

## **Zestaw 1:**

**Organizacja plików:** Wszystkie pliki oddawane do sprawdzenia należy zapisać we wspólnym folderze o nazwie będącej numerem indeksu, umieszczonym na pulpicie.

**Oddajemy tylko źródła programów (pliki o rozszerzeniach .cpp i .h)!!!**

**Zad. 1:** Napisać bibliotekę udostępniającą:

- typ Sznur będący dynamiczną listą łączoną przechowującą liczby całkowite;
- funkcję wypisz wypisującą zawartość sznura;
- funkcję wstaw wstawiającą do sznura element zawierający liczbę podaną jako parametr tak, aby sznur był uporządkowany nierosnąco (zakładając, że sznur „na wejściu” jest już uporządkowany w taki sposób);
- funkcję usun rozpatrującą rozłączne trójki elementów sznura podanego jako parametr i usuwającą pierwszy element trójki jeżeli średnia trójki jest mniejsza od liczby całkowitej K będącej parametrem funkcji, lub usuwającą drugi i ostatni element w trójce w przeciwnym przypadku. W przypadku sznura o długości nie podzielnej przez 3 końcowe elementy (nie wchodzące w skład żadnej trójki – trójki wybierane są tak, że pierwszy element sznura jest początkowym elementem pierwszej trójki) nie są usuwane. Pamięć zajmowana przez usuwane elementy ma zostać zwolniona. Funkcja nie może używać tablic, pomocniczych sznurów ani struktur danych dostępnych w bibliotece standardowej (wektorów, list, kolejek, itp);

oraz program testujący działanie biblioteki poprzez pobranie sznura (funkcja wstaw; użytkownik podaje liczby, które mają być wstawione do sznura i decyduje o zakończeniu podawania), wypisanie jego zawartości (funkcja wypisz; przykład: dla zestawu liczb wejściowych 1,3,6,1,3,7,2,6,3,1,1 sznur uzyskuje postać 7,6,6,3,3,3,2,1,1,1,1), a następnie po pobraniu od użytkownika wartości K, usunięcie w sznurze wybranych elementów z rozłącznych trójek (funkcja usun) i ponowne wypisanie jego zawartości (przykład: dla K=3 sznur 7,6,6,3,3,3,2,1,1,1,1, zostanie przekształcony do postaci 7,3,1,1,1,1, a ten sam sznur dla K=7 zostanie przekształcony do postaci 6,6,3,3,1,1,1,1).

**Zad. 2:** Rozszerzyć bibliotekę z zad.1 o funkcję przenies, która modyfikuje sznur poprzez przepięcie wybranych elementów z rozłącznych par w następujący sposób: jeżeli elementy w parze mają tę samą wartość przepinane są na początek sznura z zachowaniem wzajemnej oryginalnej kolejności par, jeżeli jednak elementy mają różne wartości, to drugi z elementów pary jest przepinany na koniec sznura z odwróceniem kolejności przenoszonych elementów (pary wybierane są tak że pierwszy element sznura jest pierwszym elementem pierwszej pary, a przypadku sznura nieparzystej długości jego ostatni element nie jest elementem żadnej pary). Przykład: dla sznura przechowującego liczby 9,9,9,8,8,6,6,6,6,5,5,5,5,2,2,1,1 dostaniemy wynik 9,9,6,6,5,5,9,8,6,5,2,1,1,2,5,6,8, sznur 9,8,7,5,4,3,2 uzyska postać 9,7,4,2,3,5,8, a sznur 3,3,2,2,1,1 nie zmieni się. Funkcja nie może używać tablic, pomocniczych sznurów ani struktur danych dostępnych w bibliotece standardowej (wektorów, list, kolejek, itp). Wykorzystać funkcję w programie testującym.

