



## **POMORSKA LIGA ZADANIOWA ZDOLNI Z POMORZA**

**Konkurs dla uczniów szkół ponadpodstawowych i ponadgimnazjalnych województwa pomorskiego**

**w roku szkolnym 2020/2021**

**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 4 (jeden do zadania pierwszego oraz trzy do zadania drugiego). 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 4 oraz 5) miały zawartość i nazwy takie jak określono w treściach zadań.
3. Nie przysyłaj 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. Libre 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. próbą 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 załączonym pliku *Zalacznik-Zadanie1-pracownicy.txt* umieszczono dane 34 pracowników pewnej firmy. W każdym wierszu poza nagłówkowym znajdziemy oddzielone znakiem tabulacji następujące dane pracownika: *LP*, *Nazwisko*, *Data urodzenia*, *Kod stanowiska*, *Wynagrodzenie pracownika*, *Etat*.

Firma bardzo dba o ochronę danych zatrudnionych w niej osób stąd wszystkie dane poza *LP* i *Nazwiskiem* nie są zapisane w standardowej postaci, ale są na różne sposoby zakodowane .

Oto niezbędne objaśnienia dotyczące poszczególnych danych:

*Data urodzenia* – jest wyrażona w systemie rzymskim zapisu liczb w standardzie Rok.Miesiąc.Dzień . Dla przykładu zapis MMI.VII.XIV oznacza datę 2001.07.14.

*Kod stanowiska* – aby nie ujawniać jakie stanowiska są utworzone w firmie przydzielono im kody liczbowe od 1 do 5, a więc każdemu z pracowników przypisano liczbę naturalną z zakresu od do 5, a nie słowny opis jego stanowiska pracy.

*Wynagrodzenie pracownika* – zostało podane w szesnastkowym systemie zapisu liczb. Przy czym ponieważ dla niektórych pracowników w ich wynagrodzeniu pojawia się część ułamkowa (dokładnie dwa miejsca), a więc dla utrzymania jednolitego formatu taka część występuje w wynagrodzeniu wszystkich osób – po prostu Ci, których wynagrodzenie nie zawiera części ułamkowej mają w niej dwa zera. Dodatkowo część całkowita od ułamkowej jest oddzielona kropką. Dla przykładu wynagrodzenie 3DE4.A0 to 15 844 oraz 10/16 grosza. Uwaga ! **W systemie dziesiętnym należy wynagrodzenia przedstawić z dokładnością tylko do dwóch miejsc po przecinku** więc 10/16 to 0.63 (bo dokładnie to 0.625), a więc ostatecznie wynagrodzenie 3DE4.A0 jest równe 15 844.63.

*Etat* – zawiera informację o tym, czy pracownik jest zatrudniony na pełnym etacie, czy na jego części. Przy czym wszystkie dane podano w tym przypadku w systemie dwójkowym zapisu liczb, zawsze z 4 miejscami po kropce (etat 0.1 jest przedstawiany na przykład jako 0.1000).



Jeżeli pracownik jest zatrudniony na pełnym etacie to jego etat jest wyrażony wartością 1.0000 (ponownie kropka oddziela część całkowitą od ułamkowej), a jeżeli nie to ułamkiem dwójkowym, którego część całkowita to na pewno 0. Dla przykładu wartość 0.1010 oznacza, że pracownik jest zatrudniony na  $5/8$  etatu. W danych nie znajdziemy na pewno osób zatrudnionych na 0 etatu.

Rozwiąż następujące problemy:

- a) zbuduj zestawienie pokazujące ile osób jest urodzonych w latach 50-tych, 60-tych, 70-tych, 80-tych, 90-tych w XX wieku oraz ile już od roku 2000 wzwyż (z 2000 włącznie). Przez lata 60-te rozumiemy okres od roku 1960 do 1969,
- b) zbuduj zestawienie pokazujące jaka liczba osób jest zatrudniona na stanowiskach od 1 do 5 oraz jakie jest średnie wynagrodzenie dla każdego stanowiska od 1 do 5,
- c) podaj nazwisko najstarszego pracownika, jest dokładnie jedna taka osoba,
- d) dla każdego pracownika oblicz ile wynosiłoby jego wynagrodzenie wyrażając je w systemie dziesiętnym, gdyby był zatrudniony na pełnym etacie (naturalnie będzie to inna wartość niż ta znajdująca się w danych tylko dla osób zatrudnionych na niepełnym etacie),
- e) oblicz ile osób urodziło się w tym samym miesiącu co najmłodsza z zatrudnionych kobiet (jest tylko jedna taka osoba) - zauważmy, że tylko nazwiska kobiet kończą się na literę „a”,
- f) przyznano specjalny dodatek, który przysługuje tylko osobom zatrudnionym na stanowiskach 4 i 5 i zatrudnionych przynajmniej na  $2/3$  etatu. Jego wartość to 10 % wynagrodzenia osoby, której dodatek się należy. Oblicz wysokość dodatku dla poszczególnych osób (dla tych, którym on nie przysługuje jego wartość to naturalnie 0 zł) podając jego wartość w systemie dziesiętnym,
- g) przedstaw na wykresie łączne wynagrodzenie (czyli z dodatkiem, o którym mowa w poprzednim poleceniu) wyrażone w systemie dziesiętnym dla wszystkich pracowników płci męskiej,
- h) przedstaw na wykresie wyrażone w procentach proporcje zatrudnionych na pełnym etacie oraz zatrudnionych na niepełnym etacie.



