Wskaźniki

Przeanalizuj dokładnie problem opisany w każdym zadaniu. Następnie skonstruuj rozwiązanie opisanego problemu (algorytm) i zaimplementuj je w języku C++.

Zadanie 1: dynamiczna tablica

Celem tego zadania jest implementacja dynamicznej tablicy zdolnej do przetwarzania różnych operacji wejściowych. Tablica ta powinna być ręcznie inicjalizowana za pomocą operacji `new` i jej zawartość powinna być wypisywana po każdej operacji. Każdą z poniższych operacji zrealizuj za pomocą osobnej funkcji. Zaprojektuj menu programu pozwalające na wybranie operacji do wykonania, lub zakończenie działania programu.

1. **Inicjalizacja tablicy:**
   Rozpocznij od stworzenia pustej tablicy o początkowej pojemności, np. 10 elementów. Pamiętaj, że pojemność ta będzie musiała być zwiększana w miarę potrzeb.
2. **Dodawanie elementu na koniec tablicy:**
   - Użytkownik powinien mieć możliwość wprowadzenia liczby, która zostanie dodana na końcu tablicy.
   - Jeżeli tablica jest pełna, powiększ jej pojemność dwukrotnie.
   - Wypisz aktualną zawartość tablicy.
3. **Wstawianie elementu na wskazane miejsce w tablicy:**
   - Użytkownik powinien podać indeks oraz wartość, która ma być wstawiona na wskazane miejsce w tablicy.
   - Jeżeli tablica jest pełna, powiększ jej pojemność dwukrotnie.
   - Pamiętaj o przesunięciu pozostałych elementów, aby zrobić miejsce dla nowego elementu.
   - Wypisz aktualną zawartość tablicy.
4. **Usunięcie elementu na wskazanej pozycji:**
   - Użytkownik podaje indeks elementu, który ma zostać usunięty.
   - Przesuń pozostałe elementy tak, aby nie było pustego miejsca po usuniętym elemencie.
   - Wypisz aktualną zawartość tablicy.
5. **Usunięcie wszystkich elementów o podanej wartości:**
   - Użytkownik podaje wartość.
   - Usuń wszystkie elementy o tej wartości z tablicy, przesuwając pozostałe elementy w odpowiedni sposób.
   - Wypisz aktualną zawartość tablicy.

Wskazówki:

- Do alokacji pamięci dla tablicy wykorzystaj funkcję `new` i pamiętaj o zwolnieniu pamięci za pomocą `delete[]`, gdy tablica nie jest już potrzebna.
- Pamiętaj, aby kontrolować pojemność tablicy. Jeśli zabraknie miejsca, tablica powinna być zwiększana **dwukrotnie**.

- Dla uproszczenia zakładamy, że użytkownik poda poprawne dane wejściowe. Jednak dla bardziej zaawansowanego podejścia można dodać odpowiednie sprawdzenie błędów.

Zadanie 2: tablica wskaźników i sortowanie

Celem zadania jest manipulacja wskaźnikami oraz sortowanie wartości nie zmieniając oryginalnej kolejności elementów tablicy.

1. **Utworzenie tablicy n losowych wartości:**
   Inicjalizuj tablicę o rozmiarze n losowymi wartościami (np. z zakresu od 1 do 1000).
2. **Utworzenie tablicy wskaźników:**
   Stwórz drugą tablicę, która będzie przechowywała wskaźniki na elementy oryginalnej tablicy.
3. **Sortowanie tablicy wskaźników:**
   Posortuj tablicę wskaźników tak, by wskaźniki były w kolejności rosnącej wartości wskazywanych przez nie elementów oryginalnej tablicy.
4. **Wypisanie pozycji w posortowanym ciągu:**
   Dla każdego elementu oryginalnej tablicy wypisz, którą pozycję zajmuje on w posortowanym ciągu. Przy tym kroku użyj posortowanej tablicy wskaźników do określenia pozycji elementu. Wypisz wynik w formie: `wartość -> pozycja w posortowanym ciągu`.

Wskazówki:

- Wykorzystaj wskaźniki do odnoszenia się do wartości w oryginalnej tablicy, zamiast przesuwać same wartości.
- Pamiętaj, że sortowanie powinno dotyczyć tylko tablicy wskaźników, oryginalna tablica wartości powinna pozostać nienaruszona.
- Możesz wykorzystać dowolny algorytm sortowania, jednak pamiętaj o jego efektywności dla różnych wielkości tablic.

Przykład:

Jeżeli oryginalna tablica zawiera wartości `[5, 3, 8]`, to po sortowaniu tablicy wskaźników, pozycje tych wartości w posortowanym ciągu będą odpowiednio: `2, 1, 3`. Wynik, jaki powinien zostać wypisany to:

```
5 -> 2
3 -> 1
8 -> 3
```