**Zad. 3:** Rozszerzyć bibliotekę z zad.1 o funkcję wymienStaleCiagi, modyfikującą dany sznur oraz drugi sznur S2 podany jako parametr w następujący sposób: pierwszy nieprzedłużalny (być może jednoelementowy) ciąg stały z „tego” sznura (oznaczonego w dalszym opisie przez S1) ma zostać zamieniony z pierwszym takim ciągiem z S2, następnie pomijamy w obu sznurach nieprzedłużalny ciąg sąsiednich różnych wartości (o ile istnieje), wybieramy kolejne dwa ciągi stałe do wymiany pomiędzy sznurami, omijamy ciągi różnowartościowe, itd. Ewentualne elementy nadmiarowych ciągów w jednym ze sznurów nie są przenoszone. Przykład: dla sznurów S1=7,7,6,5,4,4,4,4,3,2,2 i S2=8,8,6,6,6,5,4,4,2,2 wynikiem będzie S1=8,8,6,5,4,6,6,3,2,4 i S2=7,7,6,4,4,4,5,4,2,2,2, dla S1=5,5 i S2=9,8,7,6,1 wynikiem będzie S1=9 oraz S2=5,5,8,7,6,1). Funkcja nie może używać tablic, pomocniczych sznurów ani struktur danych dostępnych w bibliotece standardowej (wektorów, list, kolejek, itp). Wykorzystać funkcję w programie testującym.

## **Zestaw 2:**

**Organizacja plików:** Wszystkie pliki oddawane do sprawdzenia należy zapisać we wspólnym folderze o nazwie będącej numerem indeksu, umieszczonym na pulpicie.

**Oddajemy tylko źródła programów (pliki o rozszerzeniach .cpp i .h)!!!**

**Zad. 1:** Napisać bibliotekę udostępniającą:

- typ Sznur będący dynamiczną listą łączoną przechowującą liczby całkowite;
- funkcję wypisz wypisującą zawartość sznura;
- funkcję wstaw wstawiającą do sznura element zawierający liczbę podaną jako parametr tak, aby sznur był uporządkowany nierosnąco (zakładając, że sznur „na wejściu” jest już uporządkowany w taki sposób);
- funkcję usun rozpatrującą rozłączne czwórki elementów sznura i usuwającą całą czwórkę jeśli suma wartości jej skrajnych elementów jest równa sumie wartości elementów środkowych, albo pierwszy i ostatni element czwórki w przeciwnym przypadku. W przypadku sznura o długości nie podzielnej przez 4 końcowe elementy (nie wchodzące w skład żadnej czwórki – czwórki wybierane są tak, że pierwszy element sznura jest początkowym elementem pierwszej czwórki) nie są modyfikowane. Pamięć zajmowana przez usuwane elementy ma zostać zwolniona. Funkcja nie może używać tablic, pomocniczych sznurów ani struktur danych dostępnych w bibliotece standardowej (wektorów, list, kolejek, itp);

oraz program testujący działanie biblioteki poprzez pobranie sznura (funkcja wstaw; użytkownik podaje liczby, które mają być wstawione do sznura i decyduje o zakończeniu podawania), wypisanie jego zawartości (funkcja wypisz; przykład: dla zestawu liczb wejściowych 1,3,6,1,3,9,3,6,3,1,1 sznur uzyskuje postać 9,6,6,3,3,3,1,1,1,1), a następnie usunięcie w sznurze rozłącznych czwórek elementów lub skrajnych elementów tych czwórek (funkcja usun) i ponowne wypisanie jego zawartości (przykład: sznur 9,6,6,3,3,3,1,1,1,1, zostanie przekształcony do postaci 3,3,1,1,1, sznur 8,7,6,5,3,2,2,1 stanie się pusty, sznur 7,6,5,5,4,4,3,2,1 zostanie przekształcony do postaci 6,5,4,3,1).

**Zad. 2:** Rozszerzyć bibliotekę z zad.1 o funkcję przenies, która modyfikuje sznur poprzez przepięcie wybranych elementów z rozłącznych par w następujący sposób: jeżeli elementy w parze mają tę samą wartość przepinane są na koniec sznura z odwróceniem wzajemnej oryginalnej kolejności par, jeżeli jednak elementy mają różne wartości, to pierwszy z elementów pary jest przepinany na początek sznura z zachowaniem kolejności przenoszonych elementów (pary wybierane są tak że pierwszy element sznura jest pierwszym elementem pierwszej pary, a przypadku sznura nieparzystej długości jego ostatni element nie jest elementem żadnej pary). Przykład: dla sznura przechowującego liczby 9,9,9,8,8,6,6,6,6,5,5,5,5,2,2,1,1 dostaniemy wynik 9,8,6,5,2,8,6,5,2,1,1,5,5,6,6,9,9, sznur 9,8,7,5,4,3,2 uzyska postać 9,7,4,8,5,3,2, a sznur 3,3,2,2,1,1 postać 1,1,2,2,3,3. Funkcja nie może używać tablic, pomocniczych sznurów ani struktur danych dostępnych w bibliotece standardowej (wektorów, list, kolejek, itp). Wykorzystać funkcję w programie testującym.