Do oceny oddajesz plik(i) zawierający komputerową realizację obliczeń, na podstawie której uzyskasz odpowiedzi na wszystkie postawione w części problemy. Nazwa tego pliku to *Zadanie1* lub jeśli jest ich więcej to kolejne pliki mają nazwy *Zadanie 1a*, *Zadanie1b* itd.. Ponadto odpowiedzi liczbowe na pytania postawione w punktach a)-f) umieść w pliku tekstowym *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.

11 punktów

## Zadanie 2

W pliku *Zalacznik-Zadanie2-wypadki.txt* umieszczono fikcyjne dane dotyczące 60 wypadków zarejestrowanych na przełomie ubiegłego wieku w rejonie pewnego miasta (jest to niestety tylko fragment pełnego rejestru – w rzeczywistości wypadków było dużo więcej). W kolejnych wierszach tego pliku poza wierszem nagłówkowym znajdują się oddzielone znakiem tabulacji: *Nr*, *Data wypadku*, *Nr rejestracyjny pojazdu (chodzi o pojazd uczestnika wypadku zgłaszającego wypadek i szkodę)*, *Wyplacone ubezpieczenie*.

Uwaga ! W kolumnie *Wyplacone ubezpieczenie* w pojedynczych przypadkach pojawić się może liczba -1. Oznacza ona, że dany pojazd nie miał w chwili wypadku ważnego ubezpieczenia (konsekwencjami tego faktu tu się nie zajmujemy, ale chodzi o odróżnienie tej sytuacji od takiej, w której ubezpieczenie było ważne, ale odszkodowania nie przyznano i wtedy powinna w tej kolumnie być kwota 0 zł- choć w tych danych nie ma akurat takiego przypadku).

Dodatkowo dysponujemy następującymi danymi dotyczącymi ubezpieczyli związanych z kolejnymi wypadkami (numeracja wypadków odpowiada tej z pliku *Zalacznik-Zadanie2-wypadki.txt*) :

Nazwa ubezpieczyciela	Liczba wypadków	Numer wypadku, w którym wypłacał ubezpieczenie
PZU	21	1,2,5,9,13,15,19,24,25,26,30,36,37,38,43,44,47,52,53,54,57
LINK 4	11	3,6,10,16,31,39,45,46,48,55,56,
GENERALI	8	4,7,17,21,32,35,49,58
WARTA	9	8,14,18,22,28,33,50,51,59
COMPENSA	5	11,23,34,41,60
ERGO HESTIA	3	12,29,40

Naturalnie w tabeli nie ma tych trzech numerów wypadków, których uczestnik nie miał ważnego ubezpieczenia. Niekoniecznie też ten sam ubezpieczyciel musi być związany z kolejnym wypadkiem danego samochodu - ubezpieczyciela niekiedy się zmienia. Dane te znajdziesz również w załączonym pliku *Zalacznik-Zadanie2-dane1.txt*. W każdym wierszu umieszczono oddzielone spacją: *nazwę ubezpieczyciela liczbę wypadków, którymi się zajmował, a potem numery kolejnych wypadków*.

W kolejnej tabeli znajdziemy natomiast nazwiska 8 policjantek i policjantów (naturalnie fikcyjne) popularnej drogówki, którzy sporządzali raport z kolejnych wypadków zarejestrowanych w pliku *Zalacznik-Zadanie2-wypadki.tx* (numeracja wypadków odpowiada tej z pliku).

Nazwisko policjanta	Liczba wypadków	Numer wypadku, dla którego sporządził raport
Grozny	10	1,4,7,18,22,27,34,44,49,59
Laskawy	6	2,8,11,23,30,32
Polubowny	8	3,12,19,29,33,45,50,56
Natarczywy	8	5,13,35,37,43,46,57,60
Kompetentna	7	6,21,28,36,41,51,58
Marudny	7	9,14,20,26,38,47,52
Dziwny	6	10,15,25,31,40,42
Markotna	8	16,17,24,39,48,53,54,55

Dane te znajdziesz również w załączonym pliku *Zalacznik-Zadanie2-dane2.txt*. W każdym wierszu umieszczono oddzielone spacją: *nazwisko policjantki(policjanta)*, *liczbę wypadków*, *którymi się zajmował*, a potem numery kolejnych wypadków.

