

**Konkurs dla uczniów klas VII i VIII szkoły podstawowej oraz gimnazjów województwa pomorskiego w roku szkolnym 2018/2019**

**Etap II – powiatowy**

**Przedmiot: INFORMATYKA**

Liczba punktów do uzyskania: **50**

**Instrukcja dla rozwiązującego**

1. Zwróć uwagę, aby pliki zawierające rozwiązania poszczególnych zadań miały zawartość i nazwy takie jak określono w treści zadań. Nie przesyłaj innych plików niż te określone w treści zadań.
2. Pliki z rozwiązaniami umieszczasz poprzez formularz zamieszczony na platformie Zdolni z Pomorza lub stronie internetowej Organizatora Pomorskiej Ligi Zadaniowej. Szczegóły techniczne związane z tą czynnością będą podane osobno. Tam również otrzymasz instrukcję jakie informacje dodatkowe, oprócz plików z rozwiązaniami powinno się przekazać.
3. Przy rozwiązywaniu zadań powinno się wykorzystywać te środowiska i narzędzia programistyczne, którymi posługujesz się w szkole lub w domu. W szczególności dopuszcza się następujące środowiska:
  - a) systemy operacyjne – zarówno z grupy Windows, jak i dystrybucje systemu Linux;
  - b) pakiety oprogramowania biurowego- Microsoft Office, ale również wersje otwarte np. Libre Office;
  - c) języki programowania – C/C++, C#, Free Pascal, Java (kompilatory adekwatne do używanych środowisk systemu operacyjnego : DEV, GCC, G++ itd.);
  - d) wizualne środowiska programowania – Scratch, Logomocja lub inne mutacje LOGO;
  - e) nie określa się szczegółowo numerów wersji używanego oprogramowania, aby uczeń mógł je elastycznie dostosować do używanych w szkole.
4. Programy nie powinny zajmować się testowaniem poprawności danych. Zakłada się, że dane są poprawne.
5. Zadania rozwiązujemy samodzielnie.

***Powodzenia !***

### Zadanie 1. (0 – 12pkt)

Grupa uczniów otrzymała zadanie opracowania krótkiego **Testu** dla swoich rówieśników. **Test** dotyczyć ma znajomości podstawowych zagadnień społeczno-politycznych i ma być przeprowadzony w wersji elektronicznej. Uczniowie zdecydowali się przeprowadzić **Test** przy pomocy arkusza kalkulacyjnego, który choć nie jest typowym narzędziem do tworzenia formularzy testowych, ale jednak posiada niezbędne komponenty, by taki **Test** przygotować. Ponadto arkusz będzie bardzo pomocny przy opracowaniu wyników **Testu**. Twoje zadanie polega na tym, aby pomóc uczniom w dwóch fazach realizacji tego **Testu**: przygotowaniu formularza testowego oraz opracowaniu statystycznym wyników **Testu** dla pewnej grupy osób.

- a) W arkuszu należy wykorzystać trzy **Pytania** testowe (podane poniżej) oraz jedno **Dodatkowe** związane z płcią odpowiadającego. **Test** jest anonimowy i uczniowie nie ujawniają poza płcią innych swoich danych.

#### Dane do Testu:

*Pytanie 1.* „**Jak brzmi nazwisko aktualnego prezydenta USA?**”

Pytanie należy umieścić w arkuszu, a w jego otoczeniu (obok , pod spodem itp.) przygotować wyróżniającą ją komórkę (np. kolorem), do której odpowiadający winien wpisać odpowiedź.

*Pytanie 2.* „**Czy Mołdawia jest członkiem NATO?**”

Pytanie należy umieścić w arkuszu, a w jego otoczeniu przygotować tzw. pole wyboru (**Checkbox**). Wybranie tego pola (zaznaczenie) musi oznaczać odpowiedź twierdząco na zadane pytanie, zaś brak wyboru oznaczać odpowiedź negatywną.

