



POLSKO-JAPONSKA AKADEMIA TECHNIK KOMPUTEROWYCH

PRG1 Zadania 2

Rozwiązania należy przesłać w postaci **pojedynczego** pliku o rozszerzeniu `.cpp` (najczęściej `main.cpp`).

Niektóre z poleceń mogą sugerować używanie polskich nazw w kodzie, ale nie jest to celowe – najlepiej jest stosować wszędzie nazwy po angielsku.

Zadanie 1

Napisz funkcję `average_of()`, która przyjmie wektor liczb rzeczywistych i zwróci ich średnią arytmetyczną.

Zadanie 2

Napisz funkcję `filter_palindromes()` przyjmującą przez argument `std::vector<std::string>` i zwracającą (również w postaci takiego wektora) tylko te stringi z przekazanego wektora, które są palindromami.

Zadanie 3

Napisz funkcję `dot_product()`, która pobierze przez argument dwa wektory liczb rzeczywistych i zwróci ich iloczyn skalarny. Zadbaj o to, aby zbadać, czy te wektory są na pewno tej samej wielkości.

Zadanie 4

Stwórz strukturę `student`, która będzie przechowywała następujące pola:

- imię
- nazwisko
- oceny

To ostatnie pole zrealizuj za pomocą wektora liczb rzeczywistych.

Napisz funkcję `average_of()`, która przyjmie przez argument studenta (obiekt struktury `student`) oraz zwróci średnią jego ocen. Czy duplikacja nazwy funkcji (z pierwszego zadania) powoduje błąd?

Zadanie 5

Napisz funkcję `split()`, która przyjmie przez argument `std::string` i zwróci `std::vector<std::string>`. Załóż, że przekazany argument przechowuje jakieś zdanie (słowa oddzielone spacjami). Funkcja ma zwrócić wektor tych słów.

Na przykład wywołanie `split("ała ma kota")` powinno zwrócić wektor o następującej zawartości: `{"ała", "ma", "kota"}`.

Zadanie 6

Napisz funkcję `get_best_student()`, która przyjmie przez argument wektor studentów i zwróci obiekt tego studenta, którego średnia jest najwyższa. W przypadku kilku najlepszych studentów zwróć któregośkolwiek z nich.

Zadanie 7

Stwórz strukturę `group`, która będzie miała pola:

- `nazwa`
- `studenci`

To ostatnie pole zrealizuj za pomocą wektora przechowującego studentów.

Napisz funkcję, która przyjmie przez argument grupę studencką i pojedynczego studenta. Jej zadaniem będzie dodanie tego studenta do tej grupy. Nadaj funkcji jak najlepszą nazwę, która wskazuje jej użycie. Nic z niej nie zwracaj.

Zadanie 8

Napisz funkcję `best_groups_name()`, która przyjmie wektor grup studenckich (struktury z poprzedniego zadania). Funkcja ma zwrócić nazwę najlepszej grupy studenckiej. Najlepsza grupa studencka to jest taka, której mediana średnich ocen wszystkich studentów jest najwyższa. W przypadku kilku najlepszych grup zwróć którąkolwiek z nich.

Zadanie 9

Napisz funkcję `contains_all()`, która pobierze przez argument dwa wektory. Jej zadaniem będzie sprawdzenie, czy pierwszy wektor zawiera wszystkie elementy z drugiego wektora (czyli drugi zbiór jest podzbiorem pierwszego). Zwróć uwagę na duplikaty – są one ważne.

Zadanie 10

Napisz program, w którym poprosisz użytkownika o wprowadzanie dowolnej liczby par znaków. Zakończ pobieranie tych znaków, jak użytkownik wpisze `stop`. Następnie, korzystając z wektora stringów, stwórz wektor *zakresów*, a potem go wyświetl.

Przykładowo, jeżeli użytkownik wprowadzi następujące znaki:

```
a b
b a
a z
z a
d h
k v
x x
stop
```

To powinien zostać stworzony wektor zawierający następujące stringi:

```
ab
ab
abcdefghijklmnopqrstuvwxy
abcdefghijklmnopqrstuvwxy
defgh
klmnopqrstuv
x
```

Podpowiedź: wykorzystaj funkcję `std::getline()` jako warunek pętli.