POMORSKA LIGA ZADANIOWA *ZDOLNI Z POMORZA*

Konkurs dla uczniów dla uczniów klas VII i VIII szkoły podstawowej województwa pomorskiego w roku szkolnym 2019/2020

Etap II – powiatowy

Przedmiot: Informatyka

Przed przystąpieniem do rozwiązywania zadań zapoznaj się z instrukcją.

INSTRUKCJA

1. Oprócz arkusza z treścią zadań otrzymujesz załączniki niezbędne do rozwiązania niektórych zadań. Ich nazwy są określone w treści zadań. Łącznie są 3 (dwa do zadania pierwszego oraz jeden do zadania czwartego). Przed przystąpieniem do rozwiązywania sprawdź, czy na pewno pobrała(e)ś te wszystkie pliki. Nie wolno używać własnych plików zamiast tych załączników wyraźnie wymienionych w treści zadania.
2. Zwróć uwagę, aby pliki zawierające rozwiązania oraz pliki z danymi testowymi (takie pliki będziesz tworzyć samodzielnie rozwiązując zadania 3,4 oraz 5) miały zawartość i nazwy takie jakie określono w treściach zadań.
3. Nie przesyłaj do oceny innych plików niż te określone w treści zadań.
4. Pliki z rozwiązaniami przesyłasz organizatorom Pomorskiej Ligi Zadaniowej zgodnie z odrębną instrukcją.
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. Libra Office

c) języki programowania – C/C++,C#, Free Pascal, Java, Python (kompilatory adekwatne do używanych środowisk systemu operacyjnego np. DEV, Code Block, Eclipse, GCC,G++ itp.)

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, ale w przypadku języków programowania prosimy o dokładne podanie (np. w odrębnym pliku tekstowym) jaka wersja kompilatora (względnie jakie środowisko programistyczne) było wykorzystywane, aby adekwatnego użyć przy ocenie pracy z zastrzeżeniem punktu 6a.

6. W przypadku rozwiązań związanych z używaniem języków programowania:
   a) powinno się używać kompilatorów bez ograniczonej dostępności (np. związanej z ich komercyjnym charakterem),
   b) dopuszcza się używanie wyłącznie standardowej biblioteki (bibliotek) języka, nie jest dozwolone dołączanie zewnętrznych bibliotek np. crt, graph (poza sytuacjami wynikłymi z treści zadania np. próba realizacji rysunku wynikającego z treści zadania),
   c) nie jest dopuszczalne otwieranie przez program innych programów, plików (poza tymi z danymi wejściowymi i wyjściowymi), ani tworzenie nowych plików (np. tymczasowych) oraz tworzenie innych procesów lub wątków,
   d) błąd kompilacji przy sprawdzaniu jest traktowany jako błąd składni i sprawdzający nie ma obowiązku dalszej analizy takiego rozwiązania choć może uwzględnić poprawny zarys samego algorytmu przyznając znacząco mniejszą liczbę punktów. Podobna uwaga dotyczy pojawienia się nietypowych błędów wykonania w trakcie uruchamiania programów (np. naruszenie zasad ochrony pamięci),
   e) rozwiązania nie powinny wykorzystywać plików nagłówkowych typowych dla środowisk DOS/Windows np. conio.h lub windows.h (dotyczy języka C++),
   f) rozwiązania nie powinny naruszać bezpieczeństwa systemowego w środowisku, w którym są sprawdzane

7. Programy nie powinny zajmować się testowaniem poprawności danych. zakłada się, że ma ona miejsce.
8. Proszę zwrócić uwagę na samodzielność rozwiązań.

Życzymy powodzenia!

Zadanie 1

W tym zadaniu zajmiesz się analizą danych związanych z funkcjonowaniem pewnej fikcyjnej szkoły.

I. Opisywana szkoła jest znana z tego, że jej uczniowie uczestniczą w licznych konkursach przedmiotowych, artystycznych i sportowych. Na koniec roku szkolnego w szkole sporządzany jest raport dotyczący udziału uczniów w tych konkursach. W załączonym pliku ***Zalacznik-Zadanie1-konkursy.txt*** znajdują się dane, które pomogą w wykonaniu tego raportu. W kolejnych wierszach znajdziemy oddzielone pojedynczym znakiem odstępu następujące dane:

nazwa konkursu, , liczba uczniów klas I-IV, która wzięła w nim udział, liczba uczniów klas V-VI, która wzięła w nim udział, liczba uczniów VII-VIII, która wzięła w nim udział, liczba dziewcząt biorących udział w konkursie, liczba półfinalistów konkursu reprezentujących szkołę (o ile taki etap był w konkursie przewidziany, jeżeli nie to w tym miejscu w danych mamy literę n), liczba finalistów konkursu (zakładamy, że zestawienie zawiera konkursy, które składały się z dwóch etapów rywalizacji, a więc w tej kolumnie nie może być litery n), liczba laureatów konkursu.