*Pytanie 3.* „**Jaka metoda rozdziału mandatów jest stosowana w polskich wyborach parlamentarnych?**”

Pytanie należy umieścić w arkuszu, a w jego otoczeniu skonstruować tzw. pole kombi (lista z rozwijanymi wariantami wyboru) i na niej umieścić trzy warianty odpowiedzi;

*Odpowiedź 1.* „**Metoda Sainte-Lague**”

*Odpowiedź 2.* „**Metoda Hareya-Niemeyera**”

*Odpowiedź 3.* „**Metoda d’Hondta**”

**Odpowiadający ma wybrać z tej, listy jego zdaniem prawidłową odpowiedź!**

Pytanie **Dodatkowe** – dotyczące płci należy skonstruować również przy pomocy pola kombi, tak jak w przypadku **Pytania 3**. Pytanie ma mieć treść „**Wskaż swoją płć**”. Powinno być ono umieszczone poniżej trzech pytań testowych, a w jego otoczeniu należy umieścić pole kombi oraz umieścić dwa warianty odpowiedzi:

*Odpowiedź 1. „Dziewczyna”*

*Odpowiedź 2. „Chłopak”*

**Test jest punktowany następująco:**

błędna odpowiedź na dowolne pytanie: **0 punktów;**

poprawna odpowiedź na *Pytanie 1* lub *Pytanie 2*: **1 punkt;**

poprawna odpowiedź na *Pytanie 3*: **2 punkty.**

**Opracowanie statystycznych wyników Testu.**

W tym samym arkuszu, ale w innym miejscu niż **Test** należy umieścić specjalny wiersz, który będzie służył do automatycznego „zbierania” tzw. **Wiersz Zbierający** odpowiedzi udzielone przez aktualnie odpowiadającego. W tym wierszu, w kolejnych komórkach powinny znaleźć się następujące informacje:

- litera **D**, jeśli odpowiadała **Dziewczyna**;
- litera **C**, jeśli odpowiadał **Chłopak**.

Dane w tym wierszu powinny się aktualizować automatycznie, po każdej zmianie odpowiedzi (czyli gdy odpowiada kolejna osoba).

**Poprawne odpowiedzi do Testu to:**

*Pytanie 1. „Trump”*

*Pytanie 2. „Nie”*

*Pytanie 3. Odpowiedź 3. „Metoda d’Hondta”*

**Przykład:** Koleżanka układających **Test** odpowiedziała na pytania następująco:

*Pytanie 1. „Trump”*

*Pytanie 2. „Nie”*

**Pytanie 3. „Metoda Sainte Lague”**

Opisany wyżej **Wiersz Zbierający** odpowiedzi jest w tym przypadku postaci:

**D 1 1 0 2** (odstępy wskazują na kolejne komórki w tym wierszu).

**Uwaga ! Dla uniknięcia problemów związanych z wielkością liter przyjmij, że poprawna odpowiedź na Pytanie 1 jest zawsze udzielana w postaci „Trump” i odpowiedź „trump” oraz każda inna błędna pisownia jest niepoprawna!**

- b) Utwórz drugi arkusz o nazwie **Statystyka**, w którym znajdą się wyniki zbiorcze. Pomiń etap, w którym uczniowie zajmują się przesyłaniem wyników **Testu** (czyli opisanego na koniec punktu **a Wiersza Zbierającego**) do arkusza **Statystyka**, w którym znajdują się wyniki zbiorcze. Szczegółową organizacją arkusza **Statystyka** zajmiesz się w tej części zadania.

W arkuszu **Statystyka** należy przewidzieć miejsce na piętnaście wierszy takiej postaci jak **Wiersz Zbierający**. Dane mają być przetwarzane w „porcjach” zawierających co najwyżej 15 odpowiedzi osób wypełniających **Test** (maksymalnie 15 **Wierszy Zbierających**). W każdej chwili można jednak będzie, przetworzyć również ich mniejszą ilość, większej jednak nie.

W arkuszu **Statystyka** należy przewidzieć obszar, w którym w każdym momencie można będzie uzyskać odpowiedzi na następujące pytania:

