



OPERATING SYSTEMS (CS-329)

COMPLEX ENGINEERING PROBLEM REPORT

GROUP MEMBERS

MAHNOOR ZIA – CS22101
ANOOSHA KHALID – CS22104
LAIBA IQRAR – CS22112
SUBMITTED TO: DR. UROOJ AINUDDIN

MAY 4, 2025
NED UNIVERSITY OF ENGINEERING & TECHNOLOGY

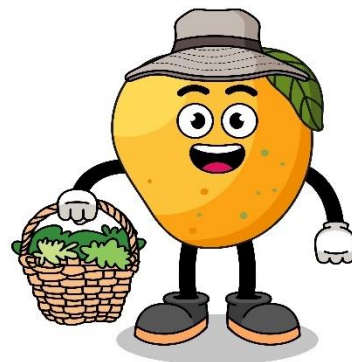
PROBLEM STATEMENT

SPRING WORKERS

Simulate a tree laden with fruits. Launch three “picker” processes and a “loader” process in parallel. If there is fruit on the tree, the picker picks it and places it in a slot of a crate with 12 slots. When a picker finds that the crate is full, it calls the loader. It waits for the loader to place this crate in a truck. Then, the loader furnishes a new crate for the pickers. We assume there is enough space in the truck for all crates. All pickers return to the main function when the tree is bare. In the end, the loader places any partially filled crate in the truck if present. If a picker is adding to the last crate, the loader waits for it to complete the action.

Points to note:

- The number of fruits on the tree is known globally.
- This tree is implemented as an integer array to represent different pieces of fruit.
- The main function provides a shared empty crate when execution starts.
- A piece of fruit can be picked only once and by only one picker for obvious reasons.



