

WSKAŹNIKI

W każdym zadaniu, w którym trzeba napisać funkcję, przetestuj napisaną funkcję.

1. Napisz funkcję otrzymującą jako argumenty wskaźniki do dwóch zmiennych typu `int`, która zwraca jako wartość mniejszą ze wskazywanych liczb.
2. Napisz funkcję otrzymującą jako argumenty wskaźniki do dwóch zmiennych typu `int`, która zamienia ze sobą wartości wskazywanych zmiennych.
3. Napisz funkcję, której argumentami są dwa wskaźniki do stałych typu `int`, zaś zwracaną wartością jest suma wartości zmiennych wskazywanych przez argumenty.
4. Napisz funkcję, której argumentami są `n` typu `int` oraz `w` wskaźnik do `int`, która przepisuje wartość `n` do zmiennej wskazywanej przez `w`.
5. Napisz funkcję otrzymującą dwa argumenty referencję `a` oraz wskaźnik `b` do zmiennych typu `int`, która zamienia ze sobą wartości zmiennych, do których wskaźnik i referencję dostała w argumentach.
6. Napisz bezargumentową funkcję, która rezerwuje pamięć dla pojedynczej zmiennej typu `int` i zwraca jako wartość wskaźnik do niej.
7. Napisz funkcję, która dostaje jako argument dodatnią liczbę całkowitą `n`, rezerwuje w pamięci blok `n` zmiennych typu `int` i zwraca jako wartość wskaźnik do początku zarezerwowanego bloku pamięci.
8. Napisz funkcję, która dostaje dwa argumenty: wskaźnik na stałą typu `int` i stały wskaźnik na zmienną typu `int`, i przepisuje zawartość stałej wskazywanej przez pierwszy argument do zmiennej wskazywanej przez drugi argument.
9. (*) Napisz funkcję o dwóch argumentach:
 - wskaźnik na funkcję o jednym argumencie typu `double` zwracającą wartość typu `double`,
 - wartość typu `double`,która zwraca wartość funkcji otrzymanej w pierwszym argumencie na liczbie podanej w drugim argumencie.
10. (*) Napisz funkcję, która otrzymuje trzy argumenty:
 - dwa wskaźniki na funkcje o jednym argumencie typu `int` zwracające wartość typu `int`,
 - wartość `n` typu `int`,i zwraca `true`, jeżeli otrzymane w argumentach funkcje są równe dla wartości od 0 do `n` i `false` w przeciwnym wypadku.

KONTENER LIST<T>

11. Przeczytaj opis metody `insert` klasy `list` oraz przeanalizuj przykład na stronie:
<https://www.cplusplus.com/reference/list/list/insert/>
12. Przeczytaj opis metody `erase` klasy `list` oraz przeanalizuj przykład na stronie:
<https://www.cplusplus.com/reference/list/list/erase/>
13. Napisz program, który wczytuje ze standardowego wejścia dodatnią liczbę całkowitą `n` a następnie wczytuje `n` liczb całkowitych i wstawia je do listy. Liczby parzyste program powinien wstawiać na koniec listy, a nieparzyste na początek listy. Na końcu program powinien wypisać na standardowym wyjściu wszystkie wczytane elementy w kolejności ich występowania na liście. W rozwiązaniu wykorzystaj kontener `list`.

14. Napisz program, który wczytuje ze standardowego wejścia dodatnią liczbę całkowitą n a następnie wczytuje n liczb całkowitych i wstawia je do listy. Liczby dodatnie program powinien wstawiać na koniec listy, ujemne na początek listy, a zera w środek (pomiędzy ujemne, a dodatnie). Na końcu program powinien wypisać na standardowym wyjściu wszystkie wczytane elementy w kolejności ich występowania na liście. W rozwiązaniu wykorzystaj kontener `list`.
15. Napisz program, który wczytuje ze standardowego wejścia dodatnią liczbę całkowitą n a następnie wczytuje n liczb całkowitych i wstawia je do listy. Następnie program usuwa z listy wszystkie zera (możesz w tym celu wykorzystać metodę `remove`; znajdź jej opis). Na końcu program powinien wypisać na standardowym wyjściu wszystkie wczytane elementy w kolejności ich występowania na liście. W rozwiązaniu wykorzystaj kontener `list`.
16. Napisz program, który wczytuje ze standardowego wejścia dodatnią liczbę całkowitą n a następnie wczytuje n liczb całkowitych i wstawia je do listy. Następnie program usuwa z listy wszystkie liczby parzyste (możesz w tym celu wykorzystać metodę `remove_if`). Na końcu program powinien wypisać na standardowym wyjściu wszystkie wczytane elementy w kolejności ich występowania na liście. W rozwiązaniu wykorzystaj kontener `list`.
17. Napisz program, który wczytuje ze standardowego wejścia dodatnią liczbę całkowitą n a następnie wczytuje n liczb całkowitych i wstawia je do listy. Następnie program usuwa z listy co drugi element (tzn. drugi, czwarty, itd.) Na końcu program wypisuje na standardowym wyjściu wszystkie elementy, które zostały na liście (w kolejności ich występowania na liście). W rozwiązaniu wykorzystaj kontener `list`.
18. (*) Rozważmy następujący problem Flawiusza:

W ogólnej wersji problem brzmi następująco: na okręgu ustawiamy n obiektów, następnie eliminujemy co k -ty obiekt, tak długo, aż zostanie tylko jeden. Należy wskazać obiekt, który pozostanie.

Napisz program, który rozwiązuje ten problem (dla zadanego n i k) poprzez przeprowadzenie symulacji. Skorzystaj z kontenera `list`.
19. Napisz program, który wczytuje ze standardowego wejścia dodatnią liczbę całkowitą n a następnie wczytuje n liczb całkowitych i wstawia je do listy. Następnie program usuwa wszystkie powtórzenie, tzn. dla każdej liczby, która się pojawi, zostawia na liście tylko jej jedną kopię. Możesz w tym celu wykorzystać metody `sort` i `unique`.
20. Czym się różni kontener `forward_list` od kontenera `list`? Odpowiedź znajdziesz na stronie https://www.cplusplus.com/reference/forward_list/forward_list/

BIBLIOGRAFIA

1. Krzaczkowski, J., Zadania z programowania w języku C/C++, cz.I, Instytut Informatyki UMCS, Lublin, 2011.
2. Krzaczkowski, J., Zbiór zadań z programowania w języku C/C++ cz. II, Instytut Informatyki UMCS, Lublin, 2012.
3. <https://www.cplusplus.com/reference/>
4. https://pl.wikipedia.org/wiki/Problem_Józefa_Flawiusza