I. Napisz program, który zmodyfikuje istniejący plik załącznika *Zalacznik-Zadanie2-wypadki.txt* oraz utworzy nowe pliki w oparciu o dostępne dane w załączonych plikach tzn.

a) utworzy plik *ubezpieczyciele.txt* zawierający 6 wierszy (tyle ile towarzystw). W każdym wierszu powinny się znaleźć oddzielone spacją dwie informacje numer identyfikacyjny firmy ubezpieczającej (np. 1,2,3 itd.) oraz nazwa firmy ubezpieczającej. Doda też wiersz tytułów z następującymi informacjami: *Idu* oraz *Nazwa\_ubezp*,

b) utworzy plik *policja.txt* zawierający 8 wierszy (tyle ilu jest policjantek i policjantów). W każdym wierszu powinny się znaleźć oddzielone spacją dwie informacje numer identyfikacyjny policjanta (np. 1,2,3 itd.) oraz jej (jego) nazwisko. Doda też wiersz tytułów z następującymi informacjami: *Idp* oraz *Nazwisko*,

c) korzystając z danych z plików *Zalacznik-Zadanie2-dane1.txt* oraz *Zalacznik-Zadanie2-dane2.txt* uzupełni plik *Zalacznik-Zadanie2-wypadki.txt* tworząc na jego podstawie plik *wypadki-uzupelnione.txt*, w którym przy każdym wypadku pojawią się





dodatkowe dwie informacje oddzielone od dotychczasowych danych i od siebie spacją tzn. numer identyfikacyjny towarzystwa ubezpieczeniowego, które realizowało wypłatę dla danego wypadku oraz numer identyfikacyjny policjanta (policjantki), którzy sporządzali raport dotyczący tego wypadku, Dodaj również tytuły nowych kolumn w brzmieniu *Nr\_ubezp* i *Nr\_polic*. Program powinien sprawić, by wszystkie dane były oddzielone spacją. Uwaga ! Dla wypadków, w których kierowca nie posiadał ważnego ubezpieczenia należy jako numer ubezpieczyciela dopisać 0,

d) utworzy nowy plik *wypadki-data.txt*, który będzie zawierał dane tylko tych wypadków z pliku *wypadki-uzupelnione.txt*, które miały miejsce po określonej przez uruchamiającego program ( z klawiatury) dacie. Rodzaj odstępu między danymi w tym pliku nie ma znaczenia.

**Do oceny w tej części zadania oddajesz plik zawierający kod programu o nazwie *Zadanie2-I* oraz pliki, które powstały w wyniku jego uruchomienia tzn. pliki *ubezpieczyciele.txt*, *policja.txt*, *wypadki-uzupelnione.txt* oraz *wypadki-data.txt* skonstruowany dla wariantu z datą 01.01.2000 (czyli zawierający dane wypadków z pliku *wypadki-uzupelnione.txt*, które miały miejsce po 1.01.2000).**

II. Na podstawie danych umieszczonych w plikach związanych z pierwszą częścią zadania rozwiąż następujące problemy (tym razem już przy pomocy dowolnego narzędzia, nie musi być nim konieczne program):

1. Utwórz zestawienie zawierające powtarzające się rejestracje, czyli rejestracje tych samochodów, które uczestniczyły w wypadkach w wymienionym okresie przynajmniej 2 razy.
2. Utwórz zestawienie nazw ubezpieczycieli, którzy wypłacali odszkodowanie za wypadki mające miejsce w roku 2002.
3. Utwórz zestawienie zawierające rejestrację samochodu oraz nazwisko policjanta sporządzającego raport dla wypadków, w których uczestniczyły pojazdy bez ważnego w tym momencie ubezpieczenia.





4. Utwórz zestawienie, które pokaże dla każdego policjanta sumę odszkodowań jaka została wypłacona we wszystkich wypadkach, w których dany policjant sporządził raport powypadkowy. Suma ta powinna być pokazana w rozbiciu na ubezpieczycieli.
5. Utwórz zapytanie, które pozwoli dla podanego jako parametr zapytania numeru wypadku podać jego datę, numer rejestracyjny pojazdu uczestniczącego w wypadku oraz nazwisko policjanta sporządzającego raport powypadkowy. W pliku z wynikami podaj wynik tego zapytania dla wypadku numer 50.