- średni wynik testu;
- średni wynik testu osobno podany dla **Dziewczyna**, a osobno dla **Chłopak**;
- suma punktów uzyskanych z całego testu osobno podany dla **Dziewczyna**, a osobno dla **Chłopak**;
- **wykres** ilustrujący liczbę punktów, uzyskaną osobno dla **Dziewczyna**, a osobno dla **Chłopak** dla poszczególnych pytań (*Pytanie 1, Pytanie 2, Pytanie 3*);
- suma punktów uzyskanych z całego testu osobno podany dla **Dziewczyna**, a osobno dla **Chłopak** z rozbiciem na:
  - **rozkład** łącznej liczby punktów uzyskanych przez rozwiązujących tzn. informację: ile osób uzyskało 0 punktów, ile 1 punkt ile 2 punkty, ile 3 punkty, a ile 4 punkty;
  - **numer najtrudniejszego** pytania czyli w osobnej komórce ma pojawić się tekst *Pytanie 1, Pytanie 2 lub Pytanie 3*. Za najtrudniejsze pytanie uznamy takie, dla którego procent uzyskanych punktów jest najmniejszy (w stosunku do maksymalnie możliwej do uzyskania liczby punktów w danym momencie, czyli dla danej liczby wyników, które posiadamy - np. jeśli w danej chwili mamy w arkuszu **Statystyka** wyniki 8 osób, to maksymalnie do zdobycia za poprawne

odpowiedzi na *Pytania 1* oraz *Pytanie 2* byłoby po 8 punktów, a maksymalnie do zdobycia za poprawną odpowiedź na *Pytanie 3* 16 punktów). Jeżeli w danym momencie mamy więcej niż jedną odpowiedź na pytanie **numer najtrudniejszego** to wystarczy wskazać dowolną z nich;

Wszystkie powyższe wyniki powinny się automatycznie aktualizować pod wpływem każdej zmiany w obszarze, w którym gromadzimy wyniki uzyskane przez rozwiązujących (dopisanie lub usunięcie **Wiersza Zbierającego**, albo korekta wyników w utworzonym już **Wierszu Zbierającym**).

Do testów pozwalających stwierdzić poprawność przygotowanego w tym punkcie arkusza, można sobie naturalnie tworzyć fikcyjne **Wiersze Zbierające** z wynikami, w formacie takim jak opisano w części *a* zadania (maksymalnie 15 wierszy). Do odpowiedzi oddajemy jednak arkusz **Statystyka** z pustymi **Wierszami Zbierającymi**, tak by sprawdzający mógł testować poprawność zastosowanych konstrukcji dla przygotowanych przez siebie danych.

**Uwaga ! W obu częściach zadania, oprócz tych elementów, które powinny się znaleźć w arkuszach, zgodnie z przedstawionym opisem możesz w innych miejscach obu arkuszy (tego z punktu *a* i tego z punktu *b*) umieszczać w różnych komórkach pomocnicze formuły i konstrukcje pomagające Ci w rozwiązaniu zadania.**

**Do oceny oddajesz plik (spakowany) zawierający komputerową realizację konstrukcji oraz obliczeń, na podstawie których uzyskałeś rozwiązanie zadania. Nazwa tego pliku to *Zadanie1*. Powinien on składać się z dwóch arkuszy o nazwach TEST i STATYSTYKA. Zawartość arkusza TEST powinna zawierać rozwiązanie części *a* zadania, zaś zawartości części STATYSTYKA rozwiązanie części *b* zadania.**

## Zadanie 2. (0 – 8pkt)

Z okazji 100 rocznicy odzyskania niepodległości przez Polskę w roku 2018 poproszono Cię o przygotowanie montażu o charakterze graficzno-multimedialnym. Montaż powinien składać się z 4 niezależnych części, których opis podano poniżej. Zakłada się, że wybór odpowiedniej części, którą oglądający ma ochotę obejrzeć odbywa się przy pomocy **Menu** postaci:

- 1 – **Flaga i życzenia.**
- 2 – **Krajobraz Polski.**
- 3 – **Dla cudzoziemca.**
- 4 – **Coś dla ciała czyli polska kuchnia.**
- 5 – **Koniec.**

Sposób realizacji **Menu** pozostawia się autorowi. Może to być dowolne rozwiązanie pozwalające z tego **Menu** przechodzić do kolejnych części montażu.

