



# POLSKO-JAPOŃSKA AKADEMIA TECHNIK KOMPUTEROWYCH

## PJC Zadania 2

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

### Zadanie 1

Zaimplementuj funkcję `modify()`, która przyjmie przez argument `vector<int>` oraz string będący poleceniem. Załóż, że polecenie może przyjąć jedynie następujące treści:

- "append", którego skutkiem będzie dodanie do końca przekazanego wektora dowolnej, wybranej przez studenta liczby;
- "truncate", którego skutkiem będzie usunięcie z przekazanego wektora ostatniej wartości;
- "reverse", którego skutkiem będzie odwrócenie przekazanego wektora.

Funkcja następnie powinna **zwrócić** przekazany (i już zmodyfikowany) wektor.

**Domyślną wartością** argumentu reprezentującego polecenie winna być opcja "reverse".

Corner-case, gdzie wysyłamy pusty wektor i precyzujemy operację "truncate" można zignorować.

### Zadanie 2

Napisz funkcję `count_chars()`, która przyjmie `vector<char>` oraz pojedynczego chara. Funkcja winna zwrócić liczbę wystąpień danego znaku w przekazanym wektorze.

Dodaj przeciążenie funkcji `count_chars()`, które przyjmie jako argument string zamiast `vector<char>`.

## Zadanie 3

Napisz funkcję `box_print()`, która przyjmie przez argument `vector<string>` i wyświetli każdy z przekazanych ciągów tekstowych w ramce zrobionej ze znaków asterisk (\*).

Przykład:

```
auto strings = vector<string>{"Hello", "world", "in", "a", "frame"};
box_print(strings);
```

Powinien wyświetlić:

```
*****
*Hello*
*world*
*in   *
*a    *
*frame*
*****
```