Kierownictwo poprosiło Cię o zaimplementowanie uniwersalnego algorytmu kolejkującego osoby ubiegające się o posadę w firmie. Jako iż kierownik firmy jest wielkim fanem natury, implementacja kolejki musi wykorzystywać drzewo BST. Program powinien składać się z trzech modułów: kolejkowania, edycji oraz raportowania, a komunikacja odbywać się będzie poprzez standardowe wejście.

Moduły przedstawiają się następująco:

- **Moduł kolejkowania**: powinien implementować podstawowe operacje na kolejce:
  - ENQUE x – dodaje osobę o priorytecie x do kolejki.
  - DEQUEMAX – zwraca oraz usuwa osobę o najwyższym priorytecie.
  - DEQUEMIN – zwraca oraz usuwa osobę o najniższym priorytecie.
  - NEXT x – zwraca najbliższą osobę o priorytecie większym od x.
  - PREV x – zwraca najbliższą osobą o priorytecie mniejszym od x.

- **Moduł edycji**: mimo, iż typowa kolejka nie implementuje takich operacji, czasami wymagane jest utworzenie jej z istniejącej bazy osób. Dodatkowo, zdarza się, że osoby zapisane na rozmowę kwalifikacyjną rezygnują. Umożliwiają to operacje:
  - CREATE order n x_1 … x_n – rekurencyjnie tworzy drzewo kolejki na podstawie listy **n** kluczy x_1 do x_n podanych w porządku wyznaczonym przez drugi argument (PREORDER albo POSTORDER). Jeśli kolejka istnieje, powinna zostać zastąpiona nową.
  - DELETE x – usuwa osobę o priorytecie x. W przypadku, gdy w drzewie usuwany węzeł ma dwóch potomków, zamienia go z jego następnikiem.

- **Moduł raportowania**: kierownik firmy zażądał stałego dostępu do danych w drzewie, a jako fan natury umieścił dodatkowe wymagania w postaci następujących komend:
  - PREORDER – wypisuje listę osób w porządku preorder.
  - INORDER – wypisuje listę osób w porządku inorder.
  - POSTORDER – wypisuje listę osób w porządku postorder.
  - HEIGHT – rekurencyjnie zwraca wysokość drzewa.

Wejście:

Dane do programu wczytywane są ze standardowego wejścia (klawiatury), zgodnie z poniższą specyfikacją:

- Pierwsza linia zawiera liczbę całkowitą n $(1 \leq n \leq 100)$ oznaczającą ilość testów.
- Pierwsza linia testu zawiera liczbę całkowitą m $(1 \leq m \leq 100)$ oznaczającą ilość komend do wykonania na kolejce.
- W każdej następnej linii znajduje się jedna z wymienionych wyżej operacji i ewentualnie jej argument(y).
- W operacjach CREATE oraz ENQUE jako x przyjmuje się trójkę `<priorytet, imię, nazwisko>`. W operacjach DELETE, PREV oraz NEXT jako x podawany jest tylko priorytet.

Wyjście:

Każdy test powinien zaczynać się od wypisania ciągu „ZESTAW n” (od 1). Dla każdej operacji w zestawie wypisz w jednej linii jej wynik zgodnie z podanymi przykładami.

Wymagania implementacyjne:

1. Jedynym możliwym importem jest skaner wczytywania z klawiatury.
2. Wszystkie komendy oprócz CREATE oraz HEIGHT muszą być zaimplementowane w wersji iteracyjnej.
3. Można założyć, że żadna operacja nie zostanie wywołana na kolejce przed wykonaniem operacji CREATE.
4. Złożoność wszystkich komend powinna być optymalna. W szczególności, żadna komenda nie powinna mieć złożoności większej niż liniowa.
5. Operacje PREV x oraz NEXT x nie mogą korzystać z listy INORDER.
6. Na końcu kodu przesyłanego submitu proszę dopisać w formie komentarza własne dane wejściowe.
7. Przypominam o komentowaniu aplikacji w formie opisanej w punkcie 3 Regulaminu zaliczania programów na BaCy z roku 2020/2021.
8. Klasa Node oraz Person powinna mieć postać:

```
class Node {
   public Person info; // element danych (klucz)
   public Node left; // lewy potomek węzła
   public Node right; // prawy lewy potomek węzła
   // konstruktor…
}
// koniec klasy Node
class Person {
  public int priority;
  public String name;
  public String surname;
  // konstruktor…
}
// koniec klasy Person
```

Test jawny:

test.in:	***test.out:***
1 16 CREATE POSTORDER 5 12 Adam Nowak 33 Anna Kowalska 30 Marek Mickiewicz 43 Zofia Krzak 37 Ola Nowicka POSTORDER PREORDER INORDER DEQUEMAX DEQUEMIN INORDER NEXT 33 NEXT 93 PREV 50 DELETE 50 DELETE 33 POSTORDER ENQUE 35 Andrzej Wolny INORDER HEIGHT	ZESTAW 1 POSTORDER: 12 - Adam Nowak, 33 - Anna Kowalska, 30 - Marek Mickiewicz, 43 - Zofia Krzak, 37 - Ola Nowicka PREORDER: 37 - Ola Nowicka, 30 - Marek Mickiewicz, 12 - Adam Nowak, 33 - Anna Kowalska, 43 - Zofia Krzak INORDER: 12 - Adam Nowak, 30 - Marek Mickiewicz, 33 - Anna Kowalska, 37 - Ola Nowicka, 43 - Zofia Krzak DEQUEMAX: 43 - Zofia Krzak DEQUEMIN: 12 - Adam Nowak INORDER: 30 - Marek Mickiewicz, 33 - Anna Kowalska, 37 - Ola Nowicka NEXT 33: 37 - Ola Nowicka NEXT 93: BRAK PREV 50: BRAK DELETE 50: BRAK POSTORDER: 30 - Marek Mickiewicz, 37 - Ola Nowicka INORDER: 30 - Marek Mickiewicz, 35 - Andrzej Wolny, 37 - Ola Nowicka HEIGHT: 2