



# POLSKO-JAPOŃSKA AKADEMIA TECHNIK KOMPUTEROWYCH

## PJC Zadania 4

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

**UWAGA:** W poniższych rozwiązaniach użycie własnoręcznie napisanych pętli lub rekurencji będzie skutkować obniżeniem oceny o 50%. Dodatkowo nie wolno korzystać z `std::for_each()`.

### Zadanie 1

Zaimplementuj funkcję `get_male_names_from()`, która przyjmie `std::vector` przechowujący elementy typu `std::string`, z którego wyciągnięte zostaną wyłącznie męskie imiona i zwróci je wszystkie w postaci innego wektora, posortowanego po długości imion, a w przypadku elementów o takich samych długościach kolejność winna być determinowana przez kolejność alfabetyczną.

### Zadanie 2

Napisz funkcję `reduce_adjacent_whitespaces()`, która przyjmie zmienną typu `std::string` i zwróci ją tak zmodyfikowaną, że wszystkie ciągi sąsiadujących białych znaków z niej zostaną zredukowane do jednej spacji. Przykład:

Następujący kod:

```
std::cout << '['
          << reduce_adjacent_whitespace("Abc")
          << "]\n"
          << '['
          << reduce_adjacent_whitespace("A b c")
          << "]\n"
          << '['
          << reduce_adjacent_whitespace("  A b c  ")
          << "]\n"
          << '['
          << reduce_adjacent_whitespace("A\t\tb\n\n\t c")
          << "]\n";
```

Wyświetli:

[Abc]

[A b c]

[ A b c ]

[A b c]

Jak widać, wszelkie ciągi sąsiadujących białych znaków są redukowane do pojedynczej spacji.

*Przykład 1*

## Zadanie 3

Napisz funkcję `max_nested_parentheses_depth()`, która przyjmie `std::string` i zwróci maksymalny poziom zagnieżdżenia nawiasów okrągłych w nim. Załóż, że nawiasy są poprawnie sparowane, czyli nie ma miejsca aby, przykładowo, funkcja otrzymała następujący argument: `)())(`. Przykład:

Następujący kod:

```
std::cout << max_nested_parentheses_depth("()") << ' '
          << max_nested_parentheses_depth("(()())") << ' '
          << max_nested_parentheses_depth("(sd()ds((ss)) a)") << ' '
          << max_nested_parentheses_depth("((S))") << ' '
          << max_nested_parentheses_depth("()()()()") << ' '
          << max_nested_parentheses_depth("()((())())");
```

Wyświetli:

1 3 3 3 1 2

Jak widać, ignorowane są znaki poza nawiasami i zliczane jest ich największe zagłębienie.

*Przykład 2*