Main ()

Otwórz plik „wyniki.txt”

Wyświetl „wybierz tryb:”

Wyświetl „ 1. Liczby z klawiatury”

Wyświetl „ 2. Liczby z pliku”

Wyświetl „ 3. Testuj”

Choice ← getch()

Wybierz ← choice

Gdy wpisano „1”

Wypisz „Podaj ile liczb”

Wprowadź n

Wypisz „podaj” n „liczb”

Dla funkcji read\_data(cin,n) przypisz name

Przerwij

Gdy wpisano ‘2’

Wypisz „podaj nazwe pliku: ”

Wpisz nazwe pliku

Dla funkcji read\_data(file\_name,-1) przypisz nums

Przerwij

Gdy wpisano ‘3’

Uruchom funkcje run\_tests

Zwróć 0

Dla pozostałych przypadkow:

Wypisz „bledny wybor”

nums\_sorted = sort\_by\_first\_appearance(nums)

przypisz nums\_sorted.begin() dla it

wykonuj dopóki it nie będzie się równać nums\_sorted.end()

Wypisz „do posortowania: ”

Dla funkcji display\_vec argumentem będzie nums

Wypisz „po pososortowaniu ”

Dla funkcji display\_vec argumentem będzie nums\_sorted

Vector read \_data()

Vector ret

For i ←1 to i<n albo n<0

Do:

if (in >> x)

rozszerz wektor ret o zmienna x na końcu tego wektora

przerwij

zwróć ret

vector sort\_by\_first\_appearance

vector ret=vec

deklaracja mapy first\_seen

for element to vec

do first\_seen.instert dolacz element

posortuj za pomoca funkcji sort zapisanej w bibliotece algorithm mape tak, aby pierwszy elelment wystepowal jako pierwszy

zwróć ret

display\_vec

for v do vec

wypisz v

wypisz „ ”

run\_tests()

Do wektora a przypisz {4,2,4,11,4,5,2,3,1,2,1}

Do b przypisz {4,4,4,2,2,2,11,5,3,1,1}

Do zmiennej a\_sorted przypisz sort\_by\_first\_appearance z argumentem a

Wypisz „do posortowania”

Dla funkcji display\_vac przypisz wektor a

Wypisz „Po posortowaniu”

Do funkcji display\_vec przypisz zmienna a\_sorted

Wypisz „ Oczekiwano”

Jeśli a\_sorted==b wypisz true, w przeciwnym wypadku false