# Bài 5: Thực hành kiểu dữ liệu ngăn xếp và hàng đợi

**Giải Chuyên đề Tin học 12 Bài 5: Thực hành kiểu dữ liệu ngăn xếp và hàng đợi**  
**Khởi động trang 20 Chuyên đề Tin học 12**: Trong bài trước, chúng ta đã sử dụng kiểu dữ liệu hàng đợi và ngăn xếp. Trong nhiều trường hợp ứng dụng trong thực tế chúng ta phải kết hợp cả hai loại dữ liệu này. Em có thể nêu được một ví dụ cần sử dụng cả hai kiểu dữ liệu này không?  
**Lời giải:**  
Ví dụ thực tế cần sử dụng cả kiểu dữ liệu hàng đợi (queue) và ngăn xếp (stack) là trong việc thực hiện kiểm tra biểu thức số học (arithmetic expression) để xem liệu biểu thức có được viết đúng (well-formed) hay không, đặc biệt là kiểm tra tính đúng đắn của dấu ngoặc.  
Ví dụ: Kiểm tra tính đúng đắn của dấu ngoặc trong biểu thức  
Giả sử có một biểu thức toán học và bạn muốn kiểm tra xem tất cả các dấu ngoặc mở đều có dấu ngoặc đóng tương ứng và được đặt đúng thứ tự. Để làm điều này, bạn có thể sử dụng ngăn xếp và hàng đợi như sau:  
1. Ngăn xếp (Stack): Dùng để theo dõi các dấu ngoặc mở.  
2. Hàng đợi (Queue): Dùng để lưu trữ biểu thức đã phân tích và kiểm tra trong trường hợp bạn cần lưu trữ và xử lý lại sau khi kiểm tra dấu ngoặc.  
Cách thực hiện như sau:  
1. Duyệt qua từng ký tự trong biểu thức:  
- Nếu gặp dấu ngoặc mở ((, [, {), đưa vào ngăn xếp.  
- Nếu gặp dấu ngoặc đóng (), ], }), kiểm tra ngăn xếp để đảm bảo dấu ngoặc đóng tương ứng với dấu ngoặc mở tương ứng. Nếu đúng, loại bỏ dấu ngoặc mở khỏi ngăn xếp.  
2. Sau khi duyệt qua tất cả các ký tự:  
- Nếu ngăn xếp trống, biểu thức có dấu ngoặc đúng.  
- Nếu ngăn xếp không trống, biểu thức có dấu ngoặc không đúng.  
**Luyện tập 1 trang 22 Chuyên đề Tin học 12**: Hãy chạy chương trình với nhiều trường hợp dữ liệu đầu vào khác nhau. Em có nhận xét gì về vị trí của những người phải đổi món ăn?  
**Lời giải:**  
Đầu tiên, viết chương trình để đọc dữ liệu từ hai tệp input1.inpvà input2.inp, sau đó mô phỏng quá trình phục vụ suất ăn. Dưới đây là gợi ý chương trình hoàn chỉnh mẫu:  
from queue import Queue  
from collections import deque  
# Định nghĩa lớp Stack sử dụng deque cho ngăn xếp  
class Stack:  
 def \_\_init\_\_(self):  
 self.items = deque()  
 def is\_empty(self):  
 return len(self.items) == 0  
 def push(self, item):  
 self.items.append(item)  
 def pop(self):  
 if not self.is\_empty():  
 return self.items.pop()  
 return None  
 def size(self):  
 return len(self.items)  
def bepan(dangkiga, dangkibo, suatan):  
 doimon = [] # list chứa ID những người phải đổi món  
 while not suatan.is\_empty():  
 tmp = suatan.pop()  
 if tmp == "bò": # nếu suất ăn lấy ra là bò  
 if not dangkibo.empty(): # còn người đăng kí suất ăn bò  
 dangkibo.get()  
 else:  
 ID\_doi = dangkiga.get()  
 doimon.append(ID\_doi)  
 elif tmp == "gà": # nếu suất ăn lấy ra là gà  
 if not dangkiga.empty(): # còn người đăng kí suất ăn gà  
 dangkiga.get()  
 else:  
 ID\_doi = dangkibo.get()  
 doimon.append(ID\_doi)  
 return doimon  
# Khởi tạo hàng đợi cho cơm gà và cơm bò  
dangkiga = Queue()  
dangkibo = Queue()  
# Khởi tạo ngăn xếp cho các suất ăn  
suatan = Stack()  
# Đọc thông tin đăng kí và đưa vào 2 queue  
with open("input1.inp", "r", encoding="utf8") as file1:  
 for line in file1.readlines():  
 id, dangki = line.strip().split()  
 if dangki == "gà":  
 dangkiga.put(id)  
 elif dangki == "bò":  
 dangkibo.put(id)  
# Đọc thông tin suất ăn và đưa vào ngăn xếp  
with open("input2.inp", "r", encoding="utf8") as file2:  
 data = file2.read().strip().split()  
 for item in data:  
 suatan.push(item)  
doimon = bepan(dangkiga, dangkibo, suatan)  
if len(doimon) == 0:  
 print("Không có người nào phải đổi món ăn")  
else:  
 print("Danh sách những người phải đổi món là:", doimon)  
- Nhận xét về vị trí của những người phải đổi món ăn  
Sau khi chạy chương trình với nhiều trường hợp dữ liệu đầu vào khác nhau, nhận xét về vị trí của những người phải đổi món ăn:  
- Những người phải đổi món ăn luôn nằm trong hàng đợi ban đầu của loại suất ăn còn lại khi loại suất ăn họ đăng ký đã được phục vụ hết.  
