POMORSKA LIGA ZADANIOWA *ZDOLNI Z POMORZA*

Konkurs dla uczniów dla uczniów klas VII i VIII szkoły podstawowej województwa pomorskiego w roku szkolnym 2021/2022

Etap II – powiatowy

Przedmiot: Informatyka

Przed przystąpieniem do rozwiązywania zadań zapoznaj się z instrukcją.

INSTRUKCJA

1. Oprócz arkusza z treścią zadań otrzymujesz załącznik niezbędny do rozwiązania zadania numer 1. Jego nazwa jest określona w treści zadania. Przed przystąpieniem do rozwiązywania sprawdź, czy na pewno pobrała(e)ś ten plik. Nie wolno używać własnych plików zamiast tego załącznika.
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

Pewien geolog podczas badań na Antarktydzie zbierał pewien ciekawy, występujący tylko na tym kontynencie okaz kamieni. Kamienie te są dlatego takie interesujące, gdyż są okrągłe. Z wcześniejszych przygotowań teoretycznych geologa wynika, że w zaokrągleniu do pełnej liczby centymetrów średnica kamieni mieści się w ogromnej większości w zakresie od 1 do 12 centymetrów (innymi słowy jest liczbą naturalną z zakresu 1-12). Sporadycznie tylko (ale jest to możliwe) znaleźć można okazy większe.

W załączonym pliku ***Zalacznik-Zadanie1-kamienie.txt*** zgromadzono (nieuporządkowane) średnice 240 kamieni, które udało się geologowi odnaleźć w trakcie dwutygodniowych badań (ich średnica zgodnie z uwagą wyżej podawana jest wyłącznie w pełnych centymetrach).

Na podstawie danych z tego pliku rozwiąż następujące problemy:

a) oblicz ile kamieni należy do kolejnych przedziałów:(1-2), (3-4),(5-6),(7-8), (9-10), (11-12), a ile zebranych okazów wbrew przewidywaniom teoretycznym ma średnicę większą niż 12 cm. Uzyskane liczebności poszczególnych przedziałów oraz liczbę kamieni o średnicy większej niż 12 przedstaw na czytelnym wykresie,

b) geolog zakłada, że kamienie o różnych średnicach występują z podobną częstotliwością, a więc kamienie o żadnej średnicy w szczególny sposób nie dominują, ani też nie występują szczególnie rzadko. Wg tej teorii w przypadku 240 okazów powinniśmy otrzymać wyniki bardzo bliskie temu, żeby w każdym z 6 przedziałów podanych w punkcie a) znalazło się 40 sztuk kamieni. Sprawdź prawdziwość tej hipotezę obliczając dla każdego z przedziałów różnicę (co do wartości bezwzględnej) między faktyczną liczbą kamieni w danym przedziale, a tą wynikającą z hipotezy. Podaj osobno największą z tych różnic, a wszystkie różnice dla kolejnych przedziałów przedstaw na czytelnym wykresie. Nie uwzględniamy w tym punkcie jako odrębnego przedziału okazów, których średnica przekracza 12 cm,

c) dla każdego z przedziałów (ponownie nie uwzględniamy kamieni o średnicy większej niż 12 cm) podaj jaki procent stanowią **w nim** kamienie o mniejszej, a jaki o większej średnicy (z dwóch możliwych dla każdego przedziału),

d) oblicz średnią średnicę kamienia, a także różnicę między kamieniem o największej i najmniejszej średnicy znajdujących się w danych (uwzględniamy wszystkie kamienie),

e) inna (niż w punkcie b) hipoteza geologa zakłada, że liczba szczególnie wartościowych kamieni (a są to te, których średnica wynosi dokładnie 6 cm) nigdy nie będzie niestety większa niż 10 % wszystkich zebranych w danym badaniu okazów. Sprawdź tę hipotezę dla podanych danych,

f) dla każdego przedziału można obliczyć w sposób nieco zmodyfikowany średnią średnicę kamienia w tym przedziale. Uzyskamy ją ze wzoru.

$$\frac{n_1 x_1 + n_2 x_2}{n_1 + n_2}$$

n_1 -liczba kamieni o pierwszej wielkości średnicy w danym przedziale (naturalnie w każdym z przedziałów mamy kamienie o tylko dwóch średnicach),

x_1- pierwsza średnica w danym przedziale,

n_2 -liczba kamieni o drugiej wielkości średnicy w danym przedziale ,

x_2- druga średnica w danym przedziale.