Twoje zadanie polega na tym, aby na podstawie tych danych pomóc w wykonaniu raportu podsumowującego, o którym mowa wyżej. W szczególności udziel odpowiedzi na następujące pytanie i rozwiąż następujące problemy:

a) Jaki procent konkursów był adresowany wyłącznie do najmłodszych uczniów z klas I-IV tzn. wzięli w nim udział tylko uczniowie tych klas (wynik podaj z dokładnością do jednego miejsca po przecinku) ?

b) Jaki procent uczniów wziął udział w poszczególnych konkursach (czyli wynik podajemy dla każdego z konkursów z osobna uwzględniając, że w szkole uczy się 550 uczniów) ? Wyniki podajemy z dokładnością do jednego miejsca po przecinku.

c) Ile równy jest dla każdego konkursu jego współczynnik efektywności, przez który rozumiemy stosunek liczby finalistów i laureatów do wszystkich startujących w konkursie (liczony z dokładnością do dwóch miejsc po przecinku). Następnie zbuduj odrębne zestawienie, w którym znajdują się dwie kolumny: w pierwszym jest nazwa konkursu, a w drugim jego współczynnik efektywności. Konkursy w tym zestawieniu winny być uszeregowane od tego, który ma najwyższy do tego, który ma najniższy współczynnik efektywności. Uwaga ! Laureat jest już liczony jako finalistą !

d) W przypadku ilu konkursów nie organizowano etapu półfinałowego ?

e) Stwórz zestawienie złożone z dwóch kolumn: w pierwszej będzie nazwa konkursu, w drugiej liczba jego laureatów ze szkoły, przy czym w zestawieniu konkursy winny być umieszczone w kolejności od tego, który miał najwięcej laureatów, do tego, który miał ich najmniej i nie powinno w nim być konkursów, w którym laureatów w ogóle nie było?

Dodatkowo raport należy uzupełnić dwoma wykresami:

f) przedstawiającym dla każdego konkursu liczbę dziewcząt oraz chłopców uczestniczących w danym konkursie,

g) przedstawiającym dla każdego konkursu liczbę uczestników z klas I-IV, V-VI oraz VII-VIII.

II. W tej części pomożesz dyrektorowi opisywanej szkoły w oszacowaniu kosztów różnych przedsięwzięć dla uczniów planowanych przez dyrektora szkoły w kolejnym roku szkolnym. W załączonym pliku ***Zalacznik-Zadanie1-imprezy.txt*** znajdują się dane, które pomogą w wykonaniu tego zadania. W kolejnych wierszach znajdziemy oddzielone pojedynczym znakiem odstępu następujące dane:

nazwa przedsięwzięcia, szacowana liczba uczestników-minimum, szacowana liczba uczestników-maksimum, czy przedsięwzięcie wyjazdowe (jeśli jest wyjazdowe w danych mamy TAK, w przeciwnym razie NIE).

Dyrektor przyjęła(ął) specyficzny sposób liczenia kosztów przedsięwzięć. Punktem wyjścia jest koszt przedsięwzięcia przypadający na jednego uczestnika w ubiegłym roku. Nazwijmy go na razie x. Dodatkowo dla przedsięwzięć wyjazdowych przyjmuje się, że koszt przypadający na jednego uczestnika powiększa się jeszcze o 60 % kwoty x.

Dla wyliczeń odnoszących się do nowego roku szkolnego dyrektor przyjęła(ął), że kwota x będzie wyższa o 8 % niż w roku poprzednim.