**Do oceny w tej części zadania oddajesz plik(i) zawierający komputerową realizację obliczeń, na podstawie której uzyskasz odpowiedzi na wszystkie postawione w drugiej części zadania problemy. Nazwa tego pliku to *Zadanie2-II* lub jeśli jest ich więcej to kolejne pliki mają nazwy *Zadanie 2-IIa, Zadanie2-IIb itd.* Ponadto załączasz plik tekstowy *Zadanie2-II-odpowiedzi.txt* zawierający odpowiedzi dla problemów 1-5.**

**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.**

**8 punktów**

### **Zadanie 3**

Po pewnej autostradzie poruszają się pojazdy. W pewnym momencie docierają one do punktu poboru opłat, który składa się z czterech bramek ponumerowanych liczbami naturalnymi 1,2,3,4. Przyjmujemy, że pojazdy przy punkcie poboru pojawiają się w losowych odstępach czasu, których wartość jest liczbą od 1 do 5. Nie jest przy tym istotne o jaką jednostkę czasu chodzi. Automatyczny system kieruje zbliżający się pojazd do jednej z czterech bramek. System jednak potrafi zawodzić, bramka może być zajęta i wtedy kierowca pojazdu próbuje wybrać sam wolną bramkę sprawdzając ich dostępność w kolejności numeracji bramek (czyli najpierw sprawdza czy wolna jest bramka 1, jeśli nie to czy wolna jest bramka 2, jeśli nie to czy wolna jest bramka 3 i na końcu czy wolna jest bramka 4- naturalnie jedna z tych bramek,



czyli ta, która została wytypowana przez automatyczny system na pewno jest w tym momencie zajęta). Gdyby okazało się, że żadna bramka nie jest w tej chwili wolna to pojazd chwilowo nie może być obsłużony. Nie będziemy rozważać co się wtedy z nim dzieje, czy zjeżdża ponownie do kolejki, czy może wybiera znajdujący się w tym miejscu zjazd z autostrady. Dla potrzeb tego problemu przyjmijmy, że jeżeli pojazd nie mógł być obsłużony to jako numer bramki obsługującej przyjmijmy nie istniejący numer bramki 0. Z kolei jeśli został obsłużony to numerem bramki obsługującej jest jedna z liczb z zakresu 1-4. Przyjmijmy dodatkowo, że czas obsługi pojazdu przy bramce jest liczbą losową z zakresu 1 do 4 (czyli obsługa może trwać od jednej do czterech jednostek czasu) i jest identyczny dla każdej bramki.

Wykorzystując dowolnie wybrane narzędzie zbuduj symulację pracy opisanego punktu poboru opłat w zakresie czasu od 0 do 150. Symulacja rozpoczyna się w chwili  $t=0$ , a kończy się w momencie, gdy dla kolejnego dojeżdżającego do punktu poboru opłat pojazdu wygenerowanym czasem przybycia będzie liczba równa, albo większa niż 150 (orientacyjnie, bo mamy przecież do czynienia ze zjawiskami losowymi). Wynikiem symulacji powinno być zestawienie złożone z następujących 3 kolumn:

Czas przybycia pojazdu do punktu poboru opłat	Bramka proponowana przez system automatyczny	Bramka, która faktycznie zrealizowała obsługę pojazdu
---	--	---

W pierwszej kolumnie powinny być odnotowane kolejne wyznaczone, losowo w zgodzie z opisanymi wyżej warunkami czasy, w których w punkcie poboru pojawił się następny pojazd. W drugiej kolumnie spodziewamy się jednej z liczb od 1 do 4 – będzie to numer bramki, którą do obsługi pojazdu czyli pobrania opłaty wyznaczył system automatyczny. W trzeciej kolumnie mogą się natomiast pojawić liczby od 0 do 4. Liczba z zakresu od 1 do 4 będą oznaczać numer bramki, która faktycznie zrealizowała obsługę (numer będzie identyczny jak ten wyznaczony przez system automatyczny, gdy bramka będzie wolna, ale będzie inny, gdy bramka nie była wolna, a kierowca pojazdu sam wybrał bramkę do obsługi na zasadach opisanych wyżej). Liczba 0 w trzeciej kolumnie oznacza, że pojazd w ogóle nie został obsłużony w punkcie poboru opłat. Naturalnie w celach obliczeniowych możesz tworzyć inne pomocnicze kolumny, ale w zestawieniu przekazywanym do oceny powinny być tylko opisane trzy kolumny.



Na podstawie tej symulacji i uzyskanego zestawienia oblicz ile razy w rozważnym czasie obsługi dokonywały poszczególne stanowiska oraz ile razy obsługa w ogóle nie została zrealizowana (czyli wg przyjętej umowy wykonało ja fikcyjne stanowisko o numerze 0). Te wyniki dla poszczególnych stanowisk (także fikcyjnego o numerze 0) przedstaw następnie na wykresie.

**Do oceny oddajesz plik(i) zawierający realizację opisaną symulacji i dodatkowe wyniki obliczeń oraz wykres. Nazwa tego pliku to *Zadanie 3* lub jeśli jest ich więcej to kolejne pliki mają nazwy *Zadanie 3a*, *Zadanie 3b* itd.**

**10 punktów**

#### **Zadanie 4**

Rozważmy dwa ciągi złożone z niekoniecznie identycznej ilości liczb naturalnych. Przyjmijmy kilka definicji określających podobieństwo jednego z tych ciągów do drugiego.

