



The first Elf is carrying food with 1000, 2000, and 3000 Calories, a total of 6000 Calories.
The second Elf is carrying one food item with 4000 Calories.
The third Elf is carrying food with 5000 and 6000 Calories, a total of 11000 Calories.
The fourth Elf is carrying food with 7000, 8000, and 9000 Calories, a total of 24000 Calories.
The fifth Elf is carrying one food item with 10000 Calories.
In case the Elves get hungry and need extra snacks, they need to know which Elf to ask: they'd like to know how many Calories are being carried by the Elf carrying the most Calories. In the example above, this is 24000 (carried by the fourth Elf).

Find the Elf carrying the most Calories. How many total Calories is that Elf carrying?



To solve this problem, we need to read the input and sum up the Calories for each Elf. We can do this by reading the input line by line and checking if the line is empty. If it is empty, we know that we have reached the end of an Elf's inventory and we can calculate the total Calories for that Elf. We can then keep track of the maximum total Calories we have seen so far and the corresponding Elf. Once we have processed all the input, we can print the Elf carrying the most Calories and the number of Calories they are carrying.

Here is some sample Python code that implements this solution:

Copy code

```
# Read the input
calories = []

total = 0
max_total = 0
max_elf = 0

for i, line in enumerate(input_lines):
    # If the line is empty, we have reached the end of an Elf's inventory
    if line.strip() == "":
        # Calculate the total Calories for the current Elf
        calories.append(total)
        # Check if this is the Elf with the most Calories
        if total > max_total:
            max_total = total
            max_elf = i
        total = 0
    else:
        # Add the Calories from the current item to the total
        total += int(line)

# Print the Elf with the most Calories and their total Calories
print("Elf %d is carrying the most Calories: %d" % (max_elf, max_total))
```

We can then use this code to solve the problem for the example input given in the prompt:

Copy code

```
# Example input
input_lines = [ "1000", "2000", "3000", "", "4000", "", "5000", "6000", "", "7000", "8000",
"9000", "", "10000",]
```



Try again