**Zad. 3:** Rozszerzyć bibliotekę z zad.1 o funkcję wymienCiagi, modyfikującą dany sznur oraz drugi sznur S2 podany jako parametr w następujący sposób: pierwszy nieprzedłużalny (być może jednoelementowy) ciąg sąsiednich elementów z „tego” sznura (oznaczonego w dalszym opisie przez S1) przechowujących różne wartości ma zostać zamieniony z pierwszym takim ciągiem z S2, następnie pomijamy w obu sznurach nieprzedłużalny ciąg stały (być może jednoelementowy), wybieramy kolejne dwa ciągi różnowartościowe do wymiany pomiędzy sznurami, omijamy ciągi stałe, itd. Ewentualne elementy nadmiarowych ciągów w jednym ze sznurów nie są przenoszone. Przykład: dla sznurów S1=7,6,6,5,4,4,4,4,3,2,1 i S2=8,8,6,6,6,5,4,4,2,2 wynikiem będzie S1=8,6,6,4,4,4,5,4 i S2=7,6,8,5,4,6,6,3,2,1,4,2,2, dla S1=5,5 i S2=9,8,7,6,1 wynikiem będzie S1=9,8,7,6,1,5 oraz S2=5). Funkcja nie może używać tablic, pomocniczych sznurów ani struktur danych dostępnych w bibliotece standardowej (wektorów, list, kolejek, itp). Wykorzystać funkcję w programie testującym.

### **Zestaw 3:**

**Organizacja plików:** Wszystkie pliki oddawane do sprawdzenia należy zapisać we wspólnym folderze o nazwie będącej numerem indeksu, umieszczonym na pulpicie.

**Oddajemy tylko źródła programów (pliki o rozszerzeniach .cpp i .h)!!!**

**Zad. 1:** Napisać bibliotekę udostępniającą:

- typ Sznur będący dynamiczną listą łączoną przechowującą liczby całkowite;
- funkcję wypisz wypisującą zawartość sznura;
- funkcję wstaw wstawiającą do sznura element zawierający liczbę podaną jako parametr tak, aby sznur był uporządkowany nierosnąco (zakładając, że sznur „na wejściu” jest już uporządkowany w taki sposób);
- funkcję usunPlus rozpatrującą rozłączne pary elementów sznura i usuwającą całą parę jeśli zawiera równe wartości. Dodatkowo, jeżeli trzy elementy następujące bezpośrednio po usuniętej parze zawierają wartości mniejsze od usuwanych o co najwyżej K, to też są usuwane (usuwamy dokładnie trzy dodatkowe elementy; w tym przypadku nie są one uwzględniane w kolejnych parach), K jest liczbą całkowitą dodatnią będącą parametrem funkcji. Pary wybierane są tak, że pierwszy element sznura jest początkowym elementem pierwszej pary. Pamięć zajmowana przez usuwane elementy ma zostać zwolniona. Funkcja nie może używać tablic, pomocniczych sznurów ani struktur danych dostępnych w bibliotece standardowej (wektorów, list, kolejek, itp);

oraz program testujący działanie biblioteki poprzez pobranie sznura (funkcja wstaw; użytkownik podaje liczby, które mają być wstawione do sznura i decyduje o zakończeniu podawania), wypisanie jego zawartości (funkcja wypisz; przykład: dla zestawu liczb wejściowych 1,5,6,1,3,7,2,7,3,1,3,1 sznur uzyskuje postać 7,7,6,5,3,3,3,2,1,1,1,1), a następnie po pobraniu od użytkownika wartości K, usunięcie w sznurze rozłącznych par identycznych elementów z ewentualnymi dodatkowymi trzema (funkcja usunPlus) i ponowne wypisanie jego zawartości (przykład: dla K=2 sznur 7,7,6,5,3,3,3,2,1,1,1,1, zostanie przekształcony do postaci 6,5,1, dla K=3 sznur 9,9,8,7,6,5,5 stanie się pusty, dla K=1 sznur 9,7,6,5,4,4,4,2,1 zostanie przekształcony do postaci 9,7,6,5,4,2,1).