Opis poszczególnych części montażu:

- a) **Flaga i życzenia** - w tej opcji można będzie zobaczyć obraz flagi polskiej (bez orła), ale w wersji, w której jest rozwiana na wietrze, niejako „pofałdowana” – nie zwykły prostokąt podzielony na dwie części w barwach czerwonej i białej. Dodatkowo flaga powinna być takiej wielkości, aby na niej odpowiednio dobraną co do wielkości i koloru dowolną czcionką zostały zredagowane krótkie, Twoje własne życzenia dla naszej Ojczyzny na jej setne urodziny. Życzenia winny zawierać minimum 50 znaków, być dobrze widoczne i w całości mieścić się na faldzie.
- b) **Krajobraz Polski** – w tej części należy zmontować widokówkę złożoną z 4 mniejszych obrazów rozmieszczonych mniej więcej w 4 rogach głównej widokówki. Każda z mniejszych widokówek ma prezentować wybrane przez Ciebie miejsce naszego kraju. Przy czym na każdej z „małych” widokówek musi to być inne miejsce i inna pora roku, w której dane miejsce jest prezentowane (wiosna, lato, jesień, zima). Dodatkowo na każdej z mniejszych widokówek umieść dyskretny, identyczny, charakterystyczny element Twojego autorstwa. Może to być np. mały ludzik podziwiający dane miejsce o danej porze roku, ew. sympatyczny zwierzak itp. Pod każdą małą widokówką należy umieścić podpis np. „Gdynia latem”, „Zakopane zimą” itd. W środku głównej widokówki umieść sporządzony dowolną czcionką tekst „**A TO POLSKA WŁAŚNIE**”. Szczegóły techniczne związane z wielkością widokówek oraz np. tłem, kolorystyką, czy też czcionką

głównej widokówki są dowolne. Ważne, aby całość była wykonana zgodnie z podanym opisem, była czytelna, a różne elementy składowe nie przysłaniały innych. Uwaga ! Niektóre pory roku są dość w naszym kraju podobne, jeśli oglądać je na zdjęciu. O ile nie będzie tu rażących sytuacji (drzewa bez liści prezentowane jako lato) te subtelności będą uwzględniane na korzyść autora.

- c) **Dla cudzoziemca** – w tej opcji Twoje zadanie polega na przygotowaniu krótkiego syntetycznego opisu naszego kraju (na wzór notki encyklopedycznej), nie krótszego niż 1000 znaków, ale nie przekraczającego 1500 znaków w języku angielskim, w dowolnym edytorze tekstu.
- d) **Coś dla ciała czyli polska kuchnia** – w tej opcji Twoje zadanie polega na wybraniu dwóch dowolnych dań charakterystycznych dla naszej kuchni oraz ich zaprezentowaniu w formie prezentacji multimedialnej, zrealizowanej przy pomocy dostępnych narzędzi.
- e) **Koniec** – opcja pozwala zakończyć oglądanie montażu.

#### Uwagi:

1. Narzędzia informatyczne niezbędne do realizacji tego zadania (edytory grafiki, programy do tworzenia prezentacji, itp.) autor dobiera samodzielnie, ale wskazane by były to narzędzia na tyle ogólnodostępne, by nie stwarzały problemu przy ocenie zadania.
2. Przy ocenie zadania ważna będzie techniczna (informatyczna) jakość jego wykonania (między innymi wypełnienie określonych wyżej wymogów szczegółowych). Część merytoryczna, też jest jednak istotna choć na pewno ma mniejsze znaczenie tzn. błędy np. językowe związane z opcją 3, albo nieścisłości w charakterystyce kulinarnej dań wybranych do opcji 4.

**Do oceny oddajesz plik pozwalający po uruchomieniu wybierać z Menu jedną z czterech opisanych w zadaniu opcji pod nazwą Zadanie2-menu, a także wszystkie pliki, które pozwalają w pełni ocenić działanie wszystkich opcji.**

### Zadanie 3. (0 – 8pkt)

Informatyka to świat kojarzony z zerami i jedynekami. W związku z tym Twoim zadaniem będzie napisanie programu, który dla zadanego parametru  $n$  ( $n$  liczba naturalna większa od 2, a mniejsza lub równą 22) wypisze na ekranie opisaną poniżej kompozycję złożoną właśnie z zer i jedynek.

Kompozycja składa się z dwóch symetrycznych w stosunku do siebie obszarów, a liczba jej wierszy jest uzależniona od wartości  $n$  (wartość  $n$  winna być odczytywana z pliku o nazwie *dane.txt*).