CODE OF SPRING WORKERS PROBLEM

```
spring-workers.py x
1 # Group Members: B'22
2 # Mahnoor Zia (CS-22101)
3 # Anooosha Khalid (CS-22104)
4 # Laiba Iqar (CS-22112)
5
6
7 import threading
8 import time
9 import random
10 from datetime import datetime
11
12
13
14 # GLOBAL VARIABLES
15 CRATE_CAPACITY = 12
16 TOTAL_FRUITS = 60
17
18
19
20 # COLORS FOR PRINTING READABILITY
21 COLOR_PINK = "\033[38;5;213m"
22 COLOR_BRIGHT_WHITE = "\033[97m"
23 COLOR_GREEN = "\033[92m"
24 COLOR_BLUE = "\033[94m"
25 COLOR_CYAN = "\033[96m"
26 COLOR_YELLOW = "\033[93m"
27 COLOR_RESET = "\033[0m"
28
29
30
31 # LOGGER FUNCTION FOR TRACKING
32 def log(message, section="", indent=0):
33     timestamp = datetime.now().strftime("%H:%M:%S")
34     section_labels = {
35         "picker": f"\n{COLOR_BLUE}[ PICKER ACTIVITY ]{COLOR_RESET}",
36         "loader": f"\n{COLOR_GREEN}[ LOADER ACTIVITY ]{COLOR_RESET}",
37         "tree": f"\n{COLOR_CYAN}[ FRUIT TREE ]{COLOR_RESET}",
38         "final": f"\n{COLOR_YELLOW}[ FINAL SUMMARY ]{COLOR_RESET}"
39     }
40     label = section_labels.get(section, "")
41
42     if label:
43         print(label)
44     print(f"{' ' * indent}{COLOR_YELLOW}[{timestamp}]{COLOR_RESET} {message}")
45
46
47
48 #SEMAPHORES AND MUTEXES
49 mutex = threading.Lock() # for mutual exclusion
50 semaphore_loader = threading.Semaphore(0) # Loader waits on this until crate is full
51 semaphore_picker = threading.Semaphore(0) # Pickers wait for a new crate after the loader takes the full one.
52
53
54
55 # SHARED RESOURCES
56 tree = list(range(1, TOTAL_FRUITS + 1)) #array
57 crate = []
58 truck = []
59 pickers = 3
60 pickers_in_critical_section = 0
```

```

64 #PICKER THREAD
65 def picker(picker_id):
66     global pickers, pickers_in_critical_section
67     picker_names = {1: "Anoosha", 2: "Laiba", 3: "Mahnoor"}
68     picker_name = picker_names[picker_id]
69
70     while True:
71         mutex.acquire() # semWait(mutex)
72
73         pickers_in_critical_section += 1
74
75         if not tree: # No more fruits left
76             pickers_in_critical_section -= 1
77
78             pickers -= 1
79             if TOTAL_FRUITS == 0:
80                 log("OOPS! No fruits available on the tree :( No need to call the loader.", section="tree")
81                 print(" " * 4 + f"{picker_name} is upset and exiting.")
82             else:
83                 log(f"{picker_name} has finished picking and is waiting for loader to finish.", section="picker", indent=4)
84                 print(" " * 4 + "Tree is bare.")
85
86             semaphore_loader.release() # semSignal(L)
87             mutex.release() # semSignal(mutex)
88             return
89
90
91     if len(crate) == CRATE_CAPACITY:
92         pickers_in_critical_section -= 1
93         mutex.release() # semSignal(mutex)
94
95         semaphore_picker.acquire() # semWait(P)
96         continue
97
98         # Pick a fruit
99         fruit = tree.pop(0)
100         crate.append(fruit)
101         log(f"{picker_name} picked fruit {fruit}.", section="picker", indent=4)
102         print(" " * 4 + f"Current crate size: {len(crate)}/{CRATE_CAPACITY}")
103
104
105         if len(crate) == CRATE_CAPACITY: # Notify loader once crate is full
106             log(f"{picker_name} has filled the crate with {CRATE_CAPACITY} fruits.", section="picker", indent=4)
107             print(" " * 4 + "Found crate full. Notifying loader.")
108             semaphore_loader.release() # semSignal(L)
109
110         pickers_in_critical_section -= 1
111         mutex.release() # semSignal(mutex)
112         time.sleep(random.uniform(0.05, 0.2)) # to alternation of pickers
113
114
115
116 # LOADER THREAD
117 def loader():
118     while True:
119         semaphore_loader.acquire() # semWait(L)
120         mutex.acquire() # semWait(mutex)
121
122         # Check if the crate is full
123         if len(crate) == CRATE_CAPACITY:
124             log("Loader triggered! Crate is full.", section="loader", indent=2)
125             print(" " * 4 + "Loading it to truck...")
126             truck.append(crate[:])
127             crate.clear()
128
129             # It will notify pickers that they can start working on a new crate
130             for _ in range(pickers):
131                 semaphore_picker.release() # semSignal(P)
132
133             mutex.release() # semSignal(mutex)
134             continue
135
136
137         # If all pickers are done and there's a partial crate
138         if pickers == 0 and pickers_in_critical_section == 0 and crate:
139             log("Loader detected partially filled crate after pickers finished.", section="loader", indent=2)
140             print(" " * 4 + "Loader is moving the final partial crate to the truck.")
141             truck.append(crate[:])
142             crate.clear()
143             if TOTAL_FRUITS == 0:

```

```

143         if TOTAL_FRUITS == 0:
144             return
145         else:
146             log("Loader has completed all operations and is exiting.", section="loader", indent=2)
147             mutex.release() # semSignal(mutex)
148             return
149
150
151     # If no pickers left and no crate, finish
152     if pickers == 0 and pickers_in_critical_section == 0 and not crate:
153         if TOTAL_FRUITS == 0:
154             return
155         else:
156             log("Loader has completed all operations and is exiting.", section="loader", indent=2)
157             mutex.release() # semSignal(mutex)
158             return
159
160     mutex.release() # semSignal(mutex)
161
162
163