**Zad. 2:** Rozszerzyć bibliotekę z zad.1 o funkcję przeniesParzyste, która modyfikuje sznur poprzez przecięcie wszystkich (lub wszystkich poza środkowym) elementów sznura przechowujących wartości parzyste, naprzemiennie przed środek sznura i za środek sznura: pierwszą przed środek, następną za środek, kolejną przed środek, itd. W przypadku sznura nieparzystej długości środkowy element nie jest przepinany (nawet gdy zawiera wartość parzystą). Przepinanie elementów ma zachować ich kolejność względem tej w oryginalnym sznurze w ramach grupy przepinanej przed środek oraz odwrócić kolejność w ramach grupy przepinanej za środek sznura. Przykład: dla sznura przechowującego liczby 9,8,7,6,5,4,3,2,1,0,-1 dostaniemy wynik 9,7,5,8,2,4,0,6,3,1,-1, sznur 9,8,8,7,6,6,5,4 uzyska postać 9,7,8,6,4,6,8,5, sznur 8,6,4,2 przekształci się do 8,4,2,6. Funkcja nie może używać tablic, pomocniczych sznurów ani struktur danych dostępnych w bibliotece standardowej (wektorów, list, kolejek, itp). Wykorzystać funkcję w programie testującym.

**Zad. 3:** Rozszerzyć bibliotekę z zad.1 o funkcję przenies, modyfikującą dany sznur (oznaczony w dalszym opisie przez S1) oraz drugi sznur podany jako parametr (oznaczony dalej jako S2) w następujący sposób: rozpatrujemy kolejne podciągi elementów sznura S1: jednoelementowe jeśli dany element jest większy od K, lub nieprzedłużalne o sumie nie przekraczającej K; pierwszy, trzeci, piąty, itd. taki podciąg ma zostać przepięty do końcowej części sznura S2 złożonej z M (lub tylu ile jest gdy jest mniej) elementów (gdzie K i M są liczbami całkowitymi dodatnimi będącymi parametrami funkcji), przy czym pierwszy z przepinanych fragmentów sznura S1 ma zostać wpięty przed pierwszym elementem „M-końcówki”, kolejny przepinany fragment – przed drugim elementem „M-końcówki”, trzeci – przed jej trzecim elementem, itd.; a M+1-szy i kolejne przepinane fragmenty są wpinane za ostatnim elementem „M-końcówki” S2 (czyli za ostatnim elementem sznura) z zachowaniem kolejności fragmentów względem tej jaką miały w S1 (porządek elementów we fragmentach ma również zostać zachowany). Przykład: dla K=12, M=2 i sznurów S1=7,6,6,5,4,4,4,3,2,1 i S2=8,8,6,6,6,5,4,4,2,2 wynikiem będzie S1=6,6,4,4,4 i S2=8,8,6,6,6,5,4,4,7,2,5,4,2,3,2,1, dla K=2, M=1 i sznurów S1=7,6,6,5,4,4,4,3,2,1,1 i S2=8,8,6,6,6,5,4,4,2,2 wynikiem będzie S1=6,5,4,3,1,1 i S2=8,8,6,6,6,5,4,4,2,2,7,2,6,4,4,2. Funkcja nie może używać tablic, pomocniczych sznurów ani struktur danych dostępnych w bibliotece standardowej (wektorów, list, kolejek, itp). Wykorzystać funkcję w programie testującym.

## **Zestaw 4:**

**Organizacja plików:** Wszystkie pliki oddawane do sprawdzenia należy zapisać we wspólnym folderze o nazwie będącej numerem indeksu, umieszczonym na pulpicie.