W pierwszym rzędzie pierwszego obszaru powinien być wypisany rząd zer i jedynek (na przemian zero i jedynek oddzielone odstępem jednej spacji) o długości  $n$ . Przy czym to czy rząd rozpoczyna się zerem czy jedyneką należy ustalić losowo. Każdy kolejny rząd zer i jedynek jest o 2 znaki zero lub jeden krótszy od poprzedniego, a dokładniej skrócenie długości rzędu polega na tym, że „zabieramy” w stosunku do poprzedniego rzędu znaki pierwszy oraz ostatni (oraz odstęp za i przed znakiem, który usunęliśmy). Dodatkowo obowiązuje zasada, że w tym pierwszym obszarze całej kompozycji w rzędach o numerze parzystym zero wypisujemy w tej kolumnie, gdzie w poprzednim rzędzie była cyfra 1, a jedynek w tym miejscu, gdzie w poprzednim rzędzie w tej samej kolumnie była cyfra zero. Z kolei w rzędach o numerze nieparzystym (oprócz pierwszego, który tworzony jest automatycznie po wylosowaniu początkowego zera lub jedynek) zero wypisujemy w tym miejscu, gdzie w poprzednim rzędzie w tej samej kolumnie była także cyfra 0, a jedynek w tym miejscu, gdzie w poprzednim rzędzie w tej samej kolumnie była również cyfra 1. Pierwszy obszar kompozycji kończy się rzędem o długości 2 (gdy  $n$  było parzyste) lub rzędem o długości 1, gdy  $n$  było nieparzyste.

Drugi obszar kompozycji jest symetryczny (względem osi poziomej) w stosunku do rzędu pierwszego. Dokładniej zaraz po ostatnim rzędzie obszaru 1 pojawia się rozpoczynający obszar drugi rząd o takiej samej długości jak jego poprzednik (czyli 2 albo 1 zależnie od tego czy  $n$  było parzyste czy nieparzyste). Każdy kolejny rząd obszaru drugiego jest o dwa znaki zero lub jeden dłuższy od rzędu poprzedniego (i oczywiście o dodatkowe odstępy oddzielający znaki 0 i 1). Tym razem dopisujemy po jednym znaku 0 lub 1 na początku i końcu każdego nowego rzędu. Ostatni rząd w drugim obszarze to rząd, który będzie miał znowu długość  $n$ . O rozmieszczeniu zer i jedynek w drugim obszarze decyduje jedna ogólna zasada – w obszarze drugim w miejscu położonym symetrycznie względem wspomnianej osi poziomej w stosunku do odpowiedniego miejsca obszaru pierwszego winna być cyfra inna niż ta w obszarze pierwszym tzn. jeżeli w obszarze pierwszym mamy cyfrę 1 w pewnym miejscu to w miejscu położonym w stosunku do niego symetrycznie względem osi poziomej w obszarze drugim



winna być wypisana cyfra 0, zaś jeżeli w obszarze pierwszym mamy cyfrę 0 w pewnym miejscu, to w miejscu położonym w stosunku do niego symetrycznie względem osi poziomej w obszarze drugim winna być wypisana cyfra 1.

Dodatkowo w pliku tekstowym o nazwie *suma.txt* program powinien podać liczbę zer (lub liczbę jedynek – zauważmy, że liczby te są identyczne), która została umieszczona w całej kompozycji.

### Dane wejściowe:

Plik *dane.txt* zawierający w swoim jedynym wierszu jedną liczbę naturalną  $n$  większa od 2 oraz nie większą niż 22.

### Dane wyjściowe:

Wynik wyświetlany na ekranie : w kolejnych liniach pojawiają się ciągi naprzemiennie w nich występujących zer i jedynek, oddzielonych pojedynczym odstępem. Pierwszy i ostatni ciąg ma długość  $n$ . Długości pozostałych wierszy oraz kolejność występowania zer i jedynek w każdym z wierszy winny być zgodne z opisem podanym w zadaniu. Drugim wynikiem działania programu winien być plik *suma.txt* składający się z jednego wiersza, w którym winna się znaleźć liczba zer (lub co na jedno wychodzi jedynek) , które zostały umieszczone na ekranie w całej kompozycji.

