



POLSKO-JAPOŃSKA AKADEMIA TECHNIK KOMPUTEROWYCH

PJC Zadania 3

Rozwiązania należy przesłać w postaci **pojedynczego pliku** o rozszerzeniu `.cpp`

UWAGA: W poniższych rozwiązaniach **nie wolno używać własnoręcznie napisanych pętli lub rekurencji**. Przy okazji **nie wolno korzystać z algorytmu `std::for_each`**.

Zadanie 1

Zaimplementuj funkcję `contains_needle()` przyjmującą dwa argumenty typu `std::string`. Funkcja winna zwrócić prawdę lub fałsz, zależnie od tego, czy w pierwszym przekazanym tekście znajduje się drugi tekst. Zadanie można rozumieć jako problem *igły w stogu siana* (ang. *needle in the haystack*), gdzie pierwszy argument to stóg siana, w którym szukamy igły, czyli drugiego argumentu.

Zadanie 2

Zaimplementuj funkcję `contains_all()` przyjmującą dwa argumenty typu `std::vector<int>`. Funkcja winna zwrócić prawdę lub fałsz, zależnie od tego, czy w pierwszym przekazanym kontenerze znajdują się wszystkie elementy z drugiego przekazanego kontenera. Zwróć uwagę na krotność elementów.

Zadanie 3

Napisz funkcję `exchange_boundaries()`, która na pobranym przez argument wektorze liczb całkowitych zamieni miejscami jego największy i najmniejszy element. Załóż, że wektor nie jest pusty. W funkcji nie wolno wprowadzać żadnych dodatkowych zmiennych. Zwróć zmodyfikowany wektor.

Zadanie 4

Napisz funkcję `copy_by_pushing()` przyjmującą wektor liczb całkowitych (v), liczbę mówiącą ile elementów chcemy skopiować (n) oraz liczbę mówiącą o ile chcemy przesunąć kopiowane elementy (p). Następnie, funkcja winna skopiować i przesunąć pierwszych n elementów z v i wkleić je po kolei p miejsc dalej. Przykładowo:

```
auto result = copy_by_pushing({0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, 3, 5);
```

Początkowy wektor: **0 1 2** 3 4 5 6 7 8 9 10

Wartości z result: 0 1 2 3 4 **0 1 2** 8 9 10

Jak widać, wzięliśmy **3** (drugi argument) **pierwsze elementy** a następnie skopiowaliśmy je i wkleiliśmy poczynając od pozycji numer **5** (trzeci argument). Efektywnie *przepchnęliśmy pierwsze 3 elementy o 5 miejsc w prawo*.

Przykład 1

```
auto result = copy_by_pushing({0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, 5, 2);
```

Początkowy wektor: **0 1 2 3 4** 5 6 7 8 9 10

Wartości z result: 0 1 **0 1 2 3 4** 7 8 9 10

Jak widać, wzięliśmy **5 pierwszych elementów** a następnie skopiowaliśmy je i wkleiliśmy poczynając od pozycji numer **2**. Efektywnie *przepchnęliśmy pierwsze 5 elementów o 2 miejsca w prawo*.

Przykład 2

W tym zadaniu alokacja dodatkowego kontenera będzie skutkowała obcięciem 50% punktów.

Zadanie 5

Napisz funkcję `translate_to_Kwiatkovsky()`, która przyjmie argument typu `std::string` i zwróci go tak zmodyfikowanego, że każde wystąpienie litery R (małe lub duże) zostanie zastąpione literą L (zachowując wielkość liter).