**Oddajemy tylko źródła programów (pliki o rozszerzeniach .cpp i .h)!!!**

**Zad. 1:** Napisać bibliotekę udostępniającą:

- typ Sznur będący dynamiczną listą łączoną przechowującą liczby całkowite;
- funkcję wypisz wypisującą zawartość sznura;
- funkcję wstaw wstawiającą do sznura element zawierający liczbę podaną jako parametr tak, aby sznur był uporządkowany nierosnąco (zakładając, że sznur „na wejściu” jest już uporządkowany w taki sposób);
- funkcję usunCiagi usuwającą ze sznura ciągi sąsiednich elementów sznura w następujący sposób: usuwamy najdłuższy możliwy ciągły fragment taki, że suma jego wartości nie przekracza S (być może pusty), omijamy kolejnych K elementów, usuwamy najdłuższy możliwy ciągły fragment o sumie nieprzekraczającej S, omijamy K elementów, itd. (S i K są liczbami całkowitymi dodatnimi będącymi parametrami funkcji). Pamięć zajmowana przez usuwane elementy ma zostać zwolniona. Funkcja nie może używać tablic, pomocniczych sznurów ani struktur danych dostępnych w bibliotece standardowej (wektorów, list, kolejek, itp);

oraz program testujący działanie biblioteki poprzez pobranie sznura (funkcja wstaw; użytkownik podaje liczby, które mają być wstawione do sznura i decyduje o zakończeniu podawania), wypisanie jego zawartości (funkcja wypisz; przykład: dla zestawu liczb wejściowych 1,5,6,1,3,7,2,7,3,1,3,1 sznur uzyskuje postać 7,7,6,5,3,3,3,2,1,1,1,1), a następnie po pobraniu od użytkownika wartości S i K, usunięcie w sznurze ciągów o sumie nieprzekraczającej S z pominięciem K elementów między nimi (funkcja usunCiagi) i ponowne wypisanie jego zawartości (przykład: dla S=10 i K=2 sznur 7,7,6,5,3,3,3,2,1,1,1,1, zostanie przekształcony do postaci 7,6,3,3, dla S=100 i dowolnego K sznur 9,9,8,7,6,5,5 stanie się pusty, dla S=5 i K=1 sznur 9,7,6,5,4,4,2,1 zostanie przekształcony do postaci 9,7,6,4,4).

**Zad. 2:** Rozszerzyć bibliotekę z zad.1 o funkcję zmien, która modyfikuje sznur poprzez przepięcie wybranych elementów z rozłącznych trójek w następujący sposób: jeżeli elementy trójki przechowują trzy kolejne liczby całkowite, to takie trójki przepinane są na początek sznura z zachowaniem wzajemnej oryginalnej kolejności trójek, natomiast jeżeli trójka nie spełnia tego warunku, to elementy trójki przechowujące największą wartość w tej trójce (jeden, dwa lub trzy) przepinane są na koniec sznura z zachowaniem kolejności przenoszonych elementów (trójki wybierane są tak że pierwszy element sznura jest pierwszym elementem pierwszej trójki, a przypadku sznura o długości nie dzielącej się przez 3 jego końcowe elementy nie są elementami żadnej trójki). Przykład: dla sznura przechowującego liczby 11,10,9,9,9,8,8,6,6,6,5,4,4,4,3,2,2,1,1 dostaniemy wynik 11,10,9,6,5,4,8,6,6,3,1,1,9,9,8,4,4,2,2, sznur 9,8,7,5,4,3,2 nie zmieni się, a sznur 3,3,2,2,1,1 uzyska postać 2,1,1,3,3,2. Funkcja nie może używać tablic, pomocniczych sznurów ani struktur danych dostępnych w bibliotece standardowej (wektorów, list, kolejek, itp). Wykorzystać funkcję w programie testującym.