### Przykład 1

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

7

To na ekranie pojawić się powinien następujący wydruk (wariant, gdy losowo ustalono, że pierwszy wiersz zaczyna się 1):

1 0 1 0 1 0 1

1 0 1 0 1

0 1 0

0

1

1 0 1

0 1 0 1 0

0 1 0 1 0 1 0

A gdyby pierwszy wiersz zaczynał się (po losowym tego ustaleniu) od 0 to na ekranie powinien być następujący wydruk:

0 1 0 1 0 1 0

0 1 0 1 0

1 0 1

1

0

0 1 0

1 0 1 0 1

1 0 1 0 1 0 1

zaś w pliku *suma.txt* (dla obu wariantów) powinna się znaleźć liczba

16

## Przykład 2

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

4

To na ekranie pojawić się powinien następujący wydruk (wariant, gdy losowo ustalono, że pierwszy wiersz zaczyna się 1):

1 0 1 0

0 1

1 0

0 1 0 1

A gdyby pierwszy wiersz zaczynał się (po losowym tego ustaleniu) od 0 to na ekranie powinien być następujący wydruk:

0 1 0 1

0 1

1 0

1 0 1 0

zaś w pliku *suma.txt* (dla obu wariantów) powinna się znaleźć liczba

6

**Uwaga ! Jedyńki i zera winny być wypisane na ekranie na tyle precyzyjnie, aby nie pozostawiały żadnych wątpliwości, w której kolumnie się znajdują**

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

#### Zadanie 4. (0 – 9pkt)

Algorytm Euklidesa wyznacza największy wspólny dzielnik dwóch liczb naturalnych (ale nie dwóch zer). Funkcja rekurencyjna realizująca ten algorytm może być zapisana następująca:

Funkcja  $NWD(a, b)$

Jeżeli  $b = 0$  to  $wynik = a$

w przeciwnym razie  $wynik = NWD(b, a \bmod b)$

gdzie użyty tu skrót  $mod$  oznacza resztę z dzielenia liczb  $a$  i  $b$ .

Rozwiąż następujące problemy

- w pliku tekstowym o nazwie **Zadanie4a** opisz dokładnie krok po kroku jak realizowane byłoby wywołanie funkcji  $NWD$  dla parametrów  $a = 56$  i  $b = 21$  i podaj wynik końcowy dla tego wywołania
- napisz program, który zawiera wersję funkcji  $NWD$ , nie korzystając z mechanizmu rekurencji oraz przy jej pomocy dla pewnej naturalnej i większej od 1 liczby  $a$  znajdzie największy z jej największych wspólnych dzielników liczonych z kolejnymi liczbami naturalnymi pewnego  $n$ -elementowego ciągu.

#### Dane wejściowe:

Plik **NWD.txt** zawierający w pierwszym wierszu liczbę naturalną  $a$  większą od 1, w drugim wierszu liczbę naturalną  $n$  (większą od zera) określającą liczbę kolejnych wierszy pliku oraz w kolejnych  $n$  liniach dowolne liczby naturalne (większe od zera), po jednej w każdym wierszu.

#### Dane wyjściowe:

Plik **wynik.txt** składający się z jednego wiersza, w którym winna się znaleźć wartość największego z spośród  $n$  największych wspólnych dzielników obliczanych dla par: liczba  $a$ , kolejna liczba ciągu (czyli kolejna z liczb począwszy od 3 wiersza pliku **NWD.txt**).



### Przykład

Jeżeli plik *NWD.txt* zawiera następujące dane:

72

4

3

15

8

5

To w pliku *wynik.txt* powinna się pojawić liczba:

8

### Objaśnienie do przykładu

Ponieważ  $NWD(72, 3) = 3$ ,  $NWD(72, 15) = 3$ ,  $NWD(72, 8) = 8$ , a  $NWD(72, 5) = 1$  więc największą liczbą spośród  $NWD$  liczonych dla liczby 72 i kolejnych liczb ciągu jest liczba 8.

**Kod źródłowy napisanego przez Ciebie programu umieść w pliku *Zadanie4b***

c) Uczniowie pewnej szkoły stają na linii startu, aby wziąć udział w pewnym biegu. Biegacze nie mają do pokonania określonego dystansu, tylko rozpoczynając bieg razem, ze wspólnego startu, biegają po trasie będącej okrężeniem o długości 1 km do czasu, aż znowu wszyscy spotkają się razem w punkcie startu. Wtedy bieg się kończy. Każdy z biegaczy chciałby wiedzieć, ile kilometrów przebiegł w czasie wyścigu. Jednak ponieważ wyścig ten potrafi trwać bardzo długo, zawodnicy często zapominają ze zmęczenia ile dokładnie przebiegli okrążeń, mimo, że każdy z nich przebiega jedno okrążenie w stałym, znanym czasie (zaokrąglonym do pełnej minuty). Napisz program, który dla następujących danych