Oblicz z powyższego wzoru średnią średnicę kamienia dla każdego z 6 przedziałów (nie uwzględniamy kamieni o średnicy powyżej 12 cm), a następnie oblicz (też dla każdego z przedziałów) wartość bezwzględną z różnicy między uzyskaną średnia dla danego przedziału , a umownym środkiem tego przedziału (dla pierwszego przedziału będzie to 1.5, dla drugiego 3.5 itd.),

g) geolog zdaje sobie sprawę, że jego pomiary w terenie mogą być obarczone błędem, poza tym zaokrągla wyniki do pełnych centymetrów. Przyjął jednak, że taki błąd nie może być większy niż 1 cm, tzn. średnica każdego kamienia może być , albo o 1 cm mniejsza (poza kamieniami o średnicy równej właśnie 1 cm), albo o 1 cm większa niż znajdująca się w danych. Przeprowadź korektę wszystkich danych właśnie o 1 cm, ale dla każdego kamienia tylko jedną z nich tzn. taką, która spowoduje, że znajdzie się on teraz w innym przedziale – zauważ, że dla każdego kamienia tylko jedna z korekt średnicy o 1 cm ta w górę, albo ta w dół zmienia jego przynależność do przedziału.(**Uwaga ! Korekta nie może być wykonana ręcznie**, **ale być wynikiem**

odpowiednich obliczeń !) Dla tych skorygowanych danych przeprowadź jeszcze raz obliczenia opisane w punkcie b tzn. oblicz ile teraz kamieni należy do poszczególnych przedziałów i ile ma średnicę większą niż 12 cm oraz dla każdego przedziału oblicz różnicę (co do wartości bezwzględnej) między nową liczbą kamieni w danym przedziale, a tą wynikającą z hipotezy o tym, ze w każdym z przedziałów powinno być teoretycznie 40 kamieni (różnicy nie obliczamy dla grupy kamieni ze średnicą powyżej 12 cm). Nie trzeba już teraz obliczać maksymalnej z tych różnic, ani tworzyć wykresu, który miał zostać wykonany w punkcie b.

Do oceny oddajesz plik zawierający komputerową realizację obliczeń, na podstawie których uzyskasz rozwiązanie zadania. Nazwa tego pliku to *Zadanie1*. Jeśli tworzysz więcej plików z realizacją obliczeń to nazwij je *Zadanie1a, Zadanie1b* itd. Dodatkowo wyniki liczbowe (wykresy nie) dla punktów od a) do f) umieść w pliku *Zadanie1-wyniki* wyraźnie zaznaczając, , które wyniki, których pytań dotyczą.
Uwaga ! Odpowiedzi liczbowe umieszczone w pliku tekstowym *Zadanie1-wyniki* nie będą mogły być uznane nawet jeśli będą poprawne, o ile nie znajdą potwierdzenia i odzwierciedlenia w zawartości pliku z komputerową realizacją obliczeń .

10 punktów

Zadanie 2

Opracuj szablon dokumentu, który pozwoli tworzyć krótkie teksty (opisy) dotyczące wybranych krajów świata. W szablonie winny się znaleźć następujące części z informacjami o danym kraju:
- podstawowe dane (rysunek flagi, stolica, powierzchnia, liczba ludności, położenie geograficzne, link do pliku z hymnem),
- klimat (czyli kilka zdań o typowych dla danego kraju warunkach atmosferycznych),
- z czego znamy ten kraj (opisy minimum 3, ale nie więcej niż 5 postaci, wydarzeń, zjawisk lub innych elementów dzięki którym najczęściej kojarzymy dany kraj),

- największe atrakcje turystyczne (dokładnie 3 miejsca, które warto w danym kraju polecić turystom).

Zadbaj, aby szablon miał własne, stałe (czyli niezależne od później opisywanego na jego podstawie kraju) logo i motto uwidocznione na początku dokumentu oraz związane tematycznie z opisywaniem różnych państw świata.