Uwzględniając te informacje odpowiedz na dwa pytania:

a) Ile trzeba mieć środków na przedsięwzięcia w nowym roku szkolnym w dwóch wariantach; udziału minimalnej oraz maksymalnej przewidywanej liczby uczniów, jeśli ubiegłoroczny x=50 ?
b) Dla jakiej najmniejszej (z dokładnością do 1 zł) ubiegłorocznej kwoty x suma 60 000 zł nie wystarczy szkole na przedsięwzięcia przyszłoroczne nawet w wariancie uczestnictwa minimalnej, zakładanej liczby uczniów ?

Do oceny oddajesz pliki zawierający komputerową realizację konstrukcji oraz obliczeń, na podstawie których uzyskasz rozwiązanie zadania. Nazwa tych plików to *Zadanie1-I* (ten plik zawiera rozwiązania części I zadania) *oraz Zadanie1-II* (ten plik zawiera rozwiązania części II zadania). Dodatkowo wyniki części I zadania (tylko problemy od I.a do I.e) umieść w pliku *Zadanie1-konkursy.txt*, a części II zadania (problemy II.a oraz II.b) w pliku *Zadanie1-imprezy.txt.*

Uwaga ! Odpowiedzi liczbowe umieszczone w plikach tekstowych *Zadanie1-konkursy* oraz *Zadanie1-imprezy* nie będą mogły być uznane nawet jeśli będą poprawne, o ile nie znajdą potwierdzenia i odzwierciedlenia w zawartości pliku z komputerową realizacją obliczeń.

11 punktów

Zadanie 2

Przy pomocy dowolnie wybranego edytora grafiki stwórz projekt na jeden z poniższych trzech tematów (czyli wybierasz jeden):

a) okładka fikcyjnej książki Twojego autorstwa „Szkolne przygody”,
b) plakat zapraszający na koncert fikcyjnego zespołu muzycznego, którego jesteś liderem (nazwa zespołu do własnego ustalenia),
c) plakat reklamujący fikcyjną restaurację, gdzie jesteś szefem kuchni.

Nie narzuca się żadnych szczegółów dotyczących tego jakie elementy mają się znaleźć w tym projekcie. Chodzi o to, aby wystarczająco trafnie i bez wątpliwości oddawał on wybrany temat (czyli nie zawierał np. przysłowiowego jednego elementu i opierał się na zasadzie domyśl się jaki to temat). **Nie będą natomiast tolerowane żadne rozwiązania „gotowe” czyli zdjęcia, obrazy z sieci internetowej itp.**

Następnie wyobraź sobie, że Twój obraz podzielono na 6 prostokątów o zbliżonej wielkości (tzn. nie muszą być dokładnie identyczne, ale tez bez rażących i zauważalnych różnic w ich wielkości). Będą to niejako puzzle, z których można ułożyć Twój obraz.

Przygotuj w dowolnej formie (np. prezentacji) instrukcję pokazującą jak należy zaczynając od rozrzuconych puzzli ułożyć krok po kroku Twój projektowy obraz w całość.

Do oceny oddajesz plik o nazwie *Zadanie2-projekt* zawierający zrealizowany przy pomocy edytora graficznego projekt na jeden z wybranych tematów oraz plik *Zadanie2-instrukcja* zawierający instrukcję układania Twojego projektu z puzzli zrealizowaną w wybrany przez Ciebie sposób.

8 punktów

Zadanie 3

Koło młodych informatyków w jednej ze szkół ostatnio bardzo pasjonuje się kryptografią. Właśnie opracowali specjalny szyfr, przy pomocy, którego szyfrują wymyślone przez siebie konstrukcje złożone z widocznych(drukowalnych) znaków kodu ASCII.

Wyobraź sobie, że poniższe dwa wiersze:

Qwerty2Ala ma kota3pies48banany7cytryna8alarm10ja

Pokemon1cos9ja7ijeszczedwa3oraz4ateraz1ikoniec2

po odszyfrowaniu wygenerowałyby na ekranie następujący wynik:

&&&&
&&&&
&&&&
&&&&
&&&&
&&&&
&&&&

aaa
aaa
aaa
aaa

KONIEC

Skąd taki właśnie wynik? Wyjaśnijmy to po kolei. Każda poprawnie zaszyfrowany wiersz tekstu (a jak zobaczymy dalej mogą istnieć również niepoprawne) zawiera kod ASCII znaku, który ma być wypisany, informację o liczbie znaków o tym kodzie wypisanych w jednym wierszu, informację o liczbie wierszy oraz informację o odstępie między tym blokiem, a kolejnym blokiem tekstu, w którym będzie już wypisywany inny znak, a któremu odpowiada już kolejny zaszyfrowany wiersz. **Ważna uwaga – wszystkie cyfry, które napotkamy**

w tekście traktujemy jako pojedyncze znaki, nawet jeśli znajdują się obok siebie, a w zaszyfrowanym wierszu nie ma spacji. Poszukajmy zatem informacji, które dały powyższy wydruk:

a) widoczne inaczej tzw. drukowalne kody ASCII obejmują zakres od 32 do126, my dodatkowo odrzućmy z tej grupy jeszcze spację (kod 32) i zawęzimy w ten sposób ten zakres do 33-126. W wymyślonym szyfrze kod znaku, który ma być wypisany został rozbity na dwie cyfry pierwszą z nich jest zawsze druga cyfra spotkana w zaszyfrowanym wierszu tekstu licząc od jego początku, natomiast miejsce drugiej określa w specyficzny sposób pierwsza cyfra , którą można znaleźć w zaszyfrowanym wierszu (**dlatego pierwsza spotkana w tekście cyfra nie może być cyfrą 0**). Dokładniej pierwsza spotkana w wierszu tekstu cyfra mówi, którą cyfrę należy dołączyć do kodu ASCII poszukiwanego znaku jako drugą, ale licząc dopiero po pierwszej, wcześniej znalezionej w tekście cyfrze tego kodu (czyli jak kto woli odliczanie zaczynamy od trzeciej cyfry spotkanej w zaszyfrowanym wierszu tekstu).Np. w pierwszym z podanych wyżej wierszy zaszyfrowany znak ma kod 38, bo pierwszą jego cyfra to 3(druga cyfra spotkana w tym wierszu). Pierwszą spotkaną w wierszu cyfrą było z kolei 2, czyli drugiej cyfry kodu należy szukać jako drugiej po wspomnianej cyfrze 3, albo jak kto woli jako czwartej w ogóle i jest to właśnie w tym tekście 8. Kod 38 ma zaś znak &. Analogiczne rozumowanie prowadzi do wniosku, że w drugim wierszu kryptogramu (czyli szyfrowanego tekstu) zaszyfrowano znak a (pierwszy znak kodu jakim jest druga cyfra to 9, a drugiego szukamy zaraz po nim, bo na to wskazywała cyfra 1, która była pierwszą w tym wierszu tekstu- ponieważ jest to 7 razem mamy 97 czyli kod litery a. I jeszcze jedna ważna sprawa. Jeśli po odczytaniu kodu otrzymamy kody od 00 do 26 to sami dodajemy do kodu cyfrę 1 z przodu, gdyż wówczas wiadomo, że chodzi o kody trzycyfrowe. Zauważmy wreszcie, że jeśli otrzymamy kody od 27 do 32 to taki wiersz jest niepoprawny, bo nie damy rady takiego znaku wydrukować. Będzie to zresztą jedyny rodzaj niepoprawności, z którym będziesz musiał(a) się zmierzyć rozwiązując to zadanie,

b) liczbę znaków, które należy wydrukować w jednym wierszu podpowie pierwsza licząc od początku wiersza tekstu, niewykorzystana przy rozszyfrowywaniu kodu ASCII znaku cyfra tego wiersza (dla pierwszego przykładowego wiersza jest nią 4 stąd po 4 znaki & w każdym wierszu, a dla drugiego 3 stąd po 3 litery a w wierszu).Ta cyfra nie może być zerem,

c) liczbę wierszy , które należy wydrukować podpowie druga licząc od początku wiersza tekstu niewykorzystana przy rozszyfrowywaniu kodu ASCII znaku cyfra wiersza tekstu (dla pierwszego przykładowego wiersza jest nią 7 stąd 7 wierszy znaków &, a dla drugiego 4 stąd 4 wierszy znaków a). Ta cyfra też nie może być zerem,