```

```

164 #MAIN
165 def main():
166     print("\n|_____|")
167     print(f"          {COLOR_PINK} SPRING WORKERS SIMULATION START {COLOR_RESET}")
168     print("|_____|")
169
170     if TOTAL_FRUITS < 0: # if enter the neg no for fruits
171         print("\nOOPS! Fruits can't be negative.\nEnter the accurate details pls.")
172         print("Exiting the simulation...")
173         return
174
175     print("\nYay! Mango season has started, it's time to pluck the mangoes from the tree!")
176     print("Pickers: 1 - Anoosha | 2 - Laiba | 3 - Mahnoor\n")
177
178     for i in range(0, len(tree), 10): # for printing the tree
179         print(" " * 4 + ' '.join(map(str, tree[i:i + 10])))
180
181
182     #creating threads
183     picker_threads = [threading.Thread(target=picker, args=(i,)) for i in range(1, 4)]
184     loader_thread = threading.Thread(target=loader)
185
186
187     # start the both picker and loader to finish their work
188     for t in picker_threads:
189         t.start()
190     loader_thread.start()
191
192
193     # wait for both picker and loader to finish their work
194     for t in picker_threads:
195         t.join()
196     loader_thread.join()
197
198     log("", section="final")
199
200
201     # printing the summary
202     if TOTAL_FRUITS > 0:
203         print("\nCrate in the Truck:")
204         for index, crate in enumerate(truck, 1):
205             print(f"[ Crate {index} ]")
206
207
208         print(f"{COLOR_GREEN}|_____|{COLOR_RESET}")
209         for i in range(0, len(crate), 12):
210             fruits_row = ' '.join(map(str, crate[i:i+12]))
211             print(f" {fruits_row:<40} ")
212             print(f"{COLOR_GREEN}|_____|{COLOR_RESET}")
213             print(f"({len(crate)} fruits)")
214
215
216     print(f"\nTotal crates loaded: {len(truck)}")
217     print(f"{COLOR_GREEN}Spring harvest has been successfully completed. Thank you, workers!{COLOR_RESET}\n")
218
219     main()
220

```

TEST CASES

CASE 1:

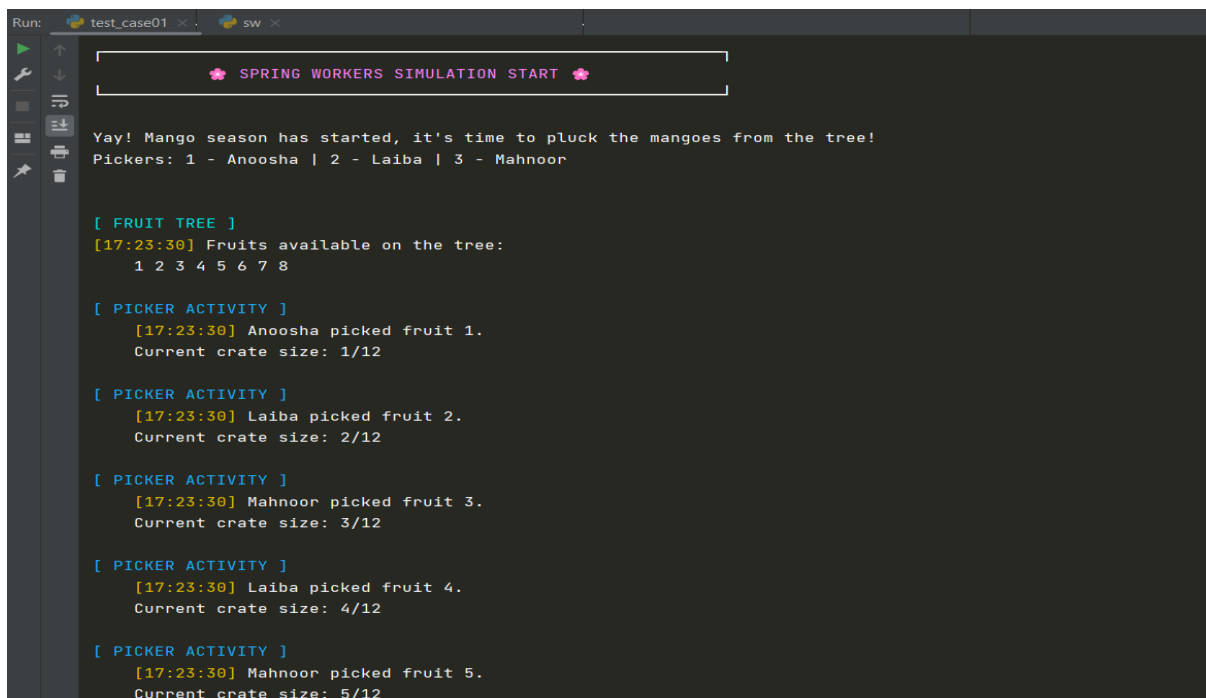
[Checking the partial crate logic]

- TOTAL_FRUITS = 8
- CRATE_CAPACITY = 12
- pickers = 3

Expected output: Since there are fewer than 12 fruits, the crate will be partially filled with 8 fruits. After all pickers finish, the loader will move the partial crate to the truck, completing the task as expected.

STATUS PASSED!

