

Kod źródłowy rozwiązania każdego z zadań zapisz z oddzielnym pliku *.cpp. Programy nie kompilujące się otrzymują **zero** punktów. Można korzystać z Internetu, jednak zabronione są wszelkie formy kontaktowania się. Powodzenia!

Zadanie 1. (10 pkt) Napisz klasę `Naukowiec` zawierającą nazwisko, liczbę publikacji oraz cytowań naukowca. Klasa powinna zapewniać następujące operacje:

- konstruktor trójargumentowy inicjalizujący nazwisko, liczbę publikacji oraz cytowań,
- funkcję składową `indeks` zwracającą średnią liczbę cytowań na jedną publikację,
- funkcję składową `publikuj` zwiększającą liczbę publikacji o zadaną wartość,
- operator preinkrementacji `++` zwiększający o 1 liczbę cytowań naukowca,
- operator `<` porównujący naukowców. Lepszy jest naukowiec z większą liczbą cytowań, a jeżeli są takie same, to ten dla którego funkcja `indeks` zwraca większą wartość.
- operator `<<` wypisywania danych naukowca do strumienia typu `ostream` oraz operator `>>` wczytywania danych naukowca ze strumienia `istream`.

PRZYKŁAD UŻYCIA KLASY:

```
int main() {
    Naukowiec Bogdan("Bogdan",50,300), Czeslaw;
    cin >> Czeslaw;
    cout << Czeslaw.indeks();
    Bogdan.publikuj(3);
    if(Bogdan < ++Czeslaw)
        cout << Czeslaw << endl;
    else
        cout << Bogdan << endl;
}
```

Zadanie 2. (10 pkt) W plikach tekstowych każdy znak zapisany jest jako jednobajtowa liczba, zwana kodem ASCII znaku. Jedną z metod szyfrowania wiadomości jest zastąpienie kodu każdego znaku jego bitową różnicą symetryczną (XOR) z zadaną jednobajtową liczbą, zwaną kluczem. Aby odczytać wiadomość, wystarczy ponownie wziąć bitową różnicę symetryczną zaszyfrowanych kodów z tym samym kluczem. Zaszyfrowane wiadomości są zapisywane przy pomocy liczb oddzielonych spacjami. Szyfrowanie odbywa się znak po znaku, włączając spacje, tabulatory, znaki końca linii, itp. Napisz program szyfrujący i rozszyfrowujący wiadomości tekstowe przy pomocy różnicy symetrycznej z kluczem. Wiadomości, zarówno zaszyfrowane jak i niezaszyfrowane, zapisane są w plikach tekstowych.

Program powinien przyjmować cztery argumenty wywołania. Pierwszy to klucz, drugi to e lub d odpowiednio dla szyfrowania i rozszyfrowywania, zaś trzeci i czwarty to nazwy plików wejściowego i wyjściowego. *Wskazówka*: operator XOR zapisujemy przy pomocy znaku ^.

PRZYKŁADOWA WIADOMOŚĆ W PLIKU WIAD.TXT:

```
Ala ma kota  
A to As.
```

ZAKODOWANIE Z KLUCZEM 45 POLEGA NA WYWOŁANIU POLECENIA ./zad2 45 e wiad.txt szyfr.txt KTÓRE TWORZY PLIK SZYFR.TXT O ZAWARTOŚCI:

```
108 65 76 13 64 76 13 70 66 89 76 39 108 13 89 66 13 108 94 3
```

PO PONOWNYM ODKODOWANIU POLECENIEM ./zad2 45 d szyfr.txt wiad2.txt DOSTAJEMY PLIK WIAD2.TXT O ZAWARTOŚCI:

```
Ala ma kota  
A to As.
```

Zadanie 3. (10 pkt) Pan Antoni jest właścicielem gaju. Znajduje się w nim n drzew o współrzędnych (x_i, y_i) . Aby nikt niepowołany nie mógł dostać się na teren gaju, zamierza ogrodzić go prostokątnym płotem. Ze względu na oszczędności płot powinien mieć możliwie najmniejszy obwód, jednocześnie obejmując wszystkie n drzew. Napisz program, który wyznaczy długość płotu według planu pana Antoniego. Normalnie zagadnienie to rozwiązuje się tzw. algorytmem Grahama, ale tutaj plan zagospodarowania przestrzennego wymaga, żeby boki prostokąta były równoległe do osi OX oraz OY kartezjańskiego układu współrzędnych.

Dane o drzewach znajdują się w pliku tekstowym gaj.txt. Każda linia pliku zaczyna się od jednowyrazowej nazwy gatunku drzewa. Kolejne $2j \geq 0$ liczb w tej linii opisuje położenia (x, y) drzew tego gatunku w gaju.

Twój program powinien wypisywać w odpowiedzi jedną liczbę — minimalną długość płotu opisanego w zadaniu.

PRZYKŁAD PLIKU GAJ.TXT

```
wierzba 3 5 3 -3  
kasztanowiec 0 12 -5 10  
brzoza -30 30  
sosna  
jarzebina 29 12 30 30 -30 -30
```

Dla tego pliku prawidłową odpowiedzią jest 240.