Naturalnie ponieważ to szablon więc należy tylko przygotować i odpowiednio sformatować podane części wyraźnie wskazując przyszłemu użytkownikowi tego szablonu, gdzie ma wprowadzać odpowiednie treści (stosując np. zapisy typu <Tu wprowadź…>, albo w przypadku obrazów obrazy przykładowe). Szczegóły dotyczące formatowania, zastosowanych stylów, układu dokumentu pozostawia się autorce(autorowi) szablonu. Te elementy powinny uwypuklać estetykę dokumentu, ale punktujemy jego zawartość merytoryczną i to jak dokładnie są wskazówki, gdzie i co należy wpisać korzystając z szablonu. **Nie zapomnij, że od dokładnego szablonu dokumentu oczekujemy, że nie tylko będzie estetycznie sformatowany, ale przede wszystkim tego, że tworzenie na jego podstawie dokumentu będzie dziecinnie łatwe dlatego, że jasno określono jakie treści i gdzie powinny się pojawić.**

Następnie przygotuj szablon prezentacji wspomagającej szablon dokumentu w kontekście prezentowania walorów wybranego kraju. Ten szablon winien spełniać następujące wymogi:

- slajd tytułowy z nazwą i flagą kraju (w szablonie oczywiście w wersji <Tu wstaw> i flaga przykładowa nieistniejącego państwa),
- slajd-menu, którego trzy pierwsze opcje „odsyłają” do odpowiednich slajdów na, których umieszczono fotografie, jakie opisano w tekście. Czwarta opcja odsyła do slajdu końcowego,
- slajdy z fotografiami (dokładnie 3 -tyle ile opisano w tekście atrakcji turystycznych). Na każdym z tych slajdów należy umieścić zdjęcie oraz tytuł sygnalizujący, jaką atrakcję turystyczną widać na tym zdjęciu. W szablonie prezentacji naturalnie zdjęcia są przykładowe, a odpowiednia instrukcja wskazuje, czym trzeba je zamienić. Z każdego slajdu z fotografiami jest zorganizowany powrót do slajdu-menu,
- slajd końcowy, który powinien zawierać link odsyłający do tekstu o danym kraju, przygotowanego na podstawie szablonu dokumentu. Ze tego slajdu nie organizujemy bezpośredniego powrotu do żadnego z poprzednich slajdów.

W oparciu o skonstruowane szablony tekstu i prezentacji przygotuj opis (tekst i prezentację) jednego z dwóch krajów skandynawskich (można sobie wybrać dowolny z tych dwóch): Norwegii, albo Szwecji. Niezbędne informacje pozyskaj z sieci internetowej. Zadbaj o poprawność merytoryczną prezentowanych danych, ale i o poszanowanie praw autorskich.

Do oceny oddajesz pliki o nazwie *Zadanie2_szablon_dokument, Zadanie2_szablon_prezentacja* zawierające odpowiednio szablon tekstu i prezentacji, o których mowa w treści zadania oraz *Zadanie2-kraj* (opis Szwecji lub Norwegii wg opracowanego szablonu tekstu) i *Zadanie2-prezentacja* (towarzysząca opisowi Szwecji lub Norwegii prezentacja przygotowana na podstawie szablonu prezentacji). Jeśli będzie to niezbędne do sprawdzenia rozwiązań to dołącz również pliki dodatkowe (techniczne) wykorzystane w trakcie tworzenia plików stanowiących rozwiązanie zadania.

8 punktów

Zadanie 3

W trakcie przetwarzania danych przedstawionych w postaci bitów czyli wartości 0 lub 1 (np. realizacji operacji arytmetycznych) w procesorze maja miejsce rozmaite operacje organizacyjne. Np. pomnożenie liczby przez 2 to przesunięcie ciągu z wartościami bitów o jedną pozycję w lewo i dopisanie zera. Na potrzeby tego zadania zdefiniujemy kilka różnych (niekiedy nie mających wiele wspólnego z rzeczywistą architekturą komputera) operacji na ciągach z wartościami bitów i przypiszemy im własne symbole. Przyjmiemy też, że ciągi z wartościami bitów w naszym zadaniu będą mogły być dowolnej długości (w realiach przetwarzania maja na ogół długości będące wielokrotnością liczby 8):

a) przesunięcie ciągu wartości bitów w prawo o n bitów – skrót P, do którego dodajemy po odstępie liczbę naturalną n (P 1- to np. przesunięcie w prawo o jeden bit).
Ta operacja przesuwa bity o n miejsc w prawo, a na początek ciągu zamiast bitów. które uległy przesunięcie wstawia zera. Operacja P 2 dla następującego ciągu:

100101

daje efekt

001001

(przesunęliśmy wszystkie bity o dwa w prawo, ostatnie dwa niejako się „schowały", a na początku ciągu wstawiliśmy dwa zera).

