Konkurs dla uczniów szkół ponadgimnazjalnych województwa pomorskiego w roku szkolnym 2018/2019

Etap II – powiatowy

Przedmiot: INFORMATYKA

Instrukcja dla rozwiązującego

1. Treść zadań jest podana poniżej w arkuszu. Dodatkowo do dwóch pierwszych zadań otrzymujesz pliki niezbędne do rozwiązania. Jest ich łącznie 5: trzy do **Zadania 1** oraz dwa do **Zadania 2**. W treści zadań są podane nazwy tych plików, których masz używać przy rozwiązywaniu każdego z nich.
2. Do niektórych zadań, co zostało wyraźnie wskazane w treści, możesz samodzielnie tworzyć pliki o charakterze testowym. Nie możesz jednak używać własnych plików, zamiast tych pięciu przeznaczonych do dwóch pierwszych zadań.
3. Zwróć uwagę, aby pliki zawierające rozwiązania poszczególnych zadań miały zawartość i nazwy takie, jakie określono w treściach zadań. Nie przesyłaj żadnych innych plików niż te określone w treści zadań.
4. Pliki z rozwiązaniami umieszczasz poprzez formularz zamieszczony na platformie Zdolni z Pomorza lub stronie internetowej Organizatora Pomorskiej Ligi Zadaniowej. Szczegóły techniczne związane z tą czynnością będą podane osobno. Tam również otrzymasz instrukcję jakie informacje dodatkowe, oprócz plików z rozwiązaniami powinny zostać przekazane.
5. Przy rozwiązywaniu zadań powinno się wykorzystywać te środowiska i narzędzia programistyczne, którymi posługujesz się w szkole lub w domu. W szczególności dopuszcza się następujące środowiska:
   a) systemy operacyjne – zarówno z grupy Windows, jak i dystrybucje systemu Linux;
   b) pakiety oprogramowania biurowego- Microsoft Office, ale również wersje otwarte np. Libre Office;
   c) języki programowania – C/C++,C#, Free Pascal, Java (kompilatory adekwatne do używanych środowisk systemu operacyjnego: DEV, GCC,G++ itd.);
   d) wizualne środowiska programowania – Scratch, Logomocja lub inne mutacje LOGO;
   e) nie określa się szczegółowo numerów wersji używanego oprogramowania, aby uczeń mógł je elastycznie dostosować do używanych w szkole.
6. Programy nie powinny zajmować się testowaniem poprawności danych. Zakłada się, że dane są poprawne.
7. Zadania rozwiązujemy samodzielnie.

Życzymy powodzenia!

Zadanie 1. (0 – 13pkt)

Załącznikami do tego zadania są trzy pliki: ***Zadanie1Załacznik1, Zadanie1Załącznik2, Zadanie1Załącznik3.*** Wszystkie zawierają ciągi liczb. W pliku ***Zadanie1Załacznik1*** znajdziesz liczby zapisane w systemie dwójkowym, zaś w pozostałych dwóch liczby zapisane w systemie dziesiętnym.

Część I

Rozwiąż następujące problemy wykorzystując dane z pliku ***Zadanie1Załacznik1.***

a) Podaj wartości dziesiętne wszystkich liczb zapisanych w tym pliku.

b) Przyjmij na chwilę, że wszystkie liczby zapisane są w kodzie U2 (o zmiennej w tym przypadku długości), w którym koduje się w informatyce liczby całkowite. Podaj jakie w tej sytuacji byłyby wartości dziesiętne wszystkich liczb zapisanych w tym pliku.

c) Podaj ile zer, a ile jedynek znajduje się łącznie w pliku ***Zadanie1Załacznik1.***

d) Przedstaw liczby występujące w pliku, w kolejności rosnącej ze względu na liczbę występujących w każdej z tych liczb jedynek.

e) Sporządź zestawienie zawierające informację ile mamy w pliku liczb o różnej ilości jedynek tzn. ile jest liczb z jedną jedynką, ile z dwiema , ile z trzema itd. (Uwaga! w pliku ***Zadanie1Załacznik1*** nie ma ani jednej liczby nie zawierającej choćby jednej jedynki). Zestawienie zilustruj wykresem.

f) Ile mamy w pliku liczb palindromicznych (czyli takich, które czytane normalnie i wspak mają tę samą wartość)?

