アルゴリズムとデータ構造　問題〇のプログラム実装の考え方

学籍番号　22059　氏名　来間　空　提出日　２０２５年２３日

(1) 入力例

このプログラムでは，標準入力から一連の整数値を受け取り，それを操作命令として処理するインタフェースを備えている．入力の最初には，操作を指定するための数値（操作コード）を 1 つ入力し，必要に応じて続けて別の値（整数データ）を入力する場合もある．操作コードは 1 から 11 までの整数であり，それぞれの数値が特定のキュー操作に対応している．各操作において要求される入力内容と意味は次の通りである．

操作コードが「1」の場合は Enque 操作に対応しており，続けて整数値を 1 つ入力することで，その値をキューの末尾に追加することができる．

操作コードが「2」の場合は Deque 操作であり，この操作に続く追加の入力は必要なく，キューの先頭にある値を取り出して削除する処理が実行される．

操作コードが「3」の場合は Peek 操作となり，キューの先頭の値を削除せずに取得するために使用される．この操作も追加の入力は不要である．

操作コードが「4」の場合は，Print 操作が実行され，キューに格納されているすべての要素が先頭から末尾の順に表示される．この操作も入力値は不要である．

操作コードが「5」の場合は Clear 操作であり，キュー内に存在する全データを削除し，空の状態に初期化する．この操作においても，追加の入力は必要とされない．

操作コードが「6」の場合は，Capacity 操作が実行され，キューの最大容量が出力される．この操作は定数として設定された値（この実装では 10000）を確認するために用いられる．

操作コードが「7」の場合は Size 操作に対応し，現在キューに格納されているデータの個数を取得して出力する．この操作も他の入力を必要としない．

操作コードが「8」の場合は IsEmpty 操作となり，キューが空であるかどうかを判定し，その結果を出力するために使用される．

操作コードが「9」の場合は IsFull 操作に対応し，キューが満杯になっているかどうかを調べ，その結果を出力する．

操作コードが「10」の場合は Search 操作となり，続けて整数値を入力することで，その値

がキュー内に存在するかどうかを探索する処理が実行される．見つかった場合はその位置（先頭から数えて何番目か）を，見つからなかった場合は-1 を出力する．
最後に，操作コード「11」が入力されると，プログラムは終了処理に入り，実行を停止するようになっている．このようにして，利用者はキューの各機能を任意の順序で試すことができる．
具体的な入力例としては，以下のような操作が可能である．たとえば，次のような入力列が与えられた場合：

```
1 5
1 10
2
3
7
10 5
5
11
```

これは，5 と 10 を順にキューに追加し，先頭の要素を 1 つ取り出し，次に先頭要素を確認し，キューのサイズを取得し，値「5」を探索し，その後キューをクリアしてプログラムを終了する流れを示している．

(2) 出力例
このプログラムでは，キューに対する操作結果を標準出力にわかりやすく表示するよう設計されている．操作ごとに決められた出力形式に従ってメッセージが表示され，ユーザーは入力に対する応答を画面上で確認することができる．出力は操作コードに基づいて自動的に行われ，以下のような形式で表現される．
Enque 操作（操作コード「1」）においては，追加された値を反映したメッセージが出力される．具体的には「op1: Enque_値」という形式で表示される．たとえば値「5」を追加した場合は「op1: Enque_5」となる．
Deque 操作（操作コード「2」）では，取り出された先頭の値が表示される．表示形式は「op2: Deque_値」であり，たとえば取り出された値が「5」であれば「op2: Deque_5」と出力される．
Peek 操作（操作コード「3」）では，現在の先頭にある値を削除せずに確認することができる．この場合も「op3: Peek_値」の形式で表示される．
Print 操作（操作コード「4」）では，キュー内の全要素が 1 行ずつ順番に出力された後，「op4: Print executed」というメッセージが表示されることで，出力完了が示される．
Clear 操作（操作コード「5」）では，キューの内容がすべて削除された後，サイズが 0 となったことが確認され，「op5: Clear_0」というメッセージが表示される．

アルゴリズムとデータ構造　問題〇のプログラム実装の考え方

学籍番号　22059　氏名　来間　空　提出日　２０２５年２３日

Capacity 操作（操作コード「6」）は，キューの最大容量を表示する機能であり，「op6: Capacity_10000」のように表示される（この実装では容量が 10000 に固定されている）．
Size 操作（操作コード「7」）は，現在格納されているデータ数を出力する機能であり，「op7: Size_値」という形式で表示される．
IsEmpty 操作（操作コード「8」）では，キューが空であれば「1」，そうでなければ「0」が「op8: IsEmpty_値」の形式で出力される．
IsFull 操作（操作コード「9」）でも同様に，満杯状態であれば「1」，それ以外なら「0」が「op9: IsFull_値」として出力される．
Search 操作（操作コード「10」）では，指定された値がキュー内に見つかった場合，その位置（1 始まり）が「op10: Search_位置」の形式で出力され，見つからなかった場合には「op10: Search_-1」が表示される．
操作コード「11」によりプログラムが終了されると，「op11: Exit executed」という明確な終了メッセージが出力されることで，ユーザーに処理の完了が伝えられる．
たとえば以下のような入力例：

```
1 10
1 20
2
3
7
10 20
5
11
```

に対しては，次のような出力が想定される：

```
op1: Enque_10
op1: Enque_20
op2: Deque_10
op3: Peek_20
op7: Size_1
op10: Search_1
op5: Clear_0
op11: Exit executed
```

これにより，プログラムの各操作が適切に実行され，対応する出力が標準出力に表示されていることが確認できる．

アルゴリズムとデータ構造　問題〇のプログラム実装の考え方

学籍番号　22059　氏名　来間　空　提出日　２０２５年２３日

(3) 入力に対する出力結果の妥当性の説明

このキューに関するプログラムは，各操作に対して適切な動作を行っており，出力内容も明確に設計されているため，機能面において基本的な要件を満たしていると評価できる．各操作は，必要なチェック（空かどうか，満杯かどうか）を行いながら処理されており，例えば Deque や Peek において，キューが空である場合は処理を行わずエラーを返す仕組みが含まれている．

操作ごとに出力を観察すると，値の追加（Enque）や取り出し（Deque），表示（Print），検索（Search）といった主要な機能が正しく実装されていることが確認できる．また，出力においても，各操作がどのような結果を返したのかが一目でわかるよう，ラベル付きで明示されており，可読性が高い．

ただし，次のような点について改善の余地がある．

第一に，Enque や Deque など一部の操作において，処理失敗時の明確な出力がないため，例えば満杯や空の状態で操作を行った場合に，ユーザーがその理由を理解しづらいという欠点がある．これを改善するためには，エラー時の処理に具体的な出力文を追加することが望ましい．

第二に，Print 操作の出力では，各要素を改行して表示しているものの，前後に区切りやラベルがないため，複数回の Print 実行時にどこからどこまでが出力なのかが曖昧になることがある．たとえば「--- Queue Contents ---」などの見出しを出力することで，視認性を向上させることができる．

第三に，main 関数内で使用されている op 変数が初期化されていないため，初回の while (op != 11) の条件判定時に未定義の状態となるリスクがある．このような初期化漏れは意図しない動作の原因となるため，明示的に op = 0 としてからループに入ることが推奨される．

以上を総合して考えると，本プログラムは基本的な操作と出力について要件を満たしており，キューに関する理解と操作の学習に十分適した構成となっている．ただし，ユーザビリティと堅牢性の向上を図るためには，エラー処理の明示化や，出力の視認性向上，初期化の適正化といった細かな改善が有効である．