Ciąg złożony z przynajmniej 3 liczb naturalnych jest  **$S_n$  podobny** do innego ciągu złożonego z przynajmniej 3 liczb naturalnych jeżeli ich największy wspólny podciąg (złożony z elementów położonych w ciągu koło siebie) składa się z  $n$  liczb, a liczba  $n$  jest równa przynajmniej 3.

Dla przykładu ciągi:

2 3 5 12 10 8 16 14 2

oraz

1 3 5 12 10 14 2

są  $S_4$  podobne (pogrubieniem wskazano ich najdłuższy wspólny podciąg, mają jeszcze inny wspólny podciąg złożony z liczb 14 i 2, ale jest on krótszy i nie odpowiada wymogom definicji podobieństwa  $S_n$ ).

Ciąg złożony z przynajmniej 2 liczb naturalnych jest  **$P_n$  podobny** do innego ciągu złożonego z przynajmniej 2 liczb naturalnych jeżeli najdłuższy podciąg rosnący znaleziony tylko w wśród



liczb pierwszych ciągu (niekoniecznie utworzony z położonych koło siebie liczb pierwszych) ma w obu ciągach taką samą liczebność równą  $n$  i równą przynajmniej 2.

Dla przykładu ciąg:

17 24 2 5 3 18

oraz ciąg

3 12 7 18 2 19 23

nie są  $P_n$  podobne, gdyż w pierwszym z nich jeśli wybierzemy tylko liczby pierwsze a więc 17 2 5 3 to najdłuższy podciąg rosnący w tym tak utworzonym ciągu liczb pierwszych tworzą 2 liczby 2 oraz 5, a w drugim ciągu taki podciąg jest złożony z trzech liczb tzn. 2, 19 i 23. Zauważmy, że te ciągi byłyby  $P_n$  podobne (dokładniej  $P_2$  podobne), gdyby zamiast liczby 23 była w drugim ciągu dowolna liczba złożona lub mniejsza od 19 liczba pierwsza.

Utwórzmy hierarchię podobieństwa dla ciągów liczb naturalnych pozwalającą stwierdzać, który z dwóch ciągów jest bardziej podobny do pewnego trzeciego ciągu nazwijmy go wzorcowym:

1. Podobieństwo  $S_n$  jest zawsze mocniejsze niż podobieństwo  $P_n$  (bez względu na wartość  $n$ ). Oznacza to, że ciąg liczb naturalnych jest zawsze bardziej podobny do pewnego wzorcowego ciągu liczb naturalnych niż inny ciąg, gdy jest do wzorcowego  $S_n$  podobny, a ten drugi jest tylko  $P_n$  podobny. Oznacza to też, że jeżeli badamy podobieństwo pewnego ciągu do ciągu wzorcowego i wykazuje on zarówno  $S_n$  jak i  $P_n$  podobieństwo do niego, to jako ostateczną jego charakterystykę przypisujemy mu zawsze  $S_n$  podobieństwo.
2. Każde podobieństwo  $S_n$  lub  $P_n$  jest zawsze mocniejsze niż brak jakiegokolwiek z tych podobieństw. Oznacza to, że ciąg liczb naturalnych jest zawsze bardziej podobny do pewnego wzorcowego ciągu liczb naturalnych niż inny ciąg, gdy jest do wzorcowego  $S_n$  lub  $P_n$  podobny, a ten drugi nie wykazuje żadnego podobieństwa do ciągu wzorcowego.
3. Jeżeli dwa ciągi są jednocześnie  $S_n$  podobne lub jednocześnie  $P_n$  podobne do innego ciągu wzorcowego to bardziej podobny jest ten dla którego liczba  $n$  jest większa, a przy równej liczbie  $n$  ich podobieństwo do ciągu wzorcowego jest nierozróżnialne.
4. Jeżeli dwa ciągi nie wykazują żadnego z dwóch zdefiniowanych podobieństw do pewnego ciągu wzorcowego to bardziej podobny do wzorcowego jest ten, który jest ciągiem



niemalejącym ( a drugi nim nie jest). Jeżeli oba nie są niemalejące, albo oba są niemalejące to ich podobieństwo do ciągu wzorcowego jest nierozróżnialne.

Napisz program, który dla  $m$  ciągów o różnej liczbie elementów  $l_1, l_2, \dots, l_m$ , ale dla każdego ciągu równej przynajmniej 3 oraz dla pewnego ciągu wzorcowego złożonego z  $n$  elementów ( $n > 2$ ) uporządkuje wprowadzone ciągi ze względu na ich podobieństwo do tego ciągu wzorcowego: od najbardziej podobnego do najmniej podobnego do ciągu wzorcowego kierując się zdefiniowanymi w hierarchii podobieństwa kryteriami. Przyjmijmy, że jeżeli podobieństwo dwóch lub większej liczby ciągów do ciągu wzorcowego jest nierozróżnialne to ciąg o niższym numerze porządkowym (wcześniej wprowadzony) poprzedza w uporządkowaniu ciąg o wyższym numerze porządkowym (później wprowadzony).