Do oceny oddajesz plik zawierający komputerową realizację obliczeń, na podstawie której uzyskałeś odpowiedzi na wszystkie postawione w **Części I** problemy. Nazwa tego pliku to ***Zadanie1-I***. Odpowiedzi liczbowe na pytania postawione w punktach od **a** do **f** umieść w pliku tekstowym o nazwie ***Zadanie1-Iwyniki***. Dokładnie zapisz, które wyniki są odpowiedziami do pytań postawionych w punktach **Części I**.

Część II

Kolejne problemy wiążą się z liczbami zapisanymi w plikach ***Zadanie1Zalacznik2*** oraz ***Zadanie1Zalacznik3.***

a) Ile razy w pliku ***Zadanie1Zalacznik3*** występują poszczególne cyfry?

b) Podaj liczbę z największą ilością wystąpień cyfry 4 w pliku ***Zadanie1Zalacznik3,*** a jeżeli takich liczb jest więcej to zapisz dowolną z nich.

c) Zapisz zbiór wszystkich liczb występujących jednocześnie w plikach ***Zadanie1Zalacznik2*** oraz ***Zadanie1Zalacznik3.***

d) Sporządź wykres kołowy przedstawiający liczebności trzech grup liczb: zaczynających się od 1, zaczynających się od 9, pozostałych liczb w pliku ***Zadanie1Zalacznik2***.

Do oceny oddajesz plik zawierający komputerową realizację obliczeń, na podstawie której uzyskałeś odpowiedzi na wszystkie postawione w **Części II** problemy. Nazwa tego pliku to ***Zadanie1-II***. Odpowiedzi liczbowe na pytania postawione w punktach od **a** do **d** zapisz w pliku tekstowym o nazwie ***Zadanie1-IIwyniki***. Dokładnie zapisz, które wyniki są odpowiedziami do pytań postawionych w punktach **Części II**.

Uwaga ! Odpowiedzi liczbowe umieszczone w plikach tekstowych ***Zadanie1-Iwyniki*** oraz ***Zadanie1-IIwyniki*** nie będą mogły być uznane nawet jeśli będą poprawne, o ile nie znajdą potwierdzenia i odzwierciedlenia w zawartości plików ***Zadanie1-I*** oraz ***Zadanie1-II*** zawierających komputerową realizację obliczeń.

Zadanie 2. (0 – 10pkt)

W dwóch załączonych plikach tekstowych wpisane są dane (**fikcyjne**) uczniów różnych szkół z całego kraju, uczestników Olimpiady Informatycznej oraz ich wyniki w I etapie tej Olimpiady.

W pliku ***Załącznik uczniowie-zadanie2.txt*** w każdym wierszu znajduje się:

Identyfikator ucznia

Imię

Nazwisko

Nazwa szkoły

Miejscowość

Klasa (dokładniej rocznik klasy tzn. 1,2,3)

Plik ***Załącznik wyniki-zadanie2.txt*** zawiera następujące informacje:

Identyfikator ucznia

Zad1

Zad2

Zad3

Zad4

Zad5

Przy czym Zad1, Zad2, zad3, Zad4 oraz Zad5 zawierają wyniki uczniów uzyskane za rozwiązanie poszczególnych zadań (maksymalnie można było uzyskać 100 punktów za każde zadanie)

Na podstawie dostarczonych danych i przy pomocy dostępnych narzędzi informatycznych rozwiąż poniższe problemy.