d) aby ustalić odstęp (liczony w wierszach) do kolejnego bloku tekstu należy utworzyć liczbę dwucyfrową z dwóch ostatnich cyfr wiersza tekstu (czyli dokładniej z przedostatniej i ostatniej) bez względu na to, czy te cyfry były wcześniej wykorzystywane do rozszyfrowania kodu ASCII , liczy znaków w wierszu lub liczby wierszy czy też nie. Następnie obliczamy największy wspólny dzielnik tej liczby dwucyfrowej oraz ustalonej wcześniej liczby wierszy i ten wynik daje nam odstęp do kolejnego bloku danych. Dla pierwszego wiersza przykładowo zaszyfrowanego wiersza tekstu mamy NWD(10,7)=1 i stąd linijka odstępu oddziela blok znaków & od bloku znaków a. Dla drugiego wiersza przykładowo zaszyfrowanego tekstu mamy NWD(12,4)=4 i stąd 4 linijki odstępu oddzielają blok znaków a od słowa KONIEC, które umieszczamy na końcu rozszyfrowanego tekstu (tzn. po stwierdzeniu, że rozszyfrowaliśmy ostatni jego wiersz). Uwaga! Odczytana z tekstu liczba dwucyfrowa może być postaci 01 lub 07 (oznacza to odpowiednio wartości 1 i 7). Może być także postaci 00- w tym przypadku bez obliczania NWD przyjmujemy, że dany blok tekstu od następnego nie dzieli żaden odstęp, po prostu znaki kolejnego bloku tekstu zaczynają się w kolejnym wierszu.

Napisz program, który odszyfrowuje zapisane w pliku *szyfr.txt* przyjmując, że wiersze te zostały zaszyfrowane zgodnie z opisanymi wyżej zasadami. Rozszyfrowane bloki tekstu należy umieścić w pliku o nazwie *wynik.txt*.

Należy założyć, że jedyną sytuacją, w której nie można rozszyfrować wiersza jest ta, w której rozszyfrowanym kodem ASCII znaku jest kod zawarty pomiędzy 27, a 32, a poza tym wszystkie wiersze są zaszyfrowane poprawnie i znajdziesz w nich odpowiednią liczbę odpowiednich cyfr, które w zgodzie z przedstawionymi zasadami pozwolą dany wiersz rozszyfrować. Jeżeli kod rozszyfrowywanego znaku jest liczbą od 27 do 32 zamiast bloku tekstu generujemy komunikat BEZ ZNAKU zaś odstęp do kolejnego bloku danych będzie równy 1. Po rozszyfrowaniu ostatniego bloku tekstu i pozostawienie po nim odpowiedniego wynikającego z szyfru odstępu umieszczamy w ostatnim wierszu pliku wynik.txt słowo „KONIEC".

Dane wejściowe:

Plik ***dane.txt*** zawierający w swoim pierwszym wierszu jedną liczbę naturalną **n** większą od 1 oraz nie większą niż 25. W kolejnych n wierszach tego pliku znajdują się teksty zawierające dowolne znaki, w tym pewną liczbę pojedynczych cyfr pozwalającą w pełni rozszyfrować zakodowany tam blok tekstu zgodnie z zasadami szyfrowania opisanymi w zadaniu (liczba cyfr jest odpowiednia do odszyfrowania tekstu, są one też dobrane zgodne z omówionymi zasadami i program nie musi tego sprawdzać !). W żadnym z tekstów nie ma spacji. Pierwsza cyfra znajdująca się w każdym z zaszyfrowanych wierszy tekstu nigdy nie jest zerem. Nie są też na pewno zerami cyfry odpowiadające w szyfrze liczbie znaków w wierszu oraz liczbie wierszy w danym bloku tekstu. Długość jednego wiersza tekstu to minimum 5 znaków (tyle pojedynczych cyfr wystarczy do jego odszyfrowania właśnie w szczególnym, minimalnym przypadku), zaś maksymalnie długości wierszy tekstu nie przekraczają 50 znaków.

Dane wyjściowe:

Umieszczone w pliku ***wynik txt*** n bloków tekstu oraz dodatkowo jako ostatni blok tekstu złożony z jednego tylko wiersza z napisem KONIEC. Każdy blok tekstu poza ostatnim oraz blokiem, w którym rozszyfrowano niedrukowalny kod ASCII zawiera pewną liczbę wierszy, a w każdym z nich taką sama liczbę takich samych znaków. Drukowany w danym bloku znak, liczba znaków w wierszu, liczba wierszy oraz liczba wierszy odstępu, które należy zachować

przed kolejnym blokiem tekstu wynikają z rozszyfrowanego przy zastosowaniu omówionych w zadaniu zasad wiersza tekstu odpowiadającego kolejnemu blokowi. Jeżeli rozszyfrowany znak tekstu nie jest znakiem drukowalnym (kody ASCII od 28 do 32) to blok tekstu odpowiadający takiemu wierszowi z pliku szyfr.txt składa się tylko z jednego wiersza zawierającego tekst „BEZ ZNAKU”, a od kolejnego bloku tekstu dzieli go tylko jeden wiersz odstępu. Z kolei po ostatnim bloku zawierającym tylko wiersz z tekstem „KONIEC” odstępu nie pozostawia się.