Actual Output:



```
Run: test_case01 x sw x

  🌸 SPRING WORKERS SIMULATION START 🌸

Yay! Mango season has started, it's time to pluck the mangoes from the tree!
Pickers: 1 - Anoosha | 2 - Laiba | 3 - Mahnoor

[ FRUIT TREE ]
[17:23:30] Fruits available on the tree:
  1 2 3 4 5 6 7 8

[ PICKER ACTIVITY ]
[17:23:30] Anoosha picked fruit 1.
Current crate size: 1/12

[ PICKER ACTIVITY ]
[17:23:30] Laiba picked fruit 2.
Current crate size: 2/12

[ PICKER ACTIVITY ]
[17:23:30] Mahnoor picked fruit 3.
Current crate size: 3/12

[ PICKER ACTIVITY ]
[17:23:30] Laiba picked fruit 4.
Current crate size: 4/12

[ PICKER ACTIVITY ]
[17:23:30] Mahnoor picked fruit 5.
Current crate size: 5/12
```

```
test_case01 x sw x
[ PICKER ACTIVITY ]
[17:23:30] Anoosha picked fruit 6.
Current crate size: 6/12

[ PICKER ACTIVITY ]
[17:23:30] Mahnoor picked fruit 7.
Current crate size: 7/12

[ PICKER ACTIVITY ]
[17:23:30] Laiba picked fruit 8.
Current crate size: 8/12

[ PICKER ACTIVITY ]
[17:23:30] Anoosha has finished picking and is exiting.
Tree is bare.

[ PICKER ACTIVITY ]
[17:23:30] Mahnoor has finished picking and is exiting.
Tree is bare.

[ PICKER ACTIVITY ]
[17:23:30] Laiba has finished picking and is exiting.
Tree is bare.

[ LOADER ACTIVITY ]
[17:23:30] Loader detected partially filled crate after pickers finished.
Loader is moving the final partial crate to the truck.

[ LOADER ACTIVITY ]
[17:23:30] Loader has completed all operations and is exiting.
```

```
[ FINAL SUMMARY ]
[17:23:30]

Crates in the Truck:

[ Crate 1 ]
1 2 3 4 5 6 7 8
(8 fruits)

Total crates loaded: 1
Spring harvest has been successfully completed. Thank you, workers!

Process finished with exit code 0
```

CASE 2: [Checking the full crate logic]

- TOTAL_FRUITS = 12
- CRATE_CAPACITY = 12
- pickers = 3

Expected output: Pickers (3) fill crates in parallel, and each crate is moved to the truck once it reaches 12 fruits. No partial crate will be detected.

STATUS PASSED!

Actual Output:

```
↑
↓
[ SPRING WORKERS SIMULATION START ]

Yay! Mango season has started, it's time to pluck the mangoes from the tree!
Pickers: 1 - Anoosha | 2 - Laiba | 3 - Mahnoor

[ FRUIT TREE ]
[19:27:15] Fruits available on the tree:
  1 2 3 4 5 6 7 8 9 10
 11 12

[ PICKER ACTIVITY ]
[19:27:15] Anoosha picked fruit 1.
Current crate size: 1/12

[ PICKER ACTIVITY ]
[19:27:15] Laiba picked fruit 2.
Current crate size: 2/12

[ PICKER ACTIVITY ]
[19:27:15] Mahnoor picked fruit 3.
Current crate size: 3/12

[ PICKER ACTIVITY ]
[19:27:15] Laiba picked fruit 4.
Current crate size: 4/12

[ PICKER ACTIVITY ]
[19:27:15] Anoosha picked fruit 5.
Current crate size: 5/12

[ PICKER ACTIVITY ]
[19:27:15] Mahnoor picked fruit 6.
Current crate size: 6/12

[ PICKER ACTIVITY ]
[19:27:15] Laiba picked fruit 7.
Current crate size: 7/12

[ PICKER ACTIVITY ]
[19:27:15] Anoosha picked fruit 8.
Current crate size: 8/12

[ PICKER ACTIVITY ]
[19:27:15] Laiba picked fruit 9.
Current crate size: 9/12

[ PICKER ACTIVITY ]
[19:27:15] Mahnoor picked fruit 10.
Current crate size: 10/12

[ PICKER ACTIVITY ]
[19:27:15] Laiba picked fruit 11.
Current crate size: 11/12
```



```

[ PICKER ACTIVITY ]
[19:27:15] Mahnoor picked fruit 12.
Current crate size: 12/12

[ PICKER ACTIVITY ]
[19:27:15] Mahnoor has filled the crate with 12 fruits.
Found crate full. Notifying loader.

[ LOADER ACTIVITY ]
[19:27:15] Loader triggered! Crate is full.
Loading it to truck...

[ PICKER ACTIVITY ]
[19:27:15] Anoosha has finished picking and is waiting for loader to finish.
Tree is bare.

[ PICKER ACTIVITY ]
[19:27:15] Laiba has finished picking and is waiting for loader to finish.
Tree is bare.

[ PICKER ACTIVITY ]
[19:27:15] Mahnoor has finished picking and is waiting for loader to finish.
Tree is bare.

[ LOADER ACTIVITY ]
[19:27:16] Loader has completed all operations and is exiting.

[ LOADER ACTIVITY ]
[19:27:16] Loader has completed all operations and is exiting.

[ FINAL SUMMARY ]
[19:27:16]

Crates in the Truck:

[ Crate 1 ]
1 2 3 4 5 6 7 8 9 10 11 12
(12 fruits)

Total crates loaded: 1
Spring harvest has been successfully completed. Thank you, workers!

Process finished with exit code 0