1. Podaj średni **wynik** uzyskany przez uczniów za rozwiązanie poszczególnych zadań od 1 do 5, z dokładnością do dwóch miejsc po przecinku.
2. **Zbuduj zestawienie i zilustruj** je wykresem obrazujące ile osób pochodziło z jakich miast. W zestawieniu należy uwzględnić wyłącznie miasta posiadające wśród olimpijczyków przynajmniej 8 uczniów.
3. **Zbuduj zestawienie podające** imię, nazwisko, wyniki za poszczególne zadania, sumaryczny wynik oraz średni wynik (z dokładnością do dwóch miejsc po przecinku) dla wszystkich osób z miast na literę J.
4. **Skonstruuj narzędzie** pozwalające wyszukiwać ucznia wg jego identyfikatora. Dla znalezionej osoby należy wyświetlić jej imię, nazwisko, miejscowość, nazwę szkoły, klasę oraz sumaryczny wynik uzyskany w I etapie Olimpiady. Do pliku z wynikami (jego nazwa jest określona na końcu zadania) zapisz wynik zapytania dla ucznia o identyfikatorze 12.
5. **Podaj uporządkowaną alfabetycznie** wg nazwisk listę imion i nazwisk osób uczących się w liceach, w klasie 3. Innymi słowy w nazwie szkoły musi być zawarte słowo „Liceum", nie wchodzi w grę np. Zespół Szkół Ogólnokształcących.
6. **Podaj średni wynik** (z dokładnością do 2 miejsc do przecinku) jaki uzyskano w poszczególnych miastach za rozwiązanie zadania 5, ale w zestawieniu uwzględnij tylko pierwszą setkę przysyłających zadania (czyli uczniów o identyfikatorze do 100) oraz nie pokazuj miast, dla których ta średnia wynosi 0.
7. **Skonstruuj narzędzie**, które pozwoli symulacyjnie określać jacy uczniowie zakwalifikują się do II etapu Olimpiady Informatycznej zależnie od ustalonego progu. Próg powinien być parametrem zapytania (przez co da możliwość testowania różnych wariantów), a zestawienie winno obejmować imię, nazwisko, szkołę, miejscowość ucznia. Zestawienie powinno być podane w porządku alfabetycznym ze względu na nazwiska. Do pliku z wynikami (jego nazwa jest określona na końcu zadania) zapisz przykładowe zestawienie dla progu równego 360 punktów (w sumie za wszystkie zadania).
8. **Zbuduj zestawienie**, które zawierać będzie imię, nazwisko, miejscowość każdego ucznia oraz dodatkową kolumnę dotyczącą nazwy województwa, w którym leży miejscowość ucznia. Przy czym załóżmy, że nie znamy specjalnie podziału administracyjnego kraju i nazwa województwa znajdzie się tylko przy dwóch miastach tzn. Gdyni (pomorskie) oraz

Krakowie (małopolskie), przy pozostałych miastach w kolumnie województwo powinien się pojawić zwrot „Niestety nie wiem”.

Odpowiedzi na te pytania (poza wykresem) umieść w pliku tekstowym ***Zadanie2-odpowiedzi.txt***

Do oceny oddajesz plik(i) zawierający komputerową realizację obliczeń, na podstawie której uzyskasz odpowiedzi na wszystkie postawione w zadaniu problemy. Nazwa tego pliku to *Zadanie2* lub jeśli jest ich więcej to kolejne pliki mają nazwy *Zadanie 2a, Zadanie2b itd.* Ponadto załączasz wymieniony już plik *Zadanie2-odpowiedzi.txt*.

Uwaga ! Odpowiedzi umieszczone w pliku *Zadanie2-odpowiedzi.txt* nie będą mogły być uznane nawet jeśli będą poprawne, o ile nie znajdą potwierdzenia i odzwierciedlenia w zawartości pliku(ów) z komputerową realizacją obliczeń.

Zadanie 3. (0 – 7pkt)

Przygotuj **formularz**, który będzie umożliwiał wysłanie krótkiej informacji tekstowej za pośrednictwem sieci internetowej oraz skrypt w języku PHP, który po stronie serwera zajmie się przetworzeniem danych wysłanych z tego formularza.

Użytkownik w formularzu znajdzie pole, w którym powinien podać swój **identyfikator** oraz pole na dowolną **wiadomość**, przy czym pole wiadomości nie powinno być dłuższe niż 50 znaków. Te dane zatwierdza następnie przyciskiem **Wyślij**.

Skrypt PHP, który przejmie dane z tego formularza powinien zrealizować następujące zadania:

a) w przypadku, gdy nie podano **identyfikatora** nie podejmuje się żadnej akcji informując użytkownika o tym;

b) w przypadku kiedy użytkownik podał tylko **identyfikator** bez wiadomości, to skrypt powinien generować dowolny **komunikat** powitalny oraz informację, że nie podano wiadomości, a następnie w pliku tekstowym ***wiadomosci.txt,*** w którym zapisuje się przesyłane wiadomości, sprawdzić, czy ten użytkownik już zapisywał jakąkolwiek wiadomość (użytkownik może przesyłać wiadomości wiele razy). Jeżeli zapisywał to na ekranie powinna się wyświetlić ostatnio zapisana przez osobę o tym identyfikatorze wiadomość, a jeżeli nie, to powinien pojawić się tekst „**Witamy po raz pierwszy**”

