

Ćwiczenia 10

Wskaźniki - cd

1. Napisać bibliotekę udostępniającą:

- typ *Sznur* będący dynamiczną listą łączoną przechowującą liczby całkowite
- funkcję wypisującą zawartość sznura
- funkcję wstawiającą na koniec sznura element zawierający liczbę podaną jako parametr
- funkcję *usunFragmenty* usuwającą ze sznura wszystkie takie ciągi sąsiednich elementów przechowujących jednakowe wartości które mają długość nie mniejszą niż *N* (gdzie *N* jest dodatnią liczbą całkowitą będącą parametrem funkcji). Pamięć zajmowana przez usuwane elementy ma zostać zwolniona; funkcja nie może używać tablic ani pomocniczych sznurów;
- funkcję *usunZPar* modyfikującą sznur w następujący sposób: rozpatrujemy rozłączne pary elementów (początkowego) sznura; jeśli suma pary jest mniejsza niż *G* (gdzie *G* jest liczbą całkowitą będącą parametrem funkcji), to usuwamy pierwszy element pary, w przeciwnym razie usuwamy jej drugi element. Pamięć zajmowana przez usuwane elementy ma zostać zwolniona, funkcja nie może użyć tablic ani pomocniczych sznurów;
- funkcję *PrzestawParzyste* modyfikującą sznur poprzez przepięcie na jego koniec wszystkich elementów zawierających liczby parzyste, odwracając przy tym ich kolejność. Funkcja nie może używać tablic i pomocniczych sznurów;
- funkcję *Odwroc* modyfikującą sznur poprzez odwrócenie kolejności elementów w *N*-elementowym początkowym fragmencie sznura. Jeśli *N* jest większe lub równe długości sznura, odwracany jest cały sznur. Odwracanie należy wykonać poprzez przepinanie elementów, niedozwolone jest używanie tablic i pomocniczych sznurów;
- funkcję *Przegrupuj* modyfikującą dany sznur oraz drugi sznur *S2* podany jako parametr. Funkcja przegrupowuje elementy sznurów tak, aby "ten" sznur (oznaczony w dalszym opisie przez *S1*) zawierał wszystkie te elementy z obu sznurów które przechowują wartości parzyste, a *S2* – wszystkie elementy obu sznurów przechowujące wartości nieparzyste. Wartości w obu sznurach mają być uporządkowane następująco: wartość z *S1*, wartość z *S2*, wartość z *S1*, wartość z *S2* itd., na końcu ewentualne pozostałe elementy danego rodzaju z tego sznura który miał ich więcej. Sznury wynikowe mają zachowywać kolejność elementów ze sznurów wejściowych (przykład: dla *S1*=2,4,6,8,9,11,13 i *S2*=10,17,18,19 wynikiem ma być *S1*=2,10,4,18,6,8 i *S2*=9,17,11,19,13). Funkcja nie może wykorzystywać tablic ani sznurów pomocniczych.

oraz program testujący działanie tej biblioteki.

2. Napisać bibliotekę udostępniającą:

- typ strukturalny *Student* o polach: *indeks* (liczba całkowita nieujemna) i *srednia_ocen* (liczba rzeczywista)
- typ *Lista_studentow* będący dynamiczną listą łączoną o elementach przechowujących dane studenta oraz wskaźnik na następny element listy
- funkcję wypisującą zawartość listy
- funkcję wstawiającą do listy element zawierający studenta podanego jako parametr tak aby lista była posortowana rosnąco według numerów indeksów. W przypadku gdy w liście istnieje już student o podanym indeksie funkcja powinna zgłosić wyjątek
- funkcję modyfikującą zawartość listy tak, aby na początku znajdowały się uporządkowane rosnąco według indeksów osoby o średniej ocen nie mniejszej niż liczba rzeczywista *G* podana jako parametr, a następnie uporządkowane rosnąco według indeksów osoby o średniej ocen mniejszej niż *G* (zakładamy, że lista wejściowa jest uporządkowana rosnąco według indeksów; modyfikacja listy powinna być wykonana poprzez przepinanie elementów)
- funkcję usuwającą z listy wszystkie elementy przechowujące dane studentów mających średnią ocen mniejszą niż liczba rzeczywista *G* podana jako parametr. Pamięć zajmowana przez usunięte elementy ma zostać zwolniona, funkcja nie może używać tablic ani list pomocniczych

oraz program testujący działanie biblioteki.