250A%250A%2520%2520%2520%2520%2523%2520Append%2520any%2520remaining%2520elements%250A%2520%2520%2520%2520merged.extend%28left%255Bi%253A%255D%29%250A%2520%2520%2520%2520merged.extend%28right%255Bj%253A%255D%29%250A%2520%2520%2520%2520return%2520merged%252C%2520inversions%250A%250Adef%2520merge_sort_count%28arr%29%253A%250A%2520%2520%2520%2520%2522%2522%2522%250A%2520%2520%2520%2520Recursively%2520sorts%2520the%2520array%2520and%2520returns%2520a%2520tuple%2520%28sorted_array%252C%2520inversion_count%29.%250A%2520%2520%2520%2520%2522%2522%2522%250A%2520%2520%2520%2520if%2520len%28arr%29%2520%253C%253D%25201%253A%250A%2520%2520%2520%2520%2520%2520%2520%2520return%2520arr%252C%25200%250A%250A%2520%2520%2520%2520mid%2520%253D%2520len%28arr%29%2520%252F%252F%25202%250A%2520%2520%2520%2520left%252C%2520left_inv%2520%253D%2520merge_sort_count%28arr%255B%253Amid%255D%29%250A%2520%2520%2520%2520right%252C%2520right_inv%2520%253D%2520merge_sort_count%28arr%255Bmid%253A%255D%29%250A%2520%2520%2520%2520merged%252C%2520merge_inv%2520%253D%2520merge_count%28left%252C%2520right%29%250A%2520%2520%2520%2520total_inv%2520%253D%2520left_inv%2520%252B%2520right_inv%2520%252B%2520merge_inv%250A%2520%2520%2520%2520return%2520merged%252C%2520total_inv%250A%250A%2523%2520Main%2520function%2520to%2520handle%2520input%252Foutput%2520%28HackerRank%2520format%29%250Aif%2520__name__%2520%253D%253D%2520%27__main__%27%253A%250A%2520%2520%2520%2520import%2520sys%250A%2520%2520%2520%2520input%2520%253D%2520sys.stdin.readline%250A%250A%2520%2520%2520%2520t%2520%253D%2520int%28input%28%29.strip%28%29%29%250A%2520%2520%2520%2520for%2520_%2520in%2520range%28t%29%253A%250A%2520%2520%2520%2520%2520%2520%2520%2520n%2520%253D%2520int%28input%28%29.strip%28%29%29%250A%2520%2520%2520%2520%2520%2520%2520%2520arr%2520%253D%2520list%28map%28int%252C%2520input%28%29.split%28%29%29%29%250A%2520%2520%2520%2520%2520%2520%2520%2520_%252C%2520inversion_count%2520%253D%2520merge_sort_count%28arr%29%250A%2520%2520%2520%2520%2520%2520%2520%2520print%28inversion_count%29%250A)