c) w przypadku, gdy podano zarówno **identyfikator**, jak i **wiadomość** to skrypt zaszyfrowuje wiadomość zwykłym losowym szyfrem przestawieniowym (czyli po prostu losowo przestawia w niej znaki), a następnie w pliku tekstowym ***wiadomosci.txt*** dopisuje na końcu nową linię, w której będzie identyfikator użytkownika i po odstępie zaszyfrowana, przesłana przez niego wiadomość. Dodatkowo, gdy użytkownik podaje wiadomość nie po raz pierwszy(i tylko wtedy, gdy podaje wiadomość) to na ekranie powinna się pojawić informacja o liczbie dotąd zapisanych przez użytkownika o tym identyfikatorze wiadomości oraz suma liczby znaków zawartych we wszystkich wiadomościach tego użytkownika.

Uwagi

1. Plik tekstowy ***wiadomosci.txt*** do działań testowych można sobie założyć samodzielnie. Natomiast nie trzeba go przesyłać. Sprawdzający rozwiązanie utworzy sobie własną wersję tego pliku w czasie kontroli przesłanego rozwiązania.
2. Rozwiązanie można testować w dowolnym środowisku umożliwiającym realizację skryptów PHP np. na lokalnym serwerze. Formularz i skrypt powinny być jedynie tak napisane, aby powiązania między nimi funkcjonowały właściwie bez względu na to środowisko.
3. Proszę dodatkowo w pliku ***uwagi.txt*** zamieścić ew. uwagi techniczne, które mogą być ważne dla sprawdzającego działanie formularza i skryptu (np. wersja PHP).

Do oceny oddajesz plik *Zadanie 3-formularz* zawierający formularz, o którym mowa w zadaniu oraz *Zadanie3-skrypt* zawierający skrypt PHP realizujący opisane w zadaniu

akcje. Dodatkowo, jeśli była taka konieczność uzupełniamy te pliki o plik *uwagi.txt*, o którym mowa wyżej.

Zadanie 4. (0 – 10pkt)

Podsłowem danego *słowa x* o długości *n znaków* nazywamy każde słowo, które rozpoczyna się od dowolnego znaku danego *słowa x*, składa się z kolejno następujących po nim znaków *słowa x* i kończy się na dowolnym znaku *słowa x*, ale ma długość krótszą niż *słowo x*. Przykład: dla słowa ***abrakadabra*** jego podsłowami są ***bra, kada, brak, brakadbra***, ale nie ***dar*** *(*bo nie jest to ciąg kolejno występujących po sobie znaków), ani też samo słowo ***abrakadabra*** (bo podsłowo musi być krótsze od słowa).

Prefiksem danego słowa o *długości n* nazywamy każde podsłowo tego słowa rozpoczynające się od jego pierwszego znaku. Przykład: dla słowa ***kark*** jego prefiksy to ***k, ka, kar***.

Sufiksem danego słowa o *długości n*, nazywamy każde podsłowo tego słowa kończące się na jego ostatnim *n*-tym znaku. Przykład: dla słowa ***kark*** jego sufiksy to ***ark, rk, k***.

Prefikso-sufiksem danego słowa o *długości n*, nazywamy każde podsłowo tego słowa będące jednocześnie jego prefiksem i sufiksem. Przykład: dla słowa ***kark*** jego jedynym prefikso-sufiksem jest słowo ***k***, ale np. słowo ***aaabaaa*** ma trzy prefikso-sufiksy to znaczy: ***a,aa, aaa***.

Powyższe definicje są znane w informatyce, ale na potrzeby tego zadania zdefiniujemy jeszcze jedno pojęcie:

Słowami zbliżonymi do prefikso-sufiksu *stopnia k* (dalej przyjmiemy skrót *SZDPS stopnia k*), gdzie *k* jest liczbą naturalną większą od zera nazywać będziemy dla danego słowa parę jego podsłów spełniającą następujące wymagania:

a) jedno z podsłów jest prefiksem, a drugie sufiksem słowa, ale nie są prefikso-sufiksem tego słowa (czyli różnią się od siebie)

b) podsłowa różnią się długością, co najwyżej o 1

c) w obu podsłowach jest dokładnie *k* takich samych znaków, ale nie ma na pewno *k+1* takich samych znaków

Przykład dla słowa ***aacbaaa*** para ***a*** oraz ***aa*** to ***SZDPS stopnia 1*** , przykładem ***SZDPS stopnia 2*** (nie jedynym) dla tego słowa może być para ***aac*** i ***aaa***, a przykładem ***SZDPS stopnia 5*** dlatego słowa (także nie jedynym) może być para ***aacbaa*** i ***cbaaa***.

