Zad. 1. Numer PESEL to jedenastocyfrowy symbol jednoznacznie identyfikujący osobę, w którym:

1. pierwsze sześć cyfr koduje datę urodzin osoby, gdzie:
2. pierwsze dwie cyfry są dwiema ostatnimi cyframi roku urodzenia,
3. trzecia i czwarta cyfra koduje numer miesiąca urodzenia, a dokładniej dla osób urodzonych w latach:

* 1900 – 1999 – numer miesiąca urodzenia,
* 2000 – 2099 – numer miesiąca urodzenia plus 20,
* 2100 – 2199 – numer miesiąca urodzenia plus 40,
* 2200 – 2299 – numer miesiąca urodzenia plus 60;

1. piąta i szósta cyfra jest numerem dnia miesiąca urodzenia danej osoby;
2. kolejne trzy cyfry (siódma, ósma i dziewiąta) tworzą liczbę porządkową – kolejne numery osób urodzonych w tym samym dniu (uwzględniając płeć – patrz podpunkt c)),
3. dziesiąta cyfra definiuje płeć (parzysta kobietę, nieparzysta mężczyznę),
4. ostatnia cyfra jest cyfrą kontrolną (oznaczmy ją przez *c*). Jest ona wyznaczana na podstawie pierwszych dziesięciu cyfr w następujący sposób:

gdzie jest -tą wagą, jest -tą cyfrą numeru PESEL, oznacza resztę z dzielenia liczby *a* przez *b*. Kolejne wagi wynoszą odpowiednio: 1, 3, 7, 9, 1, 3, 7, 9, 1, 3.

Napisz program, który będzie ewidencjonował ludność zapisując każdej ewidencjonowanej