- Vị trí của những người phải đổi món ăn phụ thuộc vào thứ tự của họ trong hàng đợi. Người đầu tiên trong hàng đợi sẽ phải đổi món trước khi đến lượt họ nếu hàng đợi của suất ăn còn lại đã hết.  
**Luyện tập 2 trang 22 Chuyên đề Tin học 12**: Hãy viết chương trình tính toán thời gian chờ đợi trung bình của mỗi người để nhận được suất ăn của mình, cho biết thời gian lấy một suất ăn ra khỏi ngăn xếp và đưa cho người lao động là 1 giây.  
**Lời giải:**  
Để tính toán thời gian chờ đợi trung bình cần theo dõi thời gian chờ của mỗi người. Dưới đây là chương trình mở rộng để tính toán thời gian chờ đợi trung bình.  
def bepan\_with\_wait\_time(dangkiga, dangkibo, suatan):  
 doimon = [] # list chứa ID những người phải đổi món  
 wait\_times = {}  
 current\_time = 0  
 while not suatan.is\_empty():  
 current\_time += 1 # Mỗi lượt lấy suất ăn là 1 giây  
 tmp = suatan.pop()  
 if tmp == "bò": # nếu suất ăn lấy ra là bò  
 if not dangkibo.empty(): # còn người đăng kí suất ăn bò  
 id = dangkibo.get()  
 wait\_times[id] = current\_time  
 else:  
 id = dangkiga.get()  
 doimon.append(id)  
 wait\_times[id] = current\_time  
 elif tmp == "gà": # nếu suất ăn lấy ra là gà  
 if not dangkiga.empty(): # còn người đăng kí suất ăn gà  
 id = dangkiga.get()  
 wait\_times[id] = current\_time  
 else:  
 id = dangkibo.get()  
 doimon.append(id)  
 wait\_times[id] = current\_time  
 return doimon, wait\_times  
# Đọc thông tin đăng kí và đưa vào 2 queue  
dangkiga = Queue()  
dangkibo = Queue()  
with open("input1.inp", "r", encoding="utf8") as file1:  
 for line in file1.readlines():  
 id, dangki = line.strip().split()  
 if dangki == "gà":  
 dangkiga.put(id)  
 elif dangki == "bò":  
 dangkibo.put(id)  
# Đọc thông tin suất ăn và đưa vào ngăn xếp  
suatan = Stack()  
with open("input2.inp", "r", encoding="utf8") as file2:  
 data = file2.read().strip().split()  
 for item in data:  
 suatan.push(item)  
doimon, wait\_times = bepan\_with\_wait\_time(dangkiga, dangkibo, suatan)  
if len(doimon) == 0:  
 print("Không có người nào phải đổi món ăn")  
else:  
 print("Danh sách những người phải đổi món là:", doimon)  
# Tính toán thời gian chờ đợi trung bình  
if wait\_times:  
 total\_wait\_time = sum(wait\_times.values())  
 average\_wait\_time = total\_wait\_time / len(wait\_times)  
 print(f"Thời gian chờ đợi trung bình là: {average\_wait\_time:.2f} giây")  
else:  
 print("Không có thời gian chờ đợi nào được tính toán.")  
**Vận dụng trang 22 Chuyên đề Tin học 12**: Bài toán nhà bếp được thay đổi như sau:  
Yêu cầu người lao động xếp thành một hàng để nhận đồ ăn, trong đó những người đăng kí suất cơm gà và những người đăng kí suất cơm bò sẽ đứng lẫn với nhau. Các suất ăn vẫn được nhà bếp thực hiện và đưa vào một ngăn xếp để phục vụ người lao động.   
*Quy tắc chọn suất ăn như sau:* Mỗi người đến lượt sẽ nhận suất ăn được đưa ra từ ngăn xếp, nếu suất ăn đó đúng với suất ăn đã đăng kí thì người này sẽ được nhận suất ăn và đưa ra khỏi hàng đợi. Ngược lại nếu suất ăn không đúng với đăng kí thì người này sẽ ra khỏi hàng đợi và xếp lại vào cuối hàng và tiếp tục đợi. Quá trình chọn suất ăn như trên sẽ dừng lại nếu tất cả số người xếp hàng đều nhận được suất ăn đúng của mình hoặc tất cả mọi người trong hàng đợi đều không thể nhận được suất ăn như đã đăng kí.   