Przykład

Jeżeli w pliku *szyfr.txt* mamy następujące dane:

3
Qwerty3Ala ma kota 9pies48banany9cytrynaalarm10ja
LEW2Ola2cx32jk7hj1
Lew1Ola2h9miu34ty12

to w pliku *wynik.txt* pojawić się powinien następujący wydruk:

cccc
cccc
cccc
cccc
cccc
cccc
cccc
cccc

zzz
zzz
zzz
zzz
zzz
zzz
zzz

BEZ ZNAKU

KONIEC

Krótkie objaśnienie do przykładu:

Pierwszy wiersz generuje 8 wierszy po 4 znaki c (kod 99) i dwie linijki odstępu do kolejnego bloku tekstu (największy wspólny dzielnik z liczb 10 oraz 8).
Drugi wiersz generuje 7 wierszy po 3 znaki z (kod 122) i jedną linijkę odstępu do kolejnego bloku tekstu (największy wspólny dzielnik z liczb 71 oraz 7)
Trzeci wiersz daje blok BEZ ZNAKU (rozszyfrowano kod ASCII 29) i jedną linijkę do kolejnego bloku tekstu.
Ostatni bloku tekstu zawiera wiersz z tekstem KONIEC, a po nim nie ma już żadnych odstępów.

Do oceny oddajesz plik zawierający kod źródłowy napisanego przez Ciebie programu. Nazwa tego pliku to *Zadanie3.*

\ **10 punktów**

Zadanie 4

I. W załączonym pliku ***Zalacznik-Zadanie4,pdf*** przedstawionych zostało kilku początkowych przedstawicieli pewnej rodziny figur. Przedstawiciele rodziny o kolejnych numerach w pewien sposób zależą od swoich poprzedników. Twoim zadaniem jest napisanie programu, który dla podanych parametrów n (liczba naturalna - numer figury w rodzinie) oraz a (to jedyny parametr związany z wielkością figur -uwidoczniony na wszystkich rysunkach jako długość swoistego „pnia" tych figur) stworzy na ekranie figurę o kształcie odpowiadającym podanemu n oraz wielkości wynikającej z a . Proporcje pozostałych elementów uwidocznionych w załączniku należy sobie dobrać samodzielnie (np. na rysunkach: promień okręgu w figurze numer 0 przyjęto jako a/2, a kąt nachylenia „poddrzew" w kolejnych figurach, w stosunku do pionu jako 45 stopni, zaś proporcje „poddrzew" w odniesieniu do całości jak 1:2, ale nie jest to obowiązująca sugestia).

Dane wejściowe:

Dwie liczby: naturalne **n (**zakres od 1 do 7) określająca numer figury oraz dowolne **a** (większe od zera) określającą pokazaną na rysunkach w załączniku długość tzw. pnia. Kształt poszczególnych figur przedstawiono w pliku *Zalacznik-Zadanie4,pdf,* zaś inne wymiary należy przyjąć samodzielnie w proporcji do a. Ze względu na różne środowiska, które mogą być zastosowane do rozwiązania tego problemu nie określa się szczegółowo sposobu wprowadzenia danych.

Dane wyjściowe:

Wyświetlona na ekranie figura, której kształt dla danego n jest zgodny z tym przedstawionym w pliku *Zalacznik-Zadanie4,pdf* , a długość tzw. pnia wynosi a.

Do oceny oddajesz pliki *Zadanie4.I* zawierający kod napisanego przez Ciebie programu.

II.

