**Homework week 3**

Basic data structures

\*Các số trên cùng 1 dòng cách ít nhất 1 dấu cách (dùng cout để đọc vào)

**Bài 1**: Cho một dãy số A có *n* chữ số nguyên. Đếm các cặp (i, j) sao cho A[i]=A[j].

Input - Dữ liệu vào:

* Dòng đầu tiên là số:n (n <= 100000)
* Dòng tiếp theo là n số A[0], A[1], .., A[n-1]

Output – Kết quả: In ra số lượng các cặp (i, j)

Ví dụ

|  |  |
| --- | --- |
| Keyboard | Screen |
| 6  5 2 4 2 2 5 | 4 |

**Bài 2**: Làm quen với linked lists với 2 thao tác:

* insert (p, x): Chèn vào vị trí p số x, vị trí đầu tiên là 0, các phần tử từ p trở đi lùi lại 1 vị trí
* delete (p): Xóa một phần tử tại vị trí p

Bắt đầu từ dãy rỗng, đọc vào 1 list các lệnh insert và delete. In ra kết quả dãy cuối cùng

Input – Dữ liệu vào:

* Dòng đầu tiên là số n, số lượng các phép toán (số dòng)
* n dòng tiếp theo là 1 trong 2 format
  + insert p x
  + delete p

với p là vị trí x là number

Output – Kết quả: In ra danh sách liên kết

**Example**

|  |  |
| --- | --- |
| Keyboard | Screen |
| 5  insert 0 1  insert 1 3  insert 1 4  insert 2 2  delete 1 | 1 2 3 |

**Bài 3**: Cho một dãy số, yêu cầu đọc vào 1 danh sách liên kết kép. Tiến hành xóa dãy số theo quy luật sau:

while (true)

- Tìm 3 số liên tiếp đầu tiên có tổng chia hết cho 2 (i-1, i, i + 1). Xóa đi phần tử i.

- nếu không thì thoát khỏi vòng lặp

In ra kết quả dãy số cuối cùng

Input:

* n
* n số của dãy số

Output: In ra kết quả cuối cùng của thuật toán

**Example**

|  |  |
| --- | --- |
| Keyboard | Screen |
| 6  1 5 6 1 2 2 | 1 2 2 |

**Bài 4**: Làm quen với queue và phép toán enqueue/dequeue.

* enqueue (x): thêm 1 phần tử vào queue
* dequeue (): lấy 1 phần tử ra khỏi queue

Bắt đầu từ queue ko chứa phần tử nào, đọc vào các phép toán trên từ bàn phím và in ra kết quả cuối cùng.

Input:

* n số lượng cac phép toán
* n dòng tiếp theo là 1 trong 2 phép toán
  + enqueue x
  + dequeue

Output: In ra kết quả của queue trong màn hình

**Example**

|  |  |
| --- | --- |
| Keyboard | Screen |
| 7  enqueue 1  enqueue 2  enqueue 3  enqueue 4  dequeue  dequeue  enqueue 5 | 3 4 5 |

**Bài 5**: Làm quen với 2 phép toán push/pop của stack

* push (x): thêm 1 phần tử vào stack
* pop (): lấy 1 phần tử ra khỏi stack

Bắt đầu từ stack ko chứa phần tử nào, đọc vào các phép toán trên từ bàn phím và in ra kết quả cuối cùng.

Input:

* n số lượng cac phép toán
* n dòng tiếp theo là 1 trong 2 phép toán
  + push x
  + pop

Output: In ra kết quả của queue trong màn hình

**Example**

|  |  |
| --- | --- |
| Keyboard | Screen |
| 7  push 1  push 2  push 3  push 4  pop  pop  push 5 | 1 2 5 |

**Bài tập nâng cao**

**Bài 6**:

Cho 1 mảng 2 chiều kích thước gồm 0 và 1. In ra diện tích của hình chữ nhật lớn nhất gồm toàn 0

Input:

- Dòng đầu tiên là 1 số n

- n dòng tiếp theo mỗi dòng n số là phần tử của ma trận

Output:

Diện tích hình chữ nhật lớn nhất

|  |  |
| --- | --- |
| Keyboard | Screen |
| 5  0 0 0 0 0  0 0 1 1 1  1 0 0 0 0  0 0 0 0 0  1 0 0 1 1 | 8 |

**Bài 7**: (ứng dụng stack)

Cho một biểu thức gồm các phép tính đơn giản + - \* và dấu ( ).

Tính giá trị của biểu thức đó:

Input:

- 1 Dòng chứa biểu thức, các phần tử liên tiếp cách nhau bởi dấu cách

Output

- Kết quả của phép toán

|  |  |
| --- | --- |
| Keyboard | Screen |
| ( 1 + 1 + 2 ) \* 2 + 2 \* 10 | 28 |
| 100 - ( 2 + 2 \* ( 4 - 3 ) ) | 96 |

**Bài 8**: (ứng dụng queue) Tìm đường đi ngắn nhất trên ma trận 2 chiều

Cho 1 mảng 2 chiều kích thước gồm 0 và 1. In ra độ dài đường đi ngắn nhất gồm toàn các số 0 từ ô trái trên đến ô phải dưới. Ở mỗi bước di chuyển chỉ được đi đến các ô cùng cạnh như hình dưới. Giả sử đang đứng tại hàng i, cột j

|  |  |  |
| --- | --- | --- |
|  | **i-1, j** |  |
| **i, j-1** | **i, j** | **i, j+1** |
|  | **i+1, j** |  |

Input:

- Dòng đầu tiên là 1 số n

- n dòng tiếp theo mỗi dòng n số là phần tử của ma trận. Điều kiện là lúc nào cũng tồn tại đường đi gồm toàn 0 nối từ ô trái trên đên ô phải dưới.

Output:

Độ dài đường đi ngắn nhất

|  |  |
| --- | --- |
| Keyboard | Screen |
| 5  0 0 0 0 0  0 0 1 1 0  1 1 0 0 0  0 0 0 1 1  1 0 0 0 0 | 12 |