Napisz program, który dla dowolnego słowa złożonego wyłącznie z małych liter (bez polskich znaków) zapisanego w pliku ***slowo.txt*** wypisze do pliku tekstowego ***wynik.txt*** następujące rezultaty w podanej kolejności:

a) wszystkie prefikso-sufiksy danego słowa od najkrótszego do najdłuższego

b) wszystkie ***SZDPS stopnia 1***

c) wszystkie ***SZDPS stopnia 2***

d) wszystkie ***SZDPS stopnia 3*** itd. aż do ***SZDPS*** maksymalnego możliwego teoretycznie dla danego słowa stopnia

Uwagi:

1. Jeżeli pewnej grupy spośród wymienionych w punktach od **a** do **d** nie ma: wypisujemy do pliku słowo brak zamiast jej elementów.
2. W grupach zawierających ***SZDPS*** kolejnych stopni kolejność wypisywania par słów jest dowolna.
3. Nie należy wypisywać więcej niż raz par złożonych z tych samych słów (np. dla podanego wyżej słowa ***aacbaaa*** para ***a*** oraz ***aa*** jako przykład ***SDZPS stopnia 1*** mogłaby być teoretycznie wyświetlone dwukrotnie, bo ***a*** podobnie jak ***aa*** raz jest prefiksem, a raz sufiksem słowa, ale wypisujemy tę parę tylko raz)

Dane wejściowe

Plik tekstowy ***słowo.txt*** zawierający tylko jeden wiersz, a w nim słowo zapisane tylko przy pomocy małych liter (bez polskich znaków) o długości większej od zera i nieprzekraczającej 30 znaków.

Dane wyjściowe

Zapisane w pliku ***wynik.txt*** kolejno:

a) wszystkie prefikso-sufiksy danego w pliku ***słowo.txt*** słowa od najkrótszego do najdłuższego

b) wszystkie ***SZDPS stopnia 1*** (w dowolnej kolejności)

c) wszystkie ***SZDPS stopnia 2*** (w dowolnej kolejności), aż do ***SZDPS*** maksymalnego (możliwego teoretycznie!) dla danego słowa stopnia

Szczegółowy wzorzec zapisu tych wyników do pliku zawierają przykłady.

Przykład 1

Jeżeli w pliku ***słowo txt*** zapisano słowo aacbaaa to zawartość pliku ***wynik.txt*** powinna być następująca:

Prefikso-sufiksy:

a

aa

SZDPS stopnia 1:

a aa

SZDPS stopnia 2:

aa aaa

aac **aa**

aac **aa**a

aac b**aa**a

aacb **aa**a

SZDPS stopnia 3:

aacb baaa

SZDPS stopnia 4:

aacb cbaaa

aac**ba baaa**

SZDPS stopnia 5:

aacba cbaaa

aacba acbaaa

aaacba **cbaaa**

SZDPS stopnia 6:

aaacba acbaaa

Przykład 2

Jeżeli w pliku ***słowo txt*** zapisano słowo **kark** to zawartość pliku ***wynik.txt*** powinna być następująca:

Prefikso-sufiksy:

k

SZDPS stopnia 1:

k r**k**

ka **k**

ka r**k**

SZDPS stopnia 2:

ka ar**k**

ka**r rk**

SZDPS stopnia 3:

kar ark

Przykład 3

Jeżeli w pliku ***słowo txt*** zapisano słowo pal to zawartość pliku ***wynik.txt*** powinna być następująca:

Prefikso-sufiksy:

brak

SZDPS stopnia 1:

p**a a**l

SZDPS stopnia 2:

brak

Zauważmy, że w przykładzie 3 maksymalny możliwy teoretycznie ***SDZPS*** jest stopnia 2 i choć akurat nie ma w tym przypadku takich par słów to wyświetlenie zgodnie z założeniami obejmuje tę grupę również tyle, że ze słowem „Brak”.

Dla ułatwienia zrozumienia przykładów w każdym z nich drukiem pogrubionym zaznaczono wspólne znaki dla obu słów tworzących ***SZDPS*** danego stopnia. Naturalnie w pliku ***wynik.txt*** powstającym w efekcie wykonania Twojego kodu tego typu pogrubienia nie są wymagane.