Oto przykładowe konstrukcje tworzone z ułamków zwykłych, z których każda kolejna w istotny sposób zależy od poprzedniej. Dla większej czytelności każdą konstrukcję zapisano w nowej linii, natomiast nie obliczano ich wyniku.

$$\frac{1}{2}$$

$$\frac{\frac{1}{2}}{\frac{1}{2}+\frac{1}{3}}$$

$$\frac{\frac{1}{2}+\frac{1}{3}}{\frac{1}{2}+\frac{1}{3}+\frac{1}{4}}$$

$$\frac{\frac{1}{2}+\frac{1}{3}+\frac{1}{4}}{\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}}$$

Wypisano tu cztery pierwsze konstrukcje. Następne są tworzone wg tej samej reguły. Twoje zadanie polega na napisaniu programu, który wykorzystując rekurencję (i tylko z niej można korzystać w tym zadaniu) dla danego parametru **n** obliczy wynik dla n-tej konstrukcji (wynik podajemy jako ułamek dziesiętny -wystarczy po prostu ten wynik obliczyć, nie trzeba podawać postaci ułamka zwykłego). Konstrukcja numer 1 to prostu wartość 0.5, zaś konstrukcja o numerze n (n>1) to ta, w której w ostatnim dodawanym ułamku w mianowniku, jak można to zauważyć w podanym wyżej przykładzie, w jego mianowniku posiada wartość n+1.

Dane wejściowe:

Plik tekstowy ***rekurencja.txt*** zawierający w pierwszym wierszu liczbę naturalną **n,** większą od zera i nie większą niż 25, oznaczającą numer konstrukcji tworzonej wg podanych w zadaniu zasad.

Dane wyjściowe:

Zapisany w pliku ***wynik.txt*** zawierający w jedynym wierszu obliczony jako wartość dziesiętną wynik dla n-tej konstrukcji tworzonej wg podanych w zadaniu zasad.

Do oceny oddajesz pliki *Zadanie4.II* zawierający kod napisanego przez Ciebie programu

10 punktów

Zadanie 5

Rozważmy ciąg **n** liczb naturalnych. Należy je uporządkować wg kryterium jakie stanowi nie ich wielkość, ale liczba posiadanych przez każdą liczbę podzielników. Dla przykładu w tym porządku liczba 17 jest mniejsza od liczby 6 (17 jest naturalnie większa od 6, gdy stosujemy zwykłe porządkowanie rosnące) , gdyż ma tylko 2 podzielniki (1,17), zaś liczba 6 ma ich aż 4 (1,2,3,6). Dodatkowo oprócz przyjęcia tego nietypowego kryterium porządkowania samo porządkowanie też nie przypomina klasycznego. Liczby w ciągu mają być uporządkowane w taki sposób, że najpierw należy na początku ciągu umieścić największą z nich (naturalnie ze względu na kryterium liczby podzielników) poprzez jej zamianę z liczbą, która jest w tej chwili na pierwszym miejscu w ciągu nie zmieniając porządku wśród pozostałych liczb, a następnie bezpośrednio po niej umieszczać takie liczby (biorąc pod uwagę ich kolejność w ciągu), z których każda następna ma tę cechę, że jest nie większa (niezmiennie ze względu na kryterium liczby podzielników) od swojej poprzedniczki umieszczonej w uporządkowanej w ten sposób części ciągu. Inne liczby, które „psują" ten porządek umieszczamy na końcu ciągu tzn. po wspomnianej uporządkowanej nierosnąco ze względu na liczbę podzielników części ciągu, w dowolnej kolejności.

Uwaga ! Jeżeli dwie lub więcej liczb mają największą i równą sobie liczbę podzielników to na początku ciągu należy umieścić tę spośród nich , która w pierwotnym ciągu występowała jako pierwsza (miała najniższy numer).

Rozważmy ciąg:

3(2) 8(4) 2(2) 4(3) 24(8) 3(2) 17(2) 51(4)

Dla ułatwienia w nawiasie podano liczbę podzielników każdej z liczb, co jest istotne dla przebiegu porządkowania.

Najwięcej podzielników ma liczba 24 więc zgodnie z podanymi wcześniej regułami należy ją umieścić na początku ciągu poprzez zamianę ze znajdującą się tam obecnie liczbą 3. W efekcie tej operacji otrzymujemy następujący ciąg liczb:

24 8 2 4 3 3 17 51

Teraz zaczynamy właściwe porządkowanie, o którym mowa wyżej. Po liczbie 24 mamy liczbę 8, która ma mniej podzielników niż 24 więc niczego nie trzeba zmieniać, następnie jest liczba 2, która ma z kolei mniej podzielników niż 8 ,a więc nadal jest bez zmian. Następna liczba 4 „burzy" ten porządek bo ma więcej podzielników niż występująca obecnie przed nią liczba 2, na razie zatem nie uwzględniamy ją w części liczb, których liczba podzielników tworzą ciąg nierosnący. Po liczbie 2 powinny być zatem umieszczone najpierw dwie liczby 3 i również liczba 17 (wszystkie mają po dwa podzielniki, a więc tyle samo ile liczba 2 i mogą występować po niej). Ostania liczba 51 ma 4 podzielniki więc podobnie jak wcześniej opisana ze względu na tę samą własność liczba 4 „zaburza" porządek i wraz z liczbą 4 powinna się znaleźć w dowolnej kolejności na końcu ciągu. Ostatecznie akceptowalnym uporządkowaniem jest:

24 8 2 3 3 17 4 51

przy czym na liczbie 17 kończy się część ciągu, która zawiera liczby , których liczba podzielników tworzy ciąg nierosnący oraz występujące w takiej kolejności w stosunku do siebie w jakiej występowały w początkowym ciągu. Po liczbie 17 znajdują się liczby 4 i 51, które musiały być tu umieszczone, gdyż „zaburzały" porządek, w którym każda następna liczba miał co najwyżej tyle podzielników, ile jej poprzedniczka w ciągu.

Ciekawostka: czy uporządkowany wg opisanych zasad ciąg będący rozwiązaniem problemu może być identyczny z ciągiem początkowym ? Otóż może. Wystarczy, że pierwszym jego elementem jest liczba o największej w ciągu liczbie podzielników, a każda kolejna ma co najwyżej tyle podzielników ile jej poprzedniczka. Ale jest też inna możliwość. Pierwszym elementem jest liczba o największej liczbie podzielników, a po niej mamy już tylko liczby zaburzające opisywany porządek(poza drugą)- w tym momencie są już jakby na końcu ciągu. Tak byłoby na przykład dla ciągu: 80 3 12 24.
Twoim zadaniem jest napisanie programu, który dla dowolnego ciągu n liczb naturalnych wykona omówione porządkowanie jego elementów.

Dane wejściowe:

Plik tekstowy ***ciag.txt*** zawierający w pierwszym wierszu liczbę naturalną **n,** większą od zera oznaczającą liczbę elementów ciągu, a w kolejnych n wierszach liczby naturalne będące elementami tego ciągu (po jednej liczbie w każdym wierszu). Zakładamy, że liczby ciągu są nie większe niż 10^9.

Dane wyjściowe:

Zapisane w pliku ***porzpodziel.txt*** w kolejnych wierszach ***(***po jednej liczbie w wierszu***)*** liczby ciągu umieszczonego poprzednio w pliku ciąg,txt , ale tym razem w kolejności od liczby, która ma najwięcej podzielników i którą należy umieścić na początku ciągu poprzez kolejne liczby, z których każda następna ma podzielników nie więcej niż jej poprzedniczka (ale z zachowaniem ich kolejności w pierwotnym ciągu). Liczby, które w pierwotnym ciągu „zaburzają" ten porządek należy umieścić w dowolnej kolejności na końcu ciągu tzn. po zakończeniu malejącego ze względu na liczbę podzielników fragmentu ciągu utworzonego na podstawie pierwotnego ciągu.

Przykład

Jeżeli w pliku ciąg.txt mamy następujące dane:

6

23

128

80

12

11

48

to w pliku ***porzpodziel.txt*** powinniśmy otrzymać następujące wyniki:

80

128

23

11

12

48

Komentarz: liczby 80 i 48 mają tyle samo podzielników (10), ale w pierwotnym ciągu liczba 80 występowała jako pierwsza z nich więc jest na początku ciągu uporządkowanego (zamieniając się miejscami z liczbą 23). Kolejnymi elementami ciągu nierosnącego ze względu na liczbę podzielników będą 128 (8 podzielników), 23(2 podzielniki). Liczba 12 ma więcej podzielników niż 23 (bo 5) więc kolejną będzie 11 (jako liczba pierwsza podobnie jak 23 ma dwa podzielniki). Liczba 12 , która będąc na „swoim" początkowych miejscu „zaburzała" porządek zostały przesunięte na koniec, gdzie znalazła się wraz z liczbą 48 (naturalnie mogą tam występować także w odwrotnej kolejności).

Do oceny oddajesz plik zawierający kod źródłowy napisanego przez Ciebie programu. Nazwa tego pliku to *Zadanie5.*

11 punktów