**Zad. 3:** Rozszerzyć bibliotekę z zad.1 o funkcję przenies, modyfikującą dany sznur (oznaczony w dalszym opisie przez S1) oraz drugi sznur podany jako parametr (oznaczony dalej jako S2) w następujący sposób: rozpatrujemy kolejne K-elementowe podciągi sznura S1; pierwszy, trzeci, piąty, itd. taki podciąg ma zostać przepięty do końcowej części sznura S2 złożonej z M (lub tylu ile jest gdy jest mniej) elementów (gdzie K i M są liczbami całkowitymi dodatnimi będącymi parametrami funkcji), przy czym pierwszy z przepinanych fragmentów sznura S1 ma zostać wpięty za ostatnim elementem „M-końcówki”, kolejny przepinany – przed ostatnim elementem „M-końcówki”, trzeci – przed jej przedostatnim elementem, itd.; a M+1-szy i kolejne przepinane fragmenty są wpinane przed pierwszym z elementów „M-końcówki” S2 z odwróceniem kolejności fragmentów względem tej jaką miały w S1 (porządek elementów we fragmentach ma zostać zachowany). W przypadku gdy długość S1 nie jest podzielna przez K końcowe elementy S1 (nie wchodzące w już skład „pełnego” K-elementowego podciągu) nie są przepinane. Przykład: dla K=3, M=2 i sznurów S1=7,6,6,5,4,4,4,3,2,1 i S2=8,8,6,6,6,5,4,4,2,2 wynikiem będzie S1=5,4,4,2,1 i S2=8,8,6,6,6,5,4,4,2,4,4,3,2,7,6,6, dla K=2, M=1 i sznurów S1=7,6,6,5,4,4,4,3,2,1 i S2=8,8,6,6,6,5,4,4,2,2 wynikiem będzie S1=6,5,4,4,1 i S2=8,8,6,6,6,5,4,4,2,3,2,4,4,2,7,6. Funkcja nie może używać tablic, pomocniczych sznurów ani struktur danych dostępnych w bibliotece standardowej (wektorów, list, kolejek, itp). Wykorzystać funkcję w programie testującym.

## **Zestaw 5:**

**Organizacja plików:** Wszystkie pliki oddawane do sprawdzenia należy zapisać we wspólnym folderze o nazwie będącej numerem indeksu, umieszczonym na pulpicie.

**Oddajemy tylko źródła programów (pliki o rozszerzeniach .cpp i .h)!!!**

**Zad. 1:** Napisać bibliotekę udostępniającą:

- typ Sznur będący dynamiczną listą łączoną przechowującą liczby całkowite;
- funkcję wypisz wypisującą zawartość sznura;
- funkcję wstaw dodającą na końcu sznura element zawierający liczbę podaną jako parametr;
- funkcję usunFragmenty usuwającą ze sznura ciągi sąsiednich elementów sznura w następujący sposób: usuwamy P elementów sznura (lub tyle ile jest, gdy jest mniej), następnie omijamy najdłuższy możliwy przynajmniej 2-elementowy ciągły fragment tworzący ciąg rosnący (jeśli taki fragment istnieje), usuwamy P elementów, omijamy ciąg rosnący, itd. (P jest liczbą całkowitą dodatnią będącą parametrem funkcji). Pamięć zajmowana przez usuwane elementy ma zostać zwolniona. Funkcja nie może używać tablic, pomocniczych sznurów ani struktur danych dostępnych w bibliotece standardowej (wektorów, list, kolejek, itp);

oraz program testujący działanie biblioteki poprzez pobranie sznura (funkcja wstaw; użytkownik podaje liczby, które mają być wstawione do sznura i decyduje o zakończeniu podawania), wypisanie jego zawartości (funkcja wypisz; przykład: dla zestawu liczb wejściowych 4,7,5,1,6,6,1,7,7,3,2,1,2,3,1 sznur uzyskuje postać 4,7,5,1,6,6,1,7,7,3,2,1,2,3,1), a następnie po pobraniu od użytkownika wartości P, usunięcie w sznurze P-elementowych ciągów z pominięciem ciągów rosnących między nimi (funkcja usunFragmenty) i ponowne wypisanie jego zawartości (przykład: dla P=3 sznur 4,7,5,1,6,6,1,7,7,3,2,1,2,3,1 uzyska postać 1,6,1,2,3, dla P=2 sznur 9,9,9,4,4,4,4,5,4,7,7,1,1,2 zostanie zredukowany do 4,5,1,2, dla P=1 sznur 1,1,1,1,1 stanie się pusty).

