

Zadanie 1.

Dana jest definicja elementu listy jednokierunkowej przechowującej wartości typu int.

```
struct elem
{
    int dane;
    elem* nast;
};
```

Napisać funkcję

```
void usun(elem* &glowa);
```

która z listy usunie wszystkie elementy, których wartość w polu *dane* występuje dokładnie jeden raz. Do funkcji przekazywany jest wskaźnik do głowy listy.

Zadanie 2.

Dana jest definicja elementu listy jednokierunkowej przechowującej wartości typu int.

```
struct elem
{
    int dane;
    elem* nast;
};
```

Napisać funkcję

```
void usun(elem* &glowa);
```

która z listy usunie duplikaty (elementy, których wartości w polu *dane* powtarzają się). Po wykonaniu funkcji lista powinna zawierać wyłącznie jedno wystąpienie danej wartości.

Zadanie 3.

Dana jest definicja elementu listy jednokierunkowej:

```
struct elem
{
    int dane;
    elem* nast;
};
```

Lista reprezentuje zbiór wartości typu int. Napisać funkcję

```
elem* iloczyn(elem* &zbiior1, elem* &zbiior2);
```

która wyznacza iloczyn zbiorów *zbiior1* i *zbiior2* reprezentowanych przez dwie listy. Wskaźnik do zbioru wynikowego (reprezentowanego przez listę) zwracany jest jako wynik wykonania funkcji. Wewnątrz funkcji nie wolno używać funkcji i operatora przydzielającego pamięć. Po wyjściu z funkcji zbiory *zbiior1* i *zbiior2* mają być puste.

Zadanie 4.

Dana jest definicja elementu listy jednokierunkowej:

```
struct elem
{
    int dane;
    elem* nast;
};
```

Napisać funkcję

```
elem* polacz_listy(elem* &lista1, elem* &lista2);
```

która połączy dwie listy, których wskaźniki do głowy przekazywane są do funkcji jako *lista1* i *lista2*, w jedną listę wynikową i wskaźnik do pierwszego elementu listy wynikowej zwróci jako wynik wykonania. Elementy w obydwu listach są posortowane w porządku niemalejącym według pola *dane*. Elementy w liście wynikowej także mają być posortowane w porządku niemalejącym według pola *dane*.

Wewnątrz funkcji nie wolno używać funkcji i operatora przydzielającego pamięć. Po wyjściu z funkcji listy *lista1* i *lista2* mają być puste.