Zauważmy jeszcze, że gdy wartość n w skrócie operacji jest większa lub równa długości ciągu to wynikiem jest zawsze ciąg o tej samej długości złożony z samych zer np. P5 dla ciągu 101 daje 000,

b) przesunięcie ciągu wartości bitów w lewo o n bitów – skrót L do którego dodajemy po odstępie liczbę naturalną n (L 2- to np. przesunięcie w lewo o dwa bity).

Ta operacja przesuwa bity o n miejsc w lewo, a na koniec ciągu zamiast bitów. które uległy przesunięcie wstawia zera. Operacja L 3 dla następującego ciągu:

100101

daje efekt

101000

(przesunęliśmy wszystkie bity o trzy w lewo, pierwsze trzy niejako się „schowały", a na końcu ciągu wstawiliśmy trzy zera).

Zauważmy jeszcze, że gdy wartość n w skrócie operacji jest większa lub równa długości ciągu to wynikiem jest zawsze ciąg o tej samej długości złożony z samych zer np. L4 dla ciągu 10 daje 00,

c) rotacja ciągu wartości bitów w prawo o n bitów – skrót RP do którego dodajemy po odstępie liczbę naturalną n (RP 2- to np. rotacja w prawo o dwa bity).

Ta operacja przesuwa bity o n miejsc w prawo, ale na początek ciągu zamiast bitów. które uległy przesunięcie wstawia nie zera jak przy przesunięciu, ale bity, które „wypadły" poza ciąg z prawej strony (w kolejności tego „wypadania"). Operacja RP 2 dla następującego ciągu:

1001111

daje efekt

1110011

(przesunęliśmy wszystkie bity o dwa w prawo, ostatnie dwa niejako się „schowały", ale pojawiły się na początku ciągu).

Zauważmy jeszcze, że wartość n dla tej operacji może być większa lub równa długości ciągu. Kiedy jest równa to otrzymujemy ten sam ciąg, który niejako wrócił na swoje miejsce np. RP 3 dla ciągu 110 daje ponownie 110, a np. RP 5 dla tego ciągu daje 101 (po wykonaniu rotacji w prawo o 3 czyli „pełnego obiegu" następuje dalsza rotacja w prawo jeszcze o 2 bity),

d) rotacja ciągu wartości bitów w lewo o n bitów – skrót RL do którego dodajemy po odstępie liczbę naturalną n (RL 2- to np. rotacja w lewo o dwa bity).

Ta operacja przesuwa bity o n miejsc w lewo, ale na koniec ciągu zamiast bitów. które uległy przesunięcie wstawia nie zera jak przy przesunięciu, ale bity, które „wypadły" poza ciąg z lewej strony (w kolejności tego „wypadania"). Operacja RL 3 dla następującego ciągu:

1001111

daje efekt

1111100

(przesunęliśmy wszystkie bity o trzy w lewo, pierwsze trzy niejako się „schowały", ale pojawiły się na końcu ciągu).

Zauważmy jeszcze, że wartość n dla operacji może być większa lub równa długości ciągu. Kiedy jest równa to otrzymujemy ten sam ciąg, który niejako wrócił na swoje miejsce np. RL 4 dla ciągu 1101 daje ponownie 1101, a np. RL 5 dla tego ciągu daje 1011 (po wykonaniu rotacji w lewo o 4 czyli „pełnego obiegu" następuje dalsza rotacja w lewo jeszcze o 1 bit),

e) negacja – -skrót N.

Ta operacja działa na wszystkich wartościach ciągu zmieniając je na wartości przeciwne tzn. 0 zamienia na 1 , a 1 na 0.

Operacja N dla następującego ciągu

11100011

daje efekt

00011100

f) odbicie symetryczne – skrót O.

Ta operacja zamienia miejscami bity położone względem siebie symetrycznie (tzn. pierwszy z ostatnim, drugi z przedostatnim itd.)

Operacja O dla następującego ciągu

11100010

daje efekt

01000111

Napisz program, który odczytuje z pliku tekstowego ***bity.txt*** n ciągów wartości bitów, a na każdym z nich wykonuje operację, której skrót jest zapisany w tym samym wierszu co dany ciąg. Powstałe po wykonaniu operacji zmodyfikowane ciągi wartości bitów powinny być zapisane w pliku ***operacje.txt.***

Dodatkowo w pliku ***operacje.txt*** powinny być zapisane następując wyniki działania programu:

a) łączna liczba zer i jedynek w ciągach wartości bitów zapisanych w pliku ***bity.txt*** (czyli przed wykonaniem wskazanych operacji),

b) łączna liczba zer i jedynek w ciągach wartości bitów zapisanych w pliku ***operacje.txt*** (czyli po wykonaniu wskazanych operacji- zauważmy że niektóre operacje wpływają na liczby zer i jedynek w ciągu bitów),

c) dziesiętną wartość największej i najmniejszej liczby ustaloną na podstawie ciągów wartości bitów (czyli liczb zapisanych w systemie dwójkowym) z pliku ***operacje.txt*** (czyli po wykonaniu operacji).