```

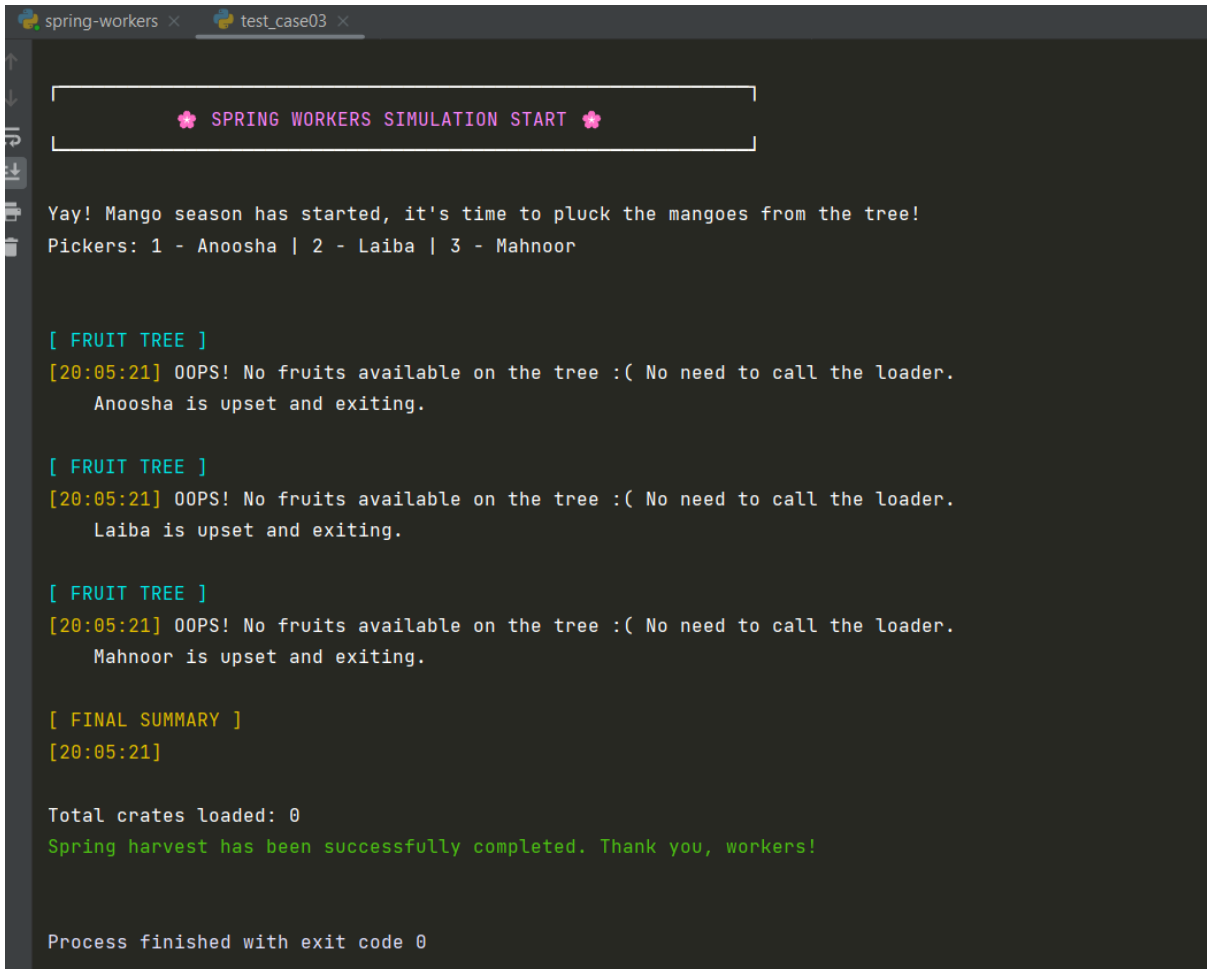
CASE 3: [if there's no fruit on the tree]

