# Podstawy Programowania Komputerów

Lista jednokierunkowa

21 listopada 2018

W programie zadeklarowano następujące typy:

```
typedef int T;

/* element listy jednokierunkowej */
struct element
{
   /* wartość przechowywana w elemencie listy */
   T wartosc;
   /* wskaźnik na następny element w liście */
   element * pNext;
};
```

Proszę zaimplementować następujące funkcje:

1.
```
/** Funkcja dodaje liczbę do listy posortowanej niemalejąco ze względu
    na przechowywane liczby. Po dodaniu lista jest nadal posortowana
    niemalejąco. Wartość dodawana jest w sposób iteracyjny.
@param pHead wskaźnik na pierwszy element listy jednokierunkowej
@param liczba wartość do dodania do listy
 */
void dodajIteracyjnieDoListyPosortowanej (element * & pHead, T liczba);
```

2.
```
/** Funkcja dodaje liczbę do listy posortowanej niemalejąco ze względu
    na przechowywane liczby. Po dodaniu lista jest nadal posortowana
    niemalejąco. Wartość dodawana jest w sposób rekurencyjny.
@param pHead wskaźnik na pierwszy element listy jednokierunkowej
@param liczba wartość do dodania do listy
 */
void dodajRekurencyjnieDoListyPosortowanej (element * & pHead, T liczba)
    ;
```

3.
```
/** Funkcja wyszukuje w liście o głowie pHead element przechowujący
    podaną liczbę. Element jest wyszukiwany iteracyjnie.
@param pHead wskaźnik na pierwszy element listy jednokierunkowej
@param liczba wartość do wyszukania w liście
@return Funkcja zwraca adres elementu przechowującego zadaną liczbę. Jeż
    eli w liście jest więcej elementów o zadanej wartości, funkcja
    zwraca adres dowolnego elementu przechowującego zadaną wartość. Jeż
    eli w liście nie występuje zadana wartość, funkcja zwraca nullptr.
 */
element * znajdzElementIteracyjnie (element * pHead, T liczba);
```

4.
```
/** Funkcja wyszukuje w liście o głowie pHead element przechowujący
    podaną liczbę. Element jest wyszukiwany rekurencyjnie.
@param pHead wskaźnik na pierwszy element listy jednokierunkowej
@param liczba wartość do wyszukania w liście
@return Funkcja zwraca adres elementu przechowującego zadaną liczbę. Jeż
    eli w liście jest więcej elementów o zadanej wartości, funkcja
    zwraca adres dowolnego elementu przechowującego zadaną wartość. Jeż
    eli w liście nie występuje zadana wartość, funkcja zwraca nullptr.
 */
element * znajdzElementRekurencyjnie (element * pHead, T liczba);
```

5.
```
/** Funkcja usuwa element o podanym adresie.
@param pHead wskaźnik na pierwszy element listy jednokierunkowej
@param pDoUsuniecia adres elementu do usunięcia
 */
void usunElement (element * & pHead, element * pDoUsuniecia);
```

6.
```
/** Funkcja usuwa element o podanej wartości. Jeżeli w liście występuje
    więcej elementów o podanej wartości, funkcja usuwa dowolny z nich.
    Jeżeli w liście nie ma elementu do usunięcia, nie robi nic. Funkcja
    wykorzystuje funkcje zdefiniowane powyżej.
@param pHead wskaźnik na pierwszy element listy jednokierunkowej
@param wartosc wartość elementu do usunięcia
 */
void usunElement (element * & pHead, T wartosc);
```

7.
```
/** Funkcja usuwa wszystkie elementy o podanej wartości. Jeżeli w liś
    cie nie ma elementu o wartości do usunięcia, nie robi nic.
@param pHead wskaźnik na pierwszy element listy jednokierunkowej
@param wartosc wartość elementu do usunięcia
 */
void usunElementy (element * & pHead, T wartosc);
```