**Zad. 2:** Rozszerzyć bibliotekę z zad.1 o funkcję przenies, która modyfikuje sznur poprzez przepięcie wszystkich (lub wszystkich poza środkowym) elementów sznura przechowujących wartości parzyste, naprzemiennie przed środek sznura i za środek sznura: pierwszą przed środek, następną za środek, kolejną przed środek, itd. W przypadku sznura nieparzystej długości środkowy element nie jest przepinany (nawet gdy zawiera wartość parzystą). Przepinanie elementów ma odwrócić ich kolejność względem tej w oryginalnym sznurze w ramach grupy przepinanej przed środek oraz zachować kolejność w ramach grupy przepinanej za środek sznura. Przykład: dla sznura przechowującego liczby 9,8,7,6,5,2,3,2,1,0,-1 dostaniemy wynik 9,7,5,2,8,2,6,0,3,1,-1, sznur 1,8,8,7,6,6,5,4 uzyska postać 1,7,4,6,8,8,6,5, sznur 8,6,4,2 przekształci się do 4,8,6,2. Funkcja nie może używać tablic, pomocniczych sznurów ani struktur danych dostępnych w bibliotece standardowej (wektorów, list, kolejek, itp). Wykorzystać funkcję w programie testującym.

**Zad. 3:** Rozszerzyć bibliotekę z zad.1 o funkcję przeniesKGrupy, modyfikującą dany sznur (oznaczony w dalszym opisie przez S1) oraz drugi sznur podany jako parametr (oznaczony dalej jako S2) w następujący sposób: rozpatrujemy kolejne K-elementowe podciągi sznura S1; pierwszy, trzeci, piąty, itd. taki podciąg ma zostać przepięty do końcowej części sznura S2 złożonej z M (lub tylu ile jest gdy jest mniej) elementów (gdzie K i M są liczbami całkowitymi dodatnimi będącymi parametrami funkcji), przy czym pierwszy z przepinanych fragmentów sznura S1 ma zostać wpięty za ostatnim elementem „M-końcówki”, kolejny przepinany – przed ostatnim elementem „M-końcówki”, trzeci – przed jej przedostatnim elementem, itd.; a M+1-szy i kolejne przepinane fragmenty są wpinane przed pierwszym z elementów „M-końcówki” S2 z zachowaniem kolejności fragmentów względem tej jaką miały w S1 (porządek elementów we fragmentach ma zostać zachowany). W przypadku gdy długość S1 nie jest podzielna przez K końcowe elementy S1 (nie wchodzące w już skład „pełnego” K-elementowego podciągu) nie są przepinane. Przykład: dla K=3, M=2 i sznurów S1=3,6,6,1,4,2,4,5,3,2,1 i S2=3,8,6,4,6,5,2,4,1,2 wynikiem będzie S1=1,4,2,2,1 i S2=3,8,6,4,6,5,2,4,1,4,5,3,2,3,6,6, dla K=2, M=1 i sznurów S1=3,6,6,1,4,2,4,5,3,2,1 i S2=3,8,6,4,6,5,2,4,1,2 wynikiem będzie S1=6,1,4,5,1 i S2=3,8,6,4,6,5,2,4,1,4,2,3,2,2,3,6. Funkcja nie może używać tablic, pomocniczych sznurów ani struktur danych dostępnych w bibliotece standardowej (wektorów, list, kolejek, itp). Wykorzystać funkcję w programie testującym.

## **Zestaw 6:**

**Organizacja plików:** Wszystkie pliki oddawane do sprawdzenia należy zapisać we wspólnym folderze o nazwie będącej numerem indeksu, umieszczonym na pulpicie.

**Oddajemy tylko źródła programów (pliki o rozszerzeniach .cpp i .h)!!!**

**Zad. 1:** Napisać bibliotekę udostępniającą:

- typ Sznur będący dynamiczną listą łączoną przechowującą liczby całkowite;
- funkcję wypisz wypisującą zawartość sznura;
- funkcję wstaw dodającą na końcu sznura element zawierający liczbę podaną jako parametr;
- funkcję usun usuwającą ze sznura naprzemiennie co M-ty i co N-ty element (M i N są liczbami całkowitymi dodatnimi będącymi parametrami funkcji). Pamięć zajmowana przez usuwane elementy ma zostać zwolniona. Funkcja nie może używać tablic, pomocniczych sznurów ani struktur danych dostępnych w bibliotece standardowej (wektorów, list, kolejek, itp);

