

## EGZAMIN Z PROGRAMOWANIA KOMPUTERÓW – TERMIN I

IMIĘ I NAZWISKO: .....

GRUPA: .....

## ZADANIE NR 1 (7 PKT)

- a) (3 pkt) Posługując się arytmetyką na wskaźnikach (bez używania operatora indeksowania) napisać funkcję odwracającą kolejność liter w łańcuchu znakowym podanym jako parametr wywołania. Nie wolno wykorzystywać żadnych standardowych funkcji operujących na łańcuchach znakowych.

```
prototyp: void reverse( char * );
```

Przykład

```
char *s = „wakacje”;
reverse( s );
printf( s );                // wypisuje napis: „ejcakaw”
```

- b) (4 pkt) Wykorzystując funkcję z podpunktu a), napisać program który w pliku tekstowym odwróci i posortuje wszystkie wyrazy występujące w nim nieparzystą ilość razy, usuwając jednocześnie wystąpienia nadmiarowe. Nazwa pliku jest parametrem wywołania programu.

Przykład

Wywołanie programu:  
Zadanie1.exe dane.txt

Zawartość pliku dane.txt:

```
Kot
Mysz
Pies
Kot
Kot
Pies
Chomik
Mysz
```

Zawartość pliku dane.txt po wywołaniu programu:

```
kimohC
toK
```

**Uwaga !**

- Prace napisane niechlujnie, a przez to nieczytelne – nie będą oceniane!
- Komentarze w programie nie są obowiązkowe, jednak mogą one ułatwić interpretację zmiennych bądź fragmentów programu przez sprawdzającego zwiększając szansę na zrozumienie intencji autora rozwiązania, dlatego wskazane jest ich stosowanie!

# EGZAMIN Z PROGRAMOWANIA KOMPUTERÓW – TERMIN I

IMIĘ I NAZWISKO: .....

GRUPA: .....

## ZADANIE NR 2 (13 PKT)

W oparciu o deklaracje klas przedstawionych w dalszej części należy wykonać następujące podzadania:

1. **(1pkt)** Zapisz deklarację klas `Student` i `Wykładowca` pochodnych po klasie `Osoba`. `Student` rozszerza klasę `Osoba` o informacje o ocenie, a `Wykładowca` – o numerze pokoju w którym udziela konsultacji.
2. **(2pkt)** Zdefiniuj klasę `Aula`, której obiekty będą służyć do przechowywania na dynamicznej liście jednokierunkowej wskaźników do znajdujących się w sali numerowanych krzesełek, na których mogą siedzieć studenci lub wykładowcy. Wszelkie atrybuty tej klasy mają być prywatne. Uwaga: w momencie utworzenia obiektu, na auli powinna znajdować się wskazana, początkowa liczba krzesełek.
3. **(2pkt)** Zdefiniuj operator `<<` wypisujący do strumienia wyjściowego wszystkie informacje o studentach i wykładowcach znajdujących się na sali (czyli siedzących na krzesełkach).
4. **(3pkt)** Dla klasy `Aula` zdefiniuj operator indeksowania pozwalający na rozmieszczanie studentów i wykładowców na wybranych (numerowanych) krzesełkach w sali. Ewentualne odwołania poza zakres należy zgłaszać wyjątkiem klasy `Exception`.
5. **(3pkt)** Zdefiniuj funkcję operatorową przeciążającą operator `++` (przyrostkowy) dodającej do sali (obiektu klasy `Aula`) studenta. W razie braku miejsc na sali, student przynosi ze sobą krzesło.
6. **(2pkt)** Zadeklaruj i zdefiniuj destruktor dla klasy `Aula`. W razie konieczności zdefiniuj też inne metody dla innych klas.

### Uwaga!

Nie wolno wykorzystywać zaprzyjaźniania oraz niestandardowych klas języka C++.

Oto definicje wykorzystywanych klas:

```
class Osoba
{
    char *nazwisko;
    char *imie;
    int wiek;
};

class Exception
{
    ...
};
```

### Uwaga !

- Prace napisane niechlujnie, a przez to nieczytelne – nie będą oceniane!
- Komentarze w programie nie są obowiązkowe, jednak mogą one ułatwić interpretację zmiennych bądź fragmentów programu przez sprawdzającego zwiększając szansę na zrozumienie intencji autora rozwiązania, dlatego wskazane jest ich stosowanie!