#### Dane wejściowe:

Plik tekstowy *ciagi.txt* zawierający w pierwszym wierszu liczbę naturalną  $m$ , większą od 2, ale nie większą niż 15 oznaczającą liczbę ciągów, dla których będziemy badać podobieństwo do ciągu wzorcowego. W 2 wierszu pliku *ciagi.txt* zdefiniowano ciąg wzorcowy. Pierwsza liczba w tym wierszu oznacza liczbę elementów  $n$  tego ciągu wzorcowego - ta liczba jest równa przynajmniej 3, ale nie większą niż 20, a dalej w tym wierszu mamy  $n$  liczb naturalnych nie większych niż  $10^6$ , oddzielonych spacją.

W każdym z kolejnych  $m$  wierszy mamy jako pierwszą liczbę naturalną  $l_i$  ( $i=1,2,\dots,m$ ) oznaczającą liczbę elementów w ciągu o numerze  $i$  (ciąg o numerze 1 jest zdefiniowany w wierszu numer 3, o numerze 2 w wierszu numer 4 itd.) - ta liczba jest równa przynajmniej 3, ale nie większą niż 20 a potem mamy oddzielonych spacją  $l_i$  liczb naturalnych należących do ciągu o numerze  $i$  opisywanego w tym wierszu. Zakładamy, że liczby ciągu są nie większe niż  $10^6$ .



### Dane wyjściowe:

Plik *wynik.txt* składający się z  $m$  wierszy. W każdym z wierszy znajduje się jedna liczba oznaczająca numer ciągu (liczba od 1 do  $m$ ) wynikający z kolejności w jakiej ciągi zostały umieszczone w pliku *ciagi.txt*. Kolejność numerów winna odpowiadać uporządkowaniu ciągów ze względu na ich podobieństwo do ciągu wzorcowego tzn. w pierwszym wierszu powinien się znaleźć numer ciągu najbardziej podobnego do ciągu wzorcowego, a w wierszu  $m$  numer ciągu najmniej podobnego do ciągu wzorcowego. Przy porządkowaniu ciągu należy uwzględnić kryteria tworzenia hierarchii podobieństwa opisane w treści zadania.

### Przykład

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

```
6
8 12 3 7 4 11 17 2 3
4 5 5 12 18
6 1 2 7 4 11 13
6 2 12 7 19 84 23
5 6 6 4 8 1
8 1 12 3 7 4 100 101 102
6 4 2 3 5 2 23
```

to w pliku *wynik.txt* powinniśmy otrzymać:

```
5
2
3
6
1
4
```

### Objaśnienie do przykładu:

6 ciągów z pliku *ciagi.txt* ma w kolejności następujące kategorie podobieństwa w stosunku do ciągu wzorcowego:



żadnej kategorii, ale jest niemalejący, S3, P4, żadnej kategorii i nawet nie jest niemalejący, S4, P4.

Kolejność numerów ciągów w pliku *wynik.txt* wynika wobec tego z wcześniej podanych kryteriów dla tworzenia hierarchii podobieństwa ciągów.

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

**11 punktów**

### **Zadanie 5**

Profesor Jan RAM jest wybitnym informatykiem i kieruje Bardzo Ważnym Instytutem Naukowym (BWIN). Instytut to ogromna instytucja zatrudniająca wielu wybitnych fachowców. Mieści się w budynku ze 100 piętrami. Gabinet profesora znajduje się na piętrze numer 100, na niższych piętrach znajdują się pracownie, sekretariaty i gabinety innych pracowników BWIN. Profesor najchętniej zajmowałby się pracą naukową, ale jako dyrektor BWIN choć niechętnie musi też załatwiać inne sprawy i podpisywać różne papiery. Niezbyt lubi to, co odrywa go od pracy naukowej więc jak może stara się uniknąć tych papierkowych obowiązków. Właśnie obmyślił algorytm wg którego codziennie winda zatrzymuje się tylko na wybranych piętrach BWIN zanim dotrze na piętro 100.

Algorytm profesora opiera się na tym, że każdego dnia listę pięter na których ma się zatrzymać winda i to w obie strony wybiera na podstawie ciągów liczbowych. Przy czym w grę wchodzi dwa rodzaje ciągów:

- a) ciąg budowany wg wzoru profesora RAMA. Te wzory nie są skomplikowane, bo są ogólnej postaci:

$k*n+h$ , gdzie  $n=1,2,3,\dots$ , a  $k$  i  $h$  dobrane przez profesora stałe (profesor ustala też ile wyrazów ciągu wyznacza danego dnia dobierając górny zakres  $n$ )

albo

$z*n*n+g$ , gdzie  $n=1,2,3,\dots$ , a  $z$  i  $g$  dobrane przez profesora stałe (profesor ustala też ile wyrazów ciągu wyznacza danego dnia dobierając górny zakres  $n$ )