oraz program testujący działanie biblioteki poprzez pobranie sznura (funkcja wstaw; użytkownik podaje liczby, które mają być wstawione do sznura i decyduje o zakończeniu podawania), wypisanie jego zawartości (funkcja wypisz; przykład: dla zestawu liczb wejściowych 4,7,5,1,6,6,1,7,7,3,2,1,2,3,1 sznur uzyskuje postać 4,7,5,1,6,6,1,7,7,3,2,1,2,3,1), a następnie po pobraniu od użytkownika wartości M i N, usunięcie w sznurze naprzemiennie co M-tych i co N-tych elementów (funkcja usun) i ponowne wypisanie jego zawartości (przykład: dla M=2 i N=3 sznur 4,7,5,1,6,6,1,7,7,3,2,1,2,3,1 uzyska postać 4,5,1,6,7,7,2,2,3, dla M=2 i N=1 sznur 9,9,2,4,4,4,4,5,4,7,7,1,1,2 zostanie zredukowany do 9,4,4,7,1, dla M=N=1 sznur 1,2,6,3,4 stanie się pusty).

**Zad. 2:** Rozszerzyć bibliotekę z zad.1 o funkcję zmien, która modyfikuje sznur poprzez przepięcie wybranych elementów z rozłącznych trójek w następujący sposób: jeżeli elementy trójki przechowują równe wartości, to takie trójki przepinane są na koniec sznura z zachowaniem wzajemnej oryginalnej kolejności trójek, natomiast jeżeli trójka nie spełnia tego warunku, to elementy trójki przechowujące najmniejszą wartość w tej trójce (jeden lub dwa) przepinane są na początek sznura z zachowaniem kolejności przenoszonych elementów (trójki wybierane są tak że pierwszy element sznura jest pierwszym elementem pierwszej trójki, a przypadku sznura o długości nie dzielącej się przez 3 jego końcowe elementy nie są elementami żadnej trójki). Przykład: dla sznura przechowującego liczby 3,3,3,9,1,8,8,6,6,4,4,4,4,4,3,2,2,1,1 dostaniemy wynik 1,6,6,3,1,9,8,8,4,4,2,2,1,3,3,3,4,4,4, sznur 7,7,7,5,5,5 nie zmieni się, a sznur 3,3,2,2,1,1 uzyska postać 2,1,1,3,3,2. Funkcja nie może używać tablic, pomocniczych sznurów ani struktur danych dostępnych w bibliotece standardowej (wektorów, list, kolejek, itp). Wykorzystać funkcję w programie testującym.

**Zad. 3:** Rozszerzyć bibliotekę z zad.1 o funkcję przeniesPary, modyfikującą dany sznur (oznaczony w dalszym opisie przez S1) oraz drugi sznur podany jako parametr (oznaczony dalej jako S2) w następujący sposób: rozpatrujemy kolejne rozłączne pary elementów sznura S1; pierwsza, trzecia, piąta, itd. taka para ma zostać przepięta do końcowej części sznura S2 złożonej z M (lub tylu ile jest gdy jest mniej) elementów (gdzie M jest liczbą całkowitą dodatnią będącą parametrem funkcji), przy czym pierwsza z przepinanych par ze sznura S1 ma zostać wpięta przed pierwszym elementem „M-końcówki”, kolejna – przed drugim elementem „M-końcówki”, trzecia – przed jej trzecim elementem, itd.; a M+1-sza i kolejne przepinane pary są wpinane za ostatnim elementem „M-końcówki” S2 (czyli za ostatnim elementem sznura) z odwróceniem kolejności par względem tej jaką miały w S1; porządek elementów we wszystkich przepinanych parach ma zostać odwrócony. W przypadku gdy długość S1 jest nieparzysta końcowy element S1 nie jest przepinany. Przykład: dla M=2 i sznurów S1=3,6,6,1,4,2,4,5,3,2,1 i S2=3,8,6,4,6,5,2,4,1,2 wynikiem będzie S1=6,1,4,5,1 i S2=3,8,6,4,6,5,2,4,6,3,1,2,4,2,2,3, dla M=1 i sznurów S1=3,6,6,1,4,2,4,5,3,2,1 i S2=3,8,6,4,6,5,2,4,1,2 wynikiem będzie S1=6,1,4,5,1 i S2=3,8,6,4,6,5,2,4,1,6,3,2,2,3,2,4. Funkcja nie może używać tablic, pomocniczych sznurów ani struktur danych dostępnych w bibliotece standardowej (wektorów, list, kolejek, itp). Wykorzystać funkcję w programie testującym.