- TOTAL_FRUITS = 0
- CRATE_CAPACITY = 12
- pickers = 3

Expected output: Pickers will immediately finish and will not call the loader since there are no fruits.

STATUS PASSED!

Actual Output:



```
spring-workers x test_case03 x

[ SPRING WORKERS SIMULATION START ]

Yay! Mango season has started, it's time to pluck the mangoes from the tree!
Pickers: 1 - Anoosha | 2 - Laiba | 3 - Mahnoor

[ FRUIT TREE ]
[20:05:21] OOPS! No fruits available on the tree :( No need to call the loader.
      Anoosha is upset and exiting.

[ FRUIT TREE ]
[20:05:21] OOPS! No fruits available on the tree :( No need to call the loader.
      Laiba is upset and exiting.

[ FRUIT TREE ]
[20:05:21] OOPS! No fruits available on the tree :( No need to call the loader.
      Mahnoor is upset and exiting.

[ FINAL SUMMARY ]
[20:05:21]

Total crates loaded: 0
Spring harvest has been successfully completed. Thank you, workers!

Process finished with exit code 0
```

CASE 4:

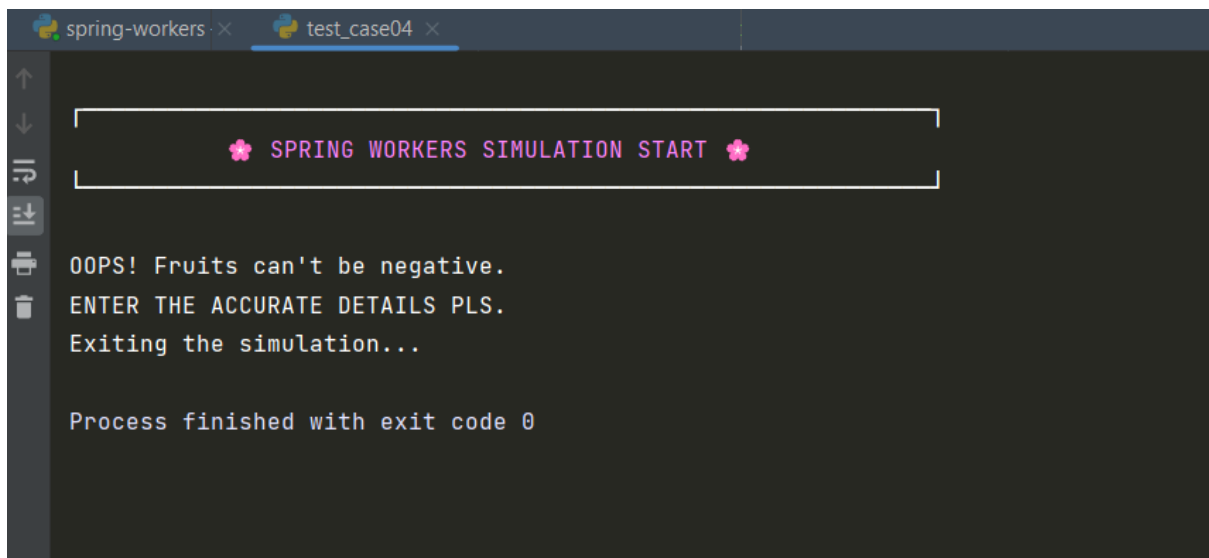
[if fruit count is in negative number]

- TOTAL_FRUITS = -5
- CRATE_CAPACITY = 12
- pickers = 3

Expected output: Immediate exit.

STATUS PASSED!

Actual Output:



```
spring-workers × test_case04 ×  
[  
  ✿ SPRING WORKERS SIMULATION START ✿  
]  
OOOPS! Fruits can't be negative.  
ENTER THE ACCURATE DETAILS PLS.  
Exiting the simulation...  
  
Process finished with exit code 0
```