Do oceny oddajesz plik zawierający kod źródłowy programu o nazwie *Zadanie4.*

Zadanie 5. **(0 – 10pkt)**

Liczby postaci $\mathbf{2^n - 1}$, gdzie n jest dowolną liczba naturalną nazywamy liczbami Mersenne'a na część francuskiego matematyka, który sądził dodatkowo, że jeżeli n jest liczba pierwszą to w ten sposób otrzymamy zawsze również liczbę pierwszą. Mersenne jak się później okazało mylił się, ale w naszym zadaniu zajmiemy się liczbami, które wskazał jako potencjalnie pierwsze.

Napisz program, który w dowolnym ciągu liczb naturalnych, większych od zera zapisanych w pliku tekstowym ***liczby.txt*** znajdzie najdłuższy możliwy podciąg niemalejący złożony z liczb postaci $\mathbf{2^p - 1}$ (gdzie p jest liczbą pierwszą), położonych niekoniecznie obok siebie w tym ciągu. Odnaleziony ciąg należy zapisać (w kolejności w jakiej liczby go tworzące występują w pliku tekstowym ***liczby.txt)*** w pliku ***ciag.txt*** wraz z jego długością**.** Jeżeli istnieje dwa lub więcej podciągów niemalejących złożonych z liczb postaci $\mathbf{2^p - 1}$ (gdzie p jest liczbą pierwszą) wystarczy wypisać jeden, dowolnie wybrany spośród nich. Jeżeli w pliku ***liczby.txt*** w ogóle nie ma liczb postaci $\mathbf{2^p - 1}$ (gdzie p jest liczbą pierwszą), to w pliku ***ciag.txt*** należy jedynie umieścić słowo **brak.**

Dane wejściowe:

Plik tekstowy ***liczby.txt*** zawierający w pierwszym wierszu liczbę naturalną $\boldsymbol{n}$ oznaczającą liczbę elementów ciągu ($1 \leq n \leq 100$), a w kolejnych n wierszach liczby naturalne, większe od zera będące elementami tego ciągu (po jednej liczbie w każdym wierszu). Przyjmujemy, że żadna z liczb ciągu nie jest większa niż $2^{63} - 1$.

Dane wyjściowe:

Plik tekstowy ***ciąg,txt*** zawierający maksymalnie $n + 1$ wierszy. W jego pierwszym wierszu znajduje się liczba $\boldsymbol{k}$ określająca długość najdłuższego ciągu niemalejącego złożonego z liczb postaci $\mathbf{2^p - 1}$ (gdzie p jest liczbą pierwszą) znajdujących się w ciągu umieszczonym w pliku ***liczby.txt*** niekoniecznie na sąsiednich miejscach, zaś w kolejnych k wierszach pliku ***ciąg,txt*** podane są elementy tego ciągu. W przypadku, gdy jest więcej ciągów o długości k wybieramy dowolny z nich. Z kolei w przypadku, gdy w pliku ***liczby.txt*** w ogóle nie ma liczb postaci $\mathbf{2^p - 1}$ (gdzie p jest liczbą pierwszą), to w jedynym wierszu pliku ***ciąg,txt*** zapisane jest słowo „***brak"***.

Przykład 1

Dla danych wejściowych umieszczonych w pliku ***liczby.txt :***

8

7

3

2

3

124566

2047

10000

8191

plik ***ciag.txt*** powinien mieć postać:

4

3

3

2047

8191

bo $3 = 2^2 - 1$, $2047 = 2^{11} - 1$, a $8191 = 2^{13} - 1$

Przykład 2

Dla danych wejściowych umieszczonych w pliku ***liczby.txt :***

4

31

7

3

2000

plik ***ciag.txt*** powinien mieć postać:

1

31

chociaż poprawną odpowiedzią będą również ciągi o długości 1 złożone tylko z liczby 7 lub tylko z liczby 3 (31, ale także 7 i 3 są liczbami postaci $\mathbf{2^p - 1}$, gdzie p jest liczbą pierwszą, ale najdłuższy podciąg niemalejący jaki można utworzyć z liczb o tej własności dla tych danych wejściowych ma długość 1)

Przykład 3

Dla danych wejściowych umieszczonych w pliku ***liczby.txt :***

3

30

7834

2

plik ***ciag.txt*** powinien mieć postać:

brak

Do oceny oddajesz plik zawierający kod źródłowy programu o nazwie *Zadanie5*