Dane wejściowe:

Plik ***bity.txt*** zawierający w swoim pierwszym wierszu liczbę naturalną n większa od 2 oraz nie większa niż 40 określającą liczbę ciągów wartości bitów zapisanych w kolejnych wierszach. W każdym z kolejnych n wierszy znajdują się dwa elementy oddzielone spacją, a w wierszach, w których zdefiniowano jedną z rotacji lub jedno z przesunięć 3 elementy oddzielone spacją. Pierwszy element wiersza to ciąg złożony wyłącznie z zer i jedynek (czyli

wartości bitów) o długości nie mniejszej niż 2, a nie większej niż 12. Drugi element to skrót literowy odpowiadający wyłącznie jednej z opisanych wyżej operacji , a więc P , L, RP, RL , N , albo O Operacja ta opisuje w jaki sposób przekształcić ciąg z wartościami bitów podany w tym samym wierszu. W wierszach, w których symbolem operacji jest P, L, RP lub RL znajduje się jeszcze trzeci element, którym jest liczba naturalna mogąca przyjmować wartości od 1 do 150 i będąca parametrem liczbowym danej rotacji lub danego przesunięcia.

Dane wyjściowe:

Umieszczone w pliku ***operacje. txt*** n+3 wiersze w których kolejno powinny się znaleźć:
- w pierwszych n wierszach ciągi wartości bitów odpowiadające w kolejności ciągom zapisanym od wiersza 2 do n+1 w pliku ***bity.txt*** przekształcone do postaci wynikającej ze zrealizowanej na nich operacji, której skrót był umieszczony w odpowiadającym wierszu pliku ***bity.txt*** ,
- w linii n+1 dwie liczby naturalne oddzielone spacją. Pierwsza oznacza łączną liczbę zer zapisanych we wszystkich ciągach wartości bitów z pliku ***bity.txt*** , a druga łączną liczbę jedynek zapisanych we wszystkich ciągach wartości bitów z pliku ***bity.txt*** ,
- w linii n+2 dwie liczby naturalne oddzielone spacją. Pierwsza oznacza łączną liczbę zer zapisanych we wszystkich ciągach wartości bitów z pliku ***operacje.txt*** , a druga łączną liczbę jedynek zapisanych we wszystkich ciągach wartości bitów z pliku ***operację.txt*** ,
- w linii n+3 dwie liczby naturalne oddzielone spacją. Pierwsza oznacza dziesiętną wartość największej liczby zapisanej w postaci jednego z ciągów wartości bitów w pliku ***operacje.txt*** , a druga oznacza dziesiętną wartość najmniejszej liczby zapisanej w postaci jednego z ciągów wartości bitów w pliku ***operacje.txt*** .

Przykład

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

6
10010 P 3
101 L 5

1100 RP 2

101 RL 6

1001 N

11001 O

to w pliku *operacje.txt* powinny się znaleźć następujące dane:

00010
000
0011
101
0110
10011
11 13
14 10
19 0

Krótki komentarz do wyników: dla drugiego ciągu wartości bitów miało miejsce przesuniecie w lewo o liczbę większą niż długość ciągu co zgodnie z wcześniejszym wyjaśnieniem daje ciąg wartości bitów o tej samej długości, ale złożony z samych zer. Dla czwartego ciągu rotacja w lewo następuje o liczbę 6 czyli dokładnie o dwa razy tyle wynosi długość ciągu, co sprawia, że powraca on niejako „na swoje miejsce". Postaci pozostałych ciągów po wykonaniu na nich operacji opisanych w pliku z danymi wynikają z objaśnień w treści zadania. Wartości największej i najmniejszej liczby w ciągu *operacje.txt* wynikają z zamiany odpowiednich liczb dwójkowych na system dziesiętny.

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

13 punktów

Zadanie 4

Jeżeli będziemy obliczać wartości wyrażenia 1/n dla coraz większych naturalnych wartości n to będzie ona coraz mniejsza i będzie zbliżać się do wartości zero. Oczywiście dla żadnego

skończonego n nigdy wartości równej zeru nie otrzymamy. W tym problemie interesuje nas jak szybko wyrażenie 1/n „zbliża się do zera".

Twoje zadanie polega na napisaniu programu, który będzie obliczał wartości wyrażenia 1/n dla kolejnych n począwszy od 1 i będzie robił tak długo, aż różnica (co do wartości bezwzględnej) między aktualnie obliczoną wartością 1/n , a zerem będzie mniejsza od wskazanej liczby d. Obliczenia należy powtórzyć dla kilku wartości d, które będą zapisane w pliku ***dokladnosci.txt.*** Dla każdej wartości d interesują nas dwa wyniki obliczeń: ile razy należało obliczyć wartość 1/n (czyli krótko mówiąc jaką wartość osiągnęło n kiedy różnica między 1/n oraz zerem była co wartości bezwzględnej mniejsza niż d) oraz suma wszystkich obliczonych wartości 1/n począwszy od n=1, a zakończywszy na tej wartości n, przy której przerwano obliczenia. **Dodatkowo wspomnianą sumę należy obliczyć wykorzystując samodzielnie zdefiniowaną funkcję rekurencyjną. Uzyskanie wartości funkcji przy pomocy zwykłej iteracji będzie niżej punktowane.**

Dane wejściowe:

W pierwszym wierszu pliku *dokladnosci.txt* liczba naturalna **n (**zakres od 1 do 10) określająca liczbę kolejnych wierszy, w których zapisano (po jednej w wierszu) liczby rzeczywiste z przedziału (0,0.5> oznaczające wartości d czyli liczby decydującej o zakończeniu obliczania kolejnych wartości wyrażenia 1/n po spełnieniu warunku opisanego w treści zadania.

Dane wyjściowe:

Zapisane w pliku *wyniki.txt* n wierszy. W każdym z wierszy powinny się znaleźć dwie liczby oddzielone spacją. Pierwsza liczba naturalna oznacza wartość n, dla którego zakończono obliczanie wartości wyrażenia 1/n po spełnieniu opisanego w treści zadania warunku dla wartości d pochodzącej z odpowiadającego obliczeniom wiersza pliku *dokladnosci.txt*. Druga liczba to liczba rzeczywista, która jest równa sumie wartości wyrażenia 1/n począwszy od n=1, a skończywszy na wartości n, która jest pierwszą liczbą zapisaną w tym wierszu.

Wartość tej sumy należy koniecznie uzyskać przy pomocy samodzielnie zdefiniowanej funkcji rekurencyjnej. Rozwiązanie iteracyjne będzie uwzględniane, ale znacznie niżej punktowane.

Przykład

Jeżeli w pliku *dokladnosci.txt* mamy następujące liczby:

2

0.3

0.001

to w pliku *wynik.txt* powinny być zapisane następujące dwa wiersze

4 2.08333

1000 7.48547

Komentarz do przykładu: Wartość 1/n różni się od zera o mniej niż 0.3 dla n=4, a o mniej niż 0.001 dla n=1000. Z kolei suma wyrażeń postaci 1/n dla n od 1 do 4 wynosi 2.08333 , a suma tych samych wyrażeń dla n od 1 do 1000 wynosi 7.48547.

Ten przykład jeśli go rozwiążesz pozwoli na ciekawe obserwacje dotyczące zachowania się wyrażenia 1/n i sumy kolejnych wartości tego wyrażenia. Gorąco polecamy takie testy i obserwacje. W związku z tym kilka praktycznych wskazówek i uwag:

a) **nie zaokrąglaj żadnych wyników -pozwoli to na ciekawsze obserwacje,**
b) **aby lepiej analizować wyniki stosuj deklarację liczb wysokiej precyzji,**
c) **dokładności d to oczywiście ułamki (czym mniejsze tym większa dokładność), nie ma sensu podawać liczb większych niż 1- chyba domyślasz się dlaczego ?**
d) **zaobserwuj jak zachowuje się suma przy rosnących wymaganiach co do dokładności (malejących d),**
e) **zaobserwuj , że dla bardzo wysokich wymagań co do dokładności obliczenia mogą się nie zakończyć, można spróbować nawet doświadczalnie znaleźć taką dokładność, przy której trudno już o wynik.**

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

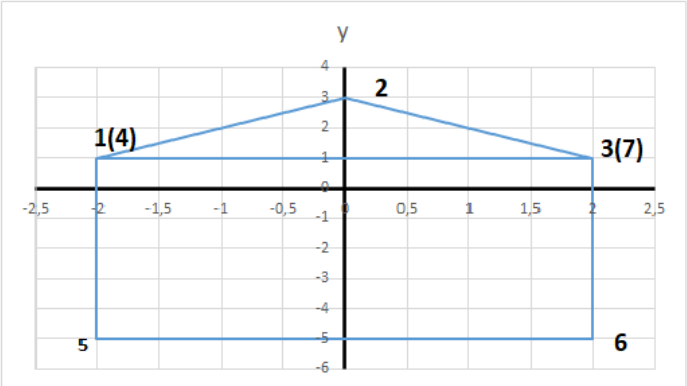
8 punktów

Zadanie 5

W pliku tekstowym umieszczono w kolejnych wierszach opisy n punktów płaszczyzny XOY. Opis pojedynczego punktu to para liczb rzeczywistych oddzielonych spacją. Pierwsza z nich oznacza współrzędną x opisywanego punktu, a druga jego współrzędną y. Kolejne punkty można ze sobą połączyć w odcinki, a te utworzą łamaną, albo wielokąt. Przyjmujmy dodatkowo, że w dwóch różnych wierszach może się pojawić para liczb opisująca ten sam punkt (co oznacza, że wykreślona łamana ponownie przechodzi przez ten sam punkt), ale nie może być tak, że punkty o identycznych współrzędnych występują jeden po drugim czyli żaden kolejny odcinek łamanej nie redukuje się do punktu, nie ma też na pewno dwóch odcinków o identycznych współrzędnych końca i początku (czyli odcinków pokrywających się z wcześniej występującymi). Na pewno nie ma też odcinków takich, że jeden zawiera się w drugim.
Jeśli dla przykładu otrzymalibyśmy następujący opis 7 punktów (liczba w pierwszym wierszu określa liczbę punktów czyli właśnie 7)

7
-2 1
0 3
2 1
-2 1
-2 -5
2 -5
2 1

to po połączeniu tych punktów otrzymalibyśmy w układzie współrzędnych następujący rysunek:



Numery naniesione na rysunkach odpowiadają kolejnym punktom opisu i jednocześnie wskazują kolejność w jakiej te punkty łączono. Zauważmy, że podwójna numeracja oznacza że pewien punkt powtórzył się. W tym przypadku dotyczy to punktu 4, który ma identyczne współrzędne jak punkt 1 oraz punktu 7 o identycznych współrzędnych jak punkt 3.

Twoje zadanie polega na napisaniu programu, który odczyta opisy punktów przygotowane w pliku, a następnie odpowie na następujące pytania:

a) ile odcinków utworzonych z sąsiadujących ze sobą w pliku punktów jest równoległych do jednej lub drugiej osi liczbowej (zliczamy takie odcinki razem , a więc nie liczymy osobno równoległych do osi poziomej i pionowej) – w naszym opisie jak widać z rysunku mamy 4 takie odcinki. **Uwaga ! Nie uwzględniamy odcinków położonych dokładnie na osi – mają swoją osobną statystykę w punkcie c**,

b) ile odcinków utworzonych z sąsiadujących ze sobą w pliku punktów -przecina jedną lub drugą oś (ponownie zliczamy to razem bez rozróżniania, o którą oś idzie, także liczymy sytuacje gdy odcinek przecina obie osie, lub przechodzi przez środek układu współrzędnych) - w naszym opisie jak widać rysunku z dotyczy to wszystkich 6 odcinków. **Uwaga ! Jeżeli tylko jeden z punktów tworzących odcinek należy do którejkolwiek osi, to odcinek klasyfikujemy jako przecinający tę oś,**

a w przypadku, gdy należą do osi zarówno początek, jak i koniec odcinka to odcinek ten liczymy tylko wtedy, gdy są to inne osie. Jeżeli początek i koniec odcinka należą do tej samej osi to uwzględniamy go w punkcie c,

c) ile odcinków utworzonych z sąsiadujących ze sobą w pliku punktów należy do którejkolwiek osi (zliczamy to razem bez rozróżniania, o którą oś chodzi) - w naszym opisie jak widać z rysunku nie ma takich odcinków,

d) ile mamy odcinków utworzonych z sąsiadujących ze sobą w pliku punktów, których środkiem jest punkt przecięcia z osią pionową lub poziomą- w tej statystyce uwzględniamy też odcinki całkowicie zawierające się w jednej z osi. W naszym opisie jak widać z rysunku mamy 2 takie odcinki – łączący punkty 3 i 4 oraz 5 i 6,

e) ile mamy par punktów w których punkt jest położony w odniesieniu do drugiego z tej pary symetrycznie względem którejś z osi, albo środka układu współrzędnych – w naszym opisie jak widać z rysunku mamy 4 takie pary: (1,3), (1,7), (4,3) , (4,7) oraz (5,6), a we wszystkich przypadkach punkt jest względem swojego „sąsiada" w parze położony symetrycznie względem osi pionowej. Zauważmy, że **zaliczamy tu ponownie punkt o tym samych współrzędnych jeśli ponownie leży na łamanej- po prostu traktujemy go niezależnie. Natomiast należy dopilnować, aby identyczne aby pary nie były zliczane dwukrotnie np. w naszym przypadku para (1,3) oraz (3,1) powinna być policzona jako jedna oraz, aby wykluczyć symetrię punktu z samym sobą- owszem łamana może ponownie przechodzić przez punkt o tych samych współrzędnych i przydzielamy mu kolejny numer (jak np. w tym przypadku 1, a potem 4), ale takiej symetrii nie uwzględniamy,**

f) ile uzyskamy po połączeniu punktów trójkątów o bokach utworzonych z kolejno zdefiniowanych punktów, a ile z tych trójkątów to trójkąty prostokątne - w naszym opisie jak widać z rysunku mamy 1 trójkąt, którego wierzchołkami są punkty 1,2,3 (pokrywający się z punktem numer 7, ale trójkąt ten liczymy tylko raz-chodzi o kolejne punkty !). Nietrudno zauważyć, że nie jest to trójkąt prostokątny. Warto jeszcze zwrócić uwagę na znaczenie zwrotu „kolejnych punktów". Jeśli w omawianym przykładzie zamienić kolejność opisu danych i punkt 4 byłby punktem 6, a punkt nr 6 punktem nr 4 to rysunek wyglądałby identycznie, ale trójkąta

nie można byłoby liczyć bo jego boki nie tworzyłyby odcinki łączące kolejne opisane punkty, ale punkty (1) i (2), (2) i (3) oraz (6) i (3).

Dane wejściowe:

W pierwszym wierszu pliku *punkty.txt* znajduje się liczba naturalna **n (**zakres od 2 do 100) określająca liczbę kolejnych wierszy. W każdym z kolejnych n wierszy zapisano oddzielone spacją dwie liczby rzeczywiste z przedziału (-500,500> oznaczające kolejno współrzędną x oraz współrzędną y pewnego punktu na płaszczyźnie.

Dane wyjściowe:

W pliku *statystyki.txt* powinno być umieszczonych 6 wierszy. W pierwszych 5 powinna być umieszczona tylko jedna liczba naturalna oznaczająca dla kolejnych wierszy:
- liczbę odcinków utworzonych z sąsiadujących ze sobą w pliku *punkty.txt* punktów, które są równoległe do jednej lub drugiej osi liczbowej,
- liczbę odcinków utworzonych z sąsiadujących ze sobą w pliku *punkty.txt* punktów, które przecinają jedną lub drugą oś,
- liczbę odcinków utworzonych z sąsiadujących ze sobą w pliku *punkty.txt* punktów, które zawierają się w jednej lub drugiej osi,
- liczbę odcinków utworzonych z sąsiadujących ze sobą w pliku *punkty.txt*, których środkiem jest punkt przecięcia z osią pionową lub poziomą,
- liczbę par punktów z pliku odcinków utworzonych z sąsiadujących ze sobą w pliku *punkty.txt*, dla których jeden punkt jest położony względem drugiego z tej pary symetrycznie względem którejś z osi, albo środka układu współrzędnych.

W 6 wierszu winny być umieszczone dwie liczby naturalne oddzielone spacją. Pierwsza oznacza liczbę trójkątów utworzonych z boków powstałych po połączeniu kolejnych punktów opisanych w pliku *punkty.txt*, a druga powinna odpowiadać na pytanie, ile z tych trójkątów to trójkąty prostokątne.

Przykład (odpowiadający rysunkowi)

W pliku punkty.txt mamy następujące dane:

7
-2 1
0 3
2 1
-2 1
-2 -5
2 -5
2 1

a w pliku ***statystyki.txt*** powinniśmy otrzymać następujące wyniki:

4
6
0
2
5
1 0

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

11 punktów

Razem w całym zestawie 50 punktów.