Em hãy viết chương trình đọc hai tệp dữ liệu chứa thông tin về các suất ăn của người lao động và tính số người không nhận được suất ăn của mình.   
Dữ liệu đầu vào gồm hai tệp, tệp input1.inp chứa thông tin về các suất ăn của người lao động trong hàng đợi, tệp input2.inp chứa thông tin về các suất ăn mà nhà ăn đã chuẩn bị và đưa vào trong ngăn xếp. Ví dụ dữ liệu đầu vào và đầu ra như sau:   
  
Dữ liệu đầu ra là một số nguyên cho trước chỉ số lượng người xếp hàng không thể chọn được suất ăn của mình. Trong ví dụ trên kết quả đưa ra là 2.  
**Lời giải:**  
Với bài toán nhà bếp với quy tắc chọn suất ăn mới, cần viết chương trình để thực hiện theo quy tắc mới: mỗi người đến lượt sẽ nhận suất ăn được đưa ra từ ngăn xếp. Nếu suất ăn đó đúng với suất ăn mà họ đã đăng ký, người đó sẽ được nhận suất ăn và đưa ra khỏi hàng đợi. Nếu không, người đó sẽ quay lại cuối hàng đợi. Quá trình này tiếp tục cho đến khi tất cả mọi người đều nhận được suất ăn đúng hoặc tất cả mọi người không thể nhận được suất ăn của mình.  
Gợi ý các bước giải quyết bài toán:  
1. Đọc dữ liệu từ tệp **input1.inp** để lấy danh sách đăng ký suất ăn của người lao động.  
2. Đọc dữ liệu từ tệp **input2.inp** để lấy danh sách các suất ăn đã chuẩn bị.  
3. Sử dụng hàng đợi để mô phỏng quá trình xếp hàng nhận suất ăn.  
4. Sử dụng ngăn xếp để lấy các suất ăn đã chuẩn bị.  
5. Thực hiện quá trình phục vụ suất ăn theo quy tắc mới.  
6. Đưa ra số lượng người không nhận được suất ăn đúng của mình.  
Mẫu mã nguồn gợi ý:  
from collections import deque  
def read\_file(filename):  
 with open(filename, 'r', encoding='utf8') as file:  
 return file.read().strip().split()  
def serve\_food(people\_queue, food\_stack):  
 initial\_queue\_length = len(people\_queue)  
 no\_change\_count = 0  
 served\_count = 0  
 while food\_stack and no\_change\_count < len(people\_queue):  
 person = people\_queue.popleft()  
 food = food\_stack.pop()  
 if person == food:  
 served\_count += 1  
 no\_change\_count = 0 # reset no change count as someone got their food  
 else:  
 people\_queue.append(person)  
 food\_stack.append(food) # put the food back on the stack  
 no\_change\_count += 1  
 return len(people\_queue) - served\_count  
# Đọc dữ liệu từ file  
people = read\_file("input1.inp")  
foods = read\_file("input2.inp")  
# Tạo hàng đợi cho người và ngăn xếp cho suất ăn  
people\_queue = deque(people)  
food\_stack = list(reversed(foods))  
# Tính số lượng người không nhận được suất ăn của mình  
unserved\_people\_count = serve\_food(people\_queue, food\_stack)  
print(unserved\_people\_count)  
Chú thích mã nguồn  
1. Đọc dữ liệu:  
- read\_file(filename): Hàm này đọc nội dung từ tệp và trả về danh sách các từ đã tách.  
- Tệp input1.inp chứa thông tin đăng ký suất ăn của người lao động.  
- Tệp input2.inpchứa thông tin các suất ăn đã chuẩn bị.  
2. Khởi tạo hàng đợi và ngăn xếp:  
- people\_queue là một hàng đợi (deque) chứa danh sách người.  
- food\_stack là một ngăn xếp chứa các suất ăn (được đảo ngược để mô phỏng ngăn xếp).  
3. Phục vụ suất ăn:  
**-** serve\_food(people\_queue, food\_stack): Hàm này thực hiện quy trình phục vụ suất ăn.  
- Sử dụng biến no\_change\_count để theo dõi số lần liên tiếp mà không có ai nhận được suất ăn.  
- Mỗi vòng lặp, lấy một người và một suất ăn ra.  
- Nếu suất ăn đúng với đăng ký của người đó, tăng served\_count và reset no\_change\_count.  
- Nếu không, người đó quay lại cuối hàng đợi và suất ăn quay lại ngăn xếp, tăngno\_change\_count.  
- Khi no\_change\_count bằng độ dài hàng đợi, nghĩa là không ai có thể nhận được suất ăn của mình nữa.