- liczba zawodników –  $n$
- czasy w minutach, jakie potrzebują kolejni zawodnicy o numerach od 1 do  $n$  do przebycia jednego okrążenia

dla każdego zawodnika obliczy, ile kilometrów przebiegł on w czasie wyścigu oraz poda dodatkowo maksymalną odległość przebytą przez zawodnika, który przebiegł najwięcej kilometrów i sumę kilometrów przebytych przez wszystkich zawodników.

#### Dane wejściowe:

Plik *biegacze.txt* zawierający w pierwszym wierszu liczbę naturalną  $n$  (większą od zera) określającą liczbę biegaczy i zarazem liczbę kolejnych wierszy pliku, zaś w kolejnych  $n$  liniach dowolne, większe od zera liczby naturalne, po jednej w każdym wierszu oznaczające czasy w minutach jaki na pokonanie jednego okrążenia potrzebują kolejni zawodnicy. Dokładniej w drugim wierszu mam czas pokonania okrążenia przez pierwszego zawodnika, w trzecim przez drugiego zawodnika itd., w ostatnim wierszu znajduje się czas pokonania jednego okrążenia przez ostatniego  $n$ -tego zawodnika.

#### Dane wyjściowe:

Plik *rezultaty.txt* składający się  $n + 2$  wierszy. W pierwszych  $n$  wierszach powinna się znaleźć informacja ile kilometrów podczas biegu przebiegli kolejni zawodnicy: w pierwszym wierszu ile kilometrów przebiegł pierwszy zawodnik, w drugim ile kilometrów przebiegł drugi zawodnik itd., w  $n$ -tym wierszu znajduje się informacja ile kilometrów przebiegł  $n$ -ty zawodnik. W wierszu  $n + 1$  powinna się znaleźć maksymalną odległość przebytą przez zawodnika (jeżeli kilku zawodników przebiegło ten sam dystans podajemy maksymalną liczbę tylko jeden raz), zaś w wierszu  $n + 2$  sumę kilometrów przebytych przez wszystkich zawodników

#### Przykład

Jeżeli plik *biegacze.txt* zawiera następujące dane:

5  
4  
3  
6  
9  
8

To w pliku *rezultaty.txt* powinny się pojawić następujące liczby:

18

24

12

8

9

24

71

#### **Objaśnienie do przykładu**

Zawodnicy uwzględniając ich indywidualne czasy, w których przebiegają jedno okrążenie zakończą bieg wspólnie po 72 minutach. Z tego wynikają liczby kilometrów jakie przebiegną kolejni zawodnicy, podane w pierwszych 5 wierszach. Największa z tych liczb to 24, a więc jest podana w szóstym wierszu, a samą przebiegniętych kilometrów przez wszystkich zawodników to 71 więc ją zawiera ostatni wiersz.

**Kod źródłowy napisanego przez Ciebie programu umieść w pliku *Zadanie4c*.**

**Do oceny oddajesz pliki *Zadanie4a*, *Zadanie4b* oraz *Zadanie4***

### Zadanie 5. (0 – 11pkt)

Rozważmy ciąg  $n$  liczb naturalnych. Należy je uporządkować rosnąco ze względu na liczbę różnych cyfr z jakich się składają. W przypadku dwóch lub większej ilości liczb składających się z takiej samej liczby różnych cyfr o kolejności decyduje wartość liczby (mniejsza poprzedza większą).

#### 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 *porzadek.txt* w kolejnych wierszach (po jednej liczbie w wierszu) liczby ciągu umieszczonego poprzednio w pliku *ciag.txt*, ale tym razem w kolejności od liczby, która ma najmniej różnych cyfr, aż do tej, która ma ich najwięcej. W przypadku dwóch lub większej ilości liczb składających się z takiej samej liczby różnych cyfr o kolejności w pliku decyduje wartość liczby (mniejsza poprzedza większą), a jeżeli mają dodatkowo tę samą wartość to po prostu występują w pliku w sąsiadujących wierszach.

#### Przykłady

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

8

2324

32

112376

123

321

11

57

3897



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

11

32

57

123

321

2324

3897

112376

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