

## Algorytmy i struktury danych I

## Zestaw zadań nr. 5: Listy (c.d.)

**Zadania programowe** przygotowujemy w postaci kodu programu oraz dokumentu z krótkim opisem kodu oraz wynikami testów: plik.pdf (używamy latex, word) i umieszczamy w systemie Pegaz.

- Zadanie 1 (programowe)  
Uzupełnij swoją implementację listy opartą o wskaźniki o metody dostępne w bibliotece STL dla listy **class forward\_list**.  
Napisz kod do generacji testowego inputu oraz kod do generowania wzorcowego outputu w oparciu o implementację **class forward\_list** w bibliotece STL.
- Zadanie 2 (programowe)  
Na wejściu mamy podaną ilość zestawów testowych  $z$ , a następnie  $z$  zestawów testowych, z których każdy składa się z 4 linii:
  - w pierwszej linii podana jest liczba  $s_1$ , ilość elementów w pierwszej liście
  - w drugiej linii podane jest  $s_1$  liczb, kolejne elementy listy (posortowane)
  - w trzeciej linii podana jest liczba  $s_2$ , ilość elementów w drugiej liście
  - w czwartej linii podane jest  $s_2$  liczb, kolejne elementów listy (posortowane)

Dla każdego zestawu:

1. Wypisz na wyjście listę powstałą po scaleniu dwóch list (z powtarzającymi się elementami), t.j. wypisz wszystkie jej elementy (posortowane)
2. Wypisz na wyjście listę powstałą po scaleniu dwóch list bez powtarzających się elementów t.j. wypisz wszystkie jej elementy (posortowane)

Dla testowego przykładu:

```
1
5
1 2 3 4 5
2
2 9
```

Program powinien wypisać:

```
1 2 2 3 4 5 9
1 2 3 4 5 9
```

Program powinien działać w czasie liniowym od długości list. Zadanie rozwiąż przy użyciu własnej implementacji listy.

Do generacji testowego inputu użyj programu **GenListUnion.cpp**, dla wygenerowania wzorcowego outputu użyj programu **ListUnion.cpp**. Przetestuj swoją implementację porównując z wzorcowym outputem.