8.
```
/** Funkcja odwraca kolejność elementów w liście.
@param[in, out] pHead wskaźnik na pierwszy element listy
    jednokierunkowej, po wykonaniu funkcji wskazuje na nowy pierwszy
    element listy
 */
void odwrocListe (element * & pHead);
```

9.
```
/** Funkcja usuwa wszystkie elementy powtarzające się w liście
    jednokierunkowej. Po wykonaniu funkcji każdy element występuje w liś
    cie co najwyżej raz.
@param[in, out] pHead wskaźnik na pierwszy element listy
    jednokierunkowej
 */
void usunPowtorzenia (element * & pHead);
```

10.
```
/** Funkcja realizuje sumę mnogościową zbiorów reprezentowanych przez
    listy. Po wykonaniu funkcji zbiory wejściowe nie są zmodyfikowane.
```

```
@param pH1 wskaźnik na pierwszy element pierwszej listy jednokierunkowej
     (pierwszy zbiór), po wykonaniu funkcji lista nie jest zmodyfikowana
@param pH2 wskaźnik na pierwszy element drugiej listy jednokierunkowej (
    drugi zbiór), po wykonaniu funkcji lista nie jest zmodyfikowana
@return wskaźnik na listę będącą sumą mnogościową zbiorów
 */
element * sumaMnogosciowa (element * pH1, element * pH2);
```

11.

```
/** Funkcja realizuje iloczyn mnogościowy zbiorów reprezentowanych przez
     listy. Po wykonaniu funkcji zbiory wejściowe nie są zmodyfikowane.
@param pH1 wskaźnik na pierwszy element pierwszej listy jednokierunkowej
     (pierwszy zbiór), po wykonaniu funkcji lista nie jest zmodyfikowana
@param pH2 wskaźnik na pierwszy element drugiej listy jednokierunkowej (
    drugi zbiór), po wykonaniu funkcji lista nie jest zmodyfikowana
@return wskaźnik na listę będącą iloczynem mnogościowym zbiorów
 */
element * iloczynMnogosciowy (element * pH1, element * pH2);
```

12.

```
/** Funkcja realizuje różnicę zbiorów reprezentowanych przez listy. Po
    wykonaniu funkcji zbiory wejściowe nie są zmodyfikowane.
@param pH1 wskaźnik na pierwszy element pierwszej listy jednokierunkowej
     (pierwszy zbiór), po wykonaniu funkcji lista nie jest zmodyfikowana
@param pH2 wskaźnik na pierwszy element drugiej listy jednokierunkowej (
    drugi zbiór), po wykonaniu funkcji lista nie jest zmodyfikowana
@return wskaźnik na listę będącą różnicą zbiorów
 */
element * roznicaMnogosciowa (element * pH1, element * pH2);
```

13.

```
/** Funkcja usuwa wszystkie elementy przechowujące wartości parzyste.
@param pHead wskaźnik na listę jednokierunkową
 */
void usunParzyste (element * & pHead);
```

14.

```
/** Funkcja usuwa ostatni element z listy. Jeżeli lista ma tylko jeden
    element, jest on usuwany. Jeżeli funkcja nie ma żadnych elementów,
    funkcja nie robi nic.
@param pHead wskaźnik na listę jednokierunkową
 */
void usunOstatni (element * & pHead);
```

15.

```
/** Funkcja scala dwie posortowane listy. W wyniku powstaje jedna lista
    posortowana. Funkcja nie alokuje ani nie zwalnia pamięci na stercie.
@param pH1 wskaźnik na pierwszy element pierwszej posortowanej listy
    jednokierunkowej, po wykonaniu funkcji -- wskaźnik zerowy
@param pH2 wskaźnik na pierwszy element drugiej posortowanej listy
    jednokierunkowej, po wykonaniu funkcji -- wskaźnik zerowy
@return wskaźnik na posortowaną listę będącą scaleniem zbiorów
 */
element * scalListyPosortowane (element * & pH1, element * & pH2);
```