Funkcje szablony. Funktorzy

Zadanie 1. Wykonane w dwóch wariantach wykonania. Zadanie dalej podano w indywidualnych zadaniach.

Zadanie 1. Wariant 1. Utwórz dwie parametryczne (szablonowe) funkcje, które wykonują akcje w tablicy, umieść wynik w tej samej tablicy. Sprawdź dwie tablice różnych typów (na przykład całkowitą i rzeczywista). Wybierz typy tablic w zależności od warunków indywidualnogo zadania lub tablice wykonuj tylko te działania, które mają sens dla tego typu danych. Aby wyprowadzić tablicę wyników, użyj funkcji szablonu printArray.

Zadanie 1. Wariant 2. Utwórz jedną sparametryzowaną (szablonową) funkcję, która wykonuje akcje przez tablicę. Ustawienia szablonów muszą mieć domyślną wartość (jest to możliwe dla wszystkich typów, typów zmiennych i funktorów). Pierwszym parametrem szablonu jest utworzenie rozmiaru tablicy, typu tablicy i funktora wykonującego operację. Funkcja musi przyjąć liczbę K do odpowiedniego typu i mieć domyślną wartość dla tej liczby. Sprawdź dwie tablice różnych typów danych statycznie powodujących każdą funkcję szablonu z obydwoma funktorami (jeśli to możliwe). Aby wyprowadzić tablicę wyników, użyj funkcji szablonej printArray.

Indywidualne zadania.

1. Suma K i element tablicy to iloczyn K i element tablicy.
2. Iloczyn K jest elementem tablicy i różnicą K oraz elementem tablicy.
3. Element tablicy w potęgie K i wynik dzielenia elementu tablicy na K (rzeczywisty lub całkowity).
4. Reszta od dzielenia elementu tablicy na K i cosinus sumy elementu tablicy i K.
5. Wartość bezwzględna sumy elementu tablicy i K, pierwiastek kwadratowy z wartości bezwzględnej sumy elementu tablicowego i K.
6. Wartość bezwzględna różnicy między elementem tablicy a K i logarytmem bezwzględnej wartości różnicy między elementem tablicy a K.
7. Suma elementu tablicy i K oraz sinusa sumy elementu tablicy i K.
8. Iloraz od dzielenia K na element tablicy oraz różnica K i elementu tablicy.
9. Iloczyn elementu tablicy i K oraz logarytmu wartości bezwzględnej iloczynu elementu tablicy i K.
10. Różnica między K i elementem tablicy oraz eksponenta różnicy K i elementem tablicy.
11. Wartość bezwzględna różnicy między elementem tablicy a K i pierwiastkiem kwadratowym różnicy bezwzględnej między elementem tablicy a K.
12. Element tablicy w potęgie K i K w potęgie elementu tablicy.
13. Iloczyn elementu tablicy i K oraz cosinusa ilczyna wartości bezwzględnej elementu tablicy i K
14. Iloraz dzielenia elementu tablicy na K (rzeczywistą lub całkowitą) i eksponenta iloczyna elementu tablicy na K.
15. Iloczyn elementu tablicy na (1 - K) i sinusa z sumy elementu tablicy i K.

Zadanie 2. Zadanie dalej podano w indywidualnych zadaniach.

Zadanie 2. W tym zadaniu zaimplementuj dwa uogólnione algorytmy do pracy z różnymi typami tablic za pomocą sparametryzowanych funkcji dla iteratorów kontenerów wejściowych i wyjściowych oraz funktora dla drugiego zadania w przykładzie copy\_if.

Indywidualne zadania.

1. Zaimplementuj algorytmy minelement () i min\_element\_if (), które zwracają minimalny element tablicy.

2. Zaimplementuj algorytmy count () i count\_if (), które zwracają liczbę elementów tablicy.

3. Zaimplementuj algorytmy search() i search\_if () pierwsze wystąpienie sekwencji elementów w innej sekwencji

4. Wprowadź algorytmy wyszukiwania search\_end () i search\_end\_if () ostatniego wpisu sekwencji elementów do innego.

5. Wdrożone algorytmy usuwania erase\_copy () i erase\_copy\_if () są kopiowane do kontenera wyjściowego.

6. Uruchom scalanie () i merge\_if () scalanie posortowanych sekwencji. Wynik jest zapisywany w oryginalnej sekwencji.

7. Zaimplementuj algorytmy do ustawiania set\_or\_if () i łączenia sekwencji w zestawy (sekwencyjnie). Wynik jest zapisywany w oryginalnej sekwencji.

8. Wdrożyć algorytmy, aby ustawić set\_and () i set\_and\_if () przecięcie sekwencji jako zestawy. Wynik jest zapisywany w oryginalnej sekwencji.

9. Zaimplementuj algorytmy, aby ustawić odejmowanie symetryczne set\_xor () i set\_xor\_if () sekwencji jako zestawy. Wynik jest zapisywany w oryginalnej sekwencji.

10. Opracuj algorytm dla for\_each (), który stosuje dany funktor do każdego elementu sekwencji. Wykonaj generate (), który używa danego funktora do wypełnienia elementów kontenera.

11. Zaimplementuj algorytmy accumulate () i accuaulate\_if (), które wykonują nagromadzenie elementów sekwencji. Metoda akumulacji jest określona przez funktora przekazanego jako argument.

12. Zaimplementuj binoperate () i binoperate\_if (), które wykonują operację dwóch przedziałów z danym funktorem i zapisz wynik w sekwencji wyjściowej.

13. Wdrożyć algorytmy unique\_copy () i unique\_copy\_if (), aby usunąć zduplikowane elementy z sekwencji. Wynik zostanie zapisany w sekwencji wyjściowej.

14. Implementuj reverse\_copy () i reverse\_copy\_if () dla odwrotności