Stałe do obu typów wzorów to liczby naturalne z zakresu od 1 do 15





Zjazd lub wjazd windą wg wzoru budowanego przez profesora będziemy oznaczać w skrócie np.  $W 2^{*n+3} 7$ , lub np.  $W 3^{*n*n+1} 7$ , gdzie po literze W pojawia się jedna z powyżej opisanych formuł już z konkretnymi wartościami stałych (zwróćmy uwagę, że zawsze we wzorze obecne są gwiazdki będące operatorem mnożenia), a ostatnia liczba podaje zakres zmienności  $n$ . Np. dla pierwszego z dwóch przykładowych zapisów

$$W 2^{*n+3} 7$$

piętremi, na których zatrzymuje się winda wjeżdżając do góry byłyby piętra 5, 7, 9, 11, 13, 15 i 17, bo te wartości wynikają ze wzoru  $2^{*n+3}$  (przy zakresie  $n=1,2,\dots,7$ ). Przy zjeździe wg tej samej formuły i zachowaniu reguły, że piętro 100 jest teraz piętrem nr 1 mielibyśmy piętra 96, 94, 92, 90, 88, 86, 84,

b) dowolny ciąg – profesor sam ustala numery pięter, na których zatrzyma się winda. Ten sam ciąg obowiązuje w danym dniu przy zjeździe z 100 piętra. Zapisywać go będziemy w postaci D 10 15 20 34 50 91. Po literze D są wymienione po prostu kolejne piętra zatrzymań windy. W drodze powrotnej dla tego ciągu byłyby to piętra 91, 86, 81, 67, 51 oraz 10 (piętro 100 to teraz ponownie piętro numer 1).

Profesor ustala parametry dla obu rodzajów ciągów dla danego dnia, ale sam musi zdecydować jaką kombinację wjazdu-zjazdu wybierze w tym dniu. W grę wchodzi jak łatwo zauważyć 4 kombinacje. Nie uwzględniając parametrów, a jedynie literowe oznaczenia dla poszczególnych ciągów kombinacje te oznaczmy następująco: WW, DD, DW, WD. Np. kombinacja WD oznacza, że gdy profesor wjechał w górę to winda zatrzymywała się wybierając piętra wg wzoru (do którego oczywiście należy podać niezbędne parametry), a w dół numery kolejnych pięter wyznaczano były przez wartości dowolnie wybranego przez profesora ciągu.

Dodatkowo w zakresie załatwiania spraw papierowych i innych przez profesora obowiązują następujące reguły:

- a) jeżeli na piętrach, na których staje winda w drodze do gabinetu, bo tak wynika z algorytmu są sprawy do załatwienia to profesor zabiera ze sobą na 100 piętro dokumenty od oczekujących pracowników, jeśli nie, to ignoruje piętra, na których

na niego czekają pracownicy choć na parterze przed wejściem do windy woźny przekazuje mu karteczkę z listą pięter, na których jest oczekiwany. Na tej karteczce znajduje się oprócz numeru piętra symbol P-sprawa pilna lub Z-sprawa zwykła. Profesor nie robi jednak wyjątku nawet dla spraw pilnych -jeśli algorytm nie przewiduje zatrzymania się na danym piętrze to zatrzymania nie będzie bez względu na to jaka sprawa do załatwienia na tym piętrze oczekuje !

- b) na jednym piętrze nigdy nie może czekać na profesora więcej niż jedna sprawa do załatwienia, wykluczone są w związku z tym także takie sytuacje, w których Profesor trafia na to samo piętro, z którego brał dokumenty w drodze powrotnej z gabinetu, a tam czeka na niego już nowa sprawa,
- c) sprawy pilne zabrane w drodze na 100 piętro profesor załatwia w pracy i oddaje dokumenty w drodze powrotnej. Wtedy zatrzymuje się specjalnie na piętrach, na których zabrał dokumenty jadąc do gabinetu nawet jeśli algorytm nie wskazał tych pięter jako miejsc zatrzymania windy w drodze powrotnej,
- d) jeśli na piętro ze sprawą pilną profesor trafi w drodze powrotnej z 100 piętra to zabiera dokumenty z nią związane do domu,
- e) inne sprawy niż pilne profesor zawsze zabiera do domu bez względu na to, czy trafi na dane piętro w drodze do swojego gabinetu czy jadąc z powrotem. Jedyne wyjątek od tej reguły stanowi sytuacja, w której profesor wie ( a potrafi to ustalić błyskawicznie), że reguła algorytmiczna zaprowadzi go również w drodze powrotnej na piętro, z którego jadąc do gabinetu zabrał dokumenty. Wtedy sprawę załatwia w gabinecie i zostawia dokumenty na danym piętrze wracając z pracy.

