

## Programowanie i struktury danych – egzamin

**Organizacja plików:** Wszystkie pliki oddawane do sprawdzenia należy zapisać we wspólnym folderze o nazwie będącej numerem indeksu, umieszczonym na pulpicie. **Oddajemy tylko źródła programów (pliki o rozszerzeniach .cpp i .h)!!!**

Napisać pomocnicze biblioteki oraz program główny dla następującego zadania:

Znaczenie numeru NIP:

NIP (numer identyfikacji podatkowej) jest zawsze 10-cyfrowy, obecnie zapisuje się go jako ciąg cyfr, bez łączników (myślników). Prefiks - trzy pierwsze cyfry każdego NIP oznaczają kod urzędu skarbowego, który nadał dany numer. Dziesiąta cyfra NIP jest cyfrą kontrolną obliczaną według poniższego algorytmu:

1. Pomnożyć każdą z pierwszych dziewięciu cyfr odpowiednio przez wagi: 6, 5, 7, 2, 3, 4, 5, 6, 7.
2. Zsumować wyniki mnożenia.
3. Obliczyć resztę z dzielenia przez 11 (operacja modulo 11).

NIP jest tak generowany, aby nigdy w wyniku tego dzielenia nie wyszła liczba 10.

Dane są kody następujących urzędów skarbowych:

771 Urząd Skarbowy w Piotrkowie Trybunalskim

772 Urząd Skarbowy w Radomsku

773 Urząd Skarbowy w Tomaszowie Mazowieckim

Dany jest plik tekstowy nipy.txt. W kolejnych liniach wypisywane są:

NIP (10 kolejnych cyfr), odstęp, nazwa podatnika (jako łańcuch o nieznanej długości)

Należy utworzyć pliki o nazwach piotrkow.txt, radomsko.txt oraz tomaszow.txt i wypisać w nich nazwy podatników, którzy składają zeznania w odpowiednich urzędach skarbowych. Jeśli w przeczytanej linii występują niewłaściwe dane, należy powtórzyć taką linię na standardowym wyjściu i wypisać odpowiedni komunikat (np. „inny urząd skarbowy”, „błędny numer nip” itp.). Linie w plikach wyjściowych powinny występować w takim samym porządku, jak w pliku wejściowym.

Niedozwolone jest korzystanie z kontenerów i algorytmów z biblioteki standardowej C++. Można stworzyć bibliotekę z abstrakcyjnym typem danych, który pomoże w rozwiązaniu zadania. Można w bibliotece (niekoniecznie tej samej) umieścić potrzebne struktury/funkcje działające na liczbach i łańcuchach znaków. **Zapisu do plików wynikowych nie wolno rozpocząć przed odczytaniem całości pliku wejściowego.**

Poprawne rozwiązanie będzie ocenione jako przynajmniej **dostateczne**. Rozwiązanie **dobrze** powinno zawierać obsługę wejścia odporną na błędy użytkownika, a na **bardzo dobrą** również poprawne komentarze, w tym w bibliotekach pomocniczych: cel funkcji, warunki wstępne, warunki końcowe, sytuacje wyjątkowe i ewentualnie zwracany wynik. Na ocenę **dostateczną** jako poprawny traktowany jest numer NIP będący ciągiem 10 cyfr. Na ocenę **dobrą** należy zweryfikować poprawność numeru NIP poprzez weryfikację jego cyfry kontrolnej, natomiast na ocenę **bardzo dobrą** należy akceptować również numery NIP złożone z 10 cyfr oddzielonych dowolną liczbą myślników. Programy na ocenę przynajmniej **dobrą** powinny obsługiwać nieograniczoną liczbę numerów NIP zapisanych w pliku wejściowym.