Ponieważ profesor nie lubi się zbytnio przemęczać sprawami organizacyjnymi również w domu Twoim zadaniem będzie napisanie programu, który dla zdefiniowanej dla danego dnia konkretnej postaci dwóch typów ciągów, przy zdefiniowanej na ten dzień liście spraw do załatwienia, którą na karteczce przekaże woźny oraz przy obowiązywaniu podanych w punktach a-e reguł wskażę jedną z opisanych wcześniej kombinacji opisujących algorytm wjazdu-zjazdu profesora, która w tym dniu sprawi, że profesor weźmie najmniejszą z możliwych liczbę spraw do załatwienia do domu oraz poda liczbę tych spraw.



### Dane wejściowe:

Pliki tekstowe *profesor.txt* oraz *wozny.txt*. W pliku *profesor.txt* znajdują się dwa wiersze, w których zapisano parametry do generowania ciągów, którymi profesor RAM posługuje się wybierając algorytm wjazdu i zjazdu w danym dniu. **Uwaga ! Nie znamy kolejności zapisu typu i parametrów ciągu dla danego dnia, bo profesor zmienia kolejność (inaczej nie wiemy z góry, czy pierwszy wiersz zaczyna się od litery W, a drugi od litery D czy też jest odwrotnie) !** Wiemy, że w wierszu odpowiadającym ciągowi generowanemu wg wzoru zapisano oddzielone spacją: literę W, wyrażenie tekstowe definiujące wzór formuły na generowanie wyrazów ciągu wg zasad opisanych w treści zadania oraz liczbę  $n$  oznaczającą górny zakres zmienności  $n$  (czyli liczbę generowanych wg formuły wyrazów ciągu) -  $n$  jest liczbą z zakresu od 1 do 7, stałe we wzorach liczbami naturalnymi z zakresu 1...15. W wierszu reprezentującym ciąg o dowolnych wyrazach umieszczono najpierw literę D, a po odstępnie równym spacji liczbę  $m$  oznaczającą liczbę wyrazów dowolnego ciągu określającego numery kolejnych pięter ( $13 > m > 0$ ), a po tej liczbie  $m$  liczb naturalnych z zakresu od 1 do 100 oznaczające wartości wyrazów tego ciągu.

W pierwszym wierszu pliku *wozny.txt* znajduje się jedna liczba  $k$  oznaczająca na ilu piętrach czekają na Profesora sprawy do załatwienia. Liczba  $k$  ma wartość z zakresu 1..15. Z kolei w kolejnych  $k$  wierszach mamy najpierw numer piętra, na którym jest sprawa do załatwienia (numer jest liczbą od 1 do 100, a numery pięter są zapisane w pliku *wozny.txt* w porządku rosnącym), a po odstępnie spacji litera Z albo P. Litera P oznacza, że sprawa do załatwienia z danego piętra jest pilna, a Z, że sprawa jest zwykła.

### Dane wyjściowe:

Plik tekstowy *dokumenty.txt*, w którego jedynym wierszu zapisano dwulitrową kombinację oznaczającą wg jakiego algorytmu powinien w danym dniu wjechać i potem zjechać z gabinetu profesor (tzn. który ciąg wybierze do generowania algorytmu wjazdu, a który zjazdu), aby uwzględniając opisane w treści zadania reguły załatwiania przez niego spraw zabrał tych spraw do domu najmniej. Szczegóły dotyczące oznaczania kombinacji opisano w treści zadania. Po



spacji w tym jedynym wierszu pliku *dokumenty.txt* powinna się znaleźć najmniejsza liczba spraw jaką zabierze do załatwienia do domu profesor RAM wybierając podaną w pliku kombinację algorytmów wjazdu i zjazdu do gabinetu. W przypadku, gdy więcej niż jedna kombinacja gwarantuje uzyskanie najmniejszej liczby spraw do załatwienia w domu należy wskazać dowolną z nich.

### Przykład

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

W  $2*n+3$  7

D 6 10 15 20 34 50 96

oraz w pliku *wozny.txt*

6

15 P

17 Z

41 P

67 Z

91 Z

96 Z

plik *dokumenty.txt* powinien mieć postać:

DW 0

**Objaśnienie do przykładu:** przy wjeździe z wykorzystaniem ciągów wyrazu dowolnego 10,15,20,34,50 oraz 96 profesor zabierze po drodze dokumenty sprawy pilnej(15 piętro) oraz zwykłej (piętro 96). Zjeżdżając , gdy wykorzysta formułę  $2*n+3$  oraz uwzględni, że sprawę pilną musi załatwić otrzymamy następujące piętra 96,94,92, 90, 88, 86, 84, 15. Ponieważ ponownie zatrzyma się na piętrze 96 więc zgodnie z regułą e) załatwi i tę sprawę. Do domu nie będzie musiał zabierać zatem żadnej sprawy co jest najlepszym możliwym wynikiem, gdyż



przy obowiązywaniu każdej z pozostałych 3 kombinacji profesor zabrałby do domu albo 3 sprawy, albo 2.

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

**10 punktów**

**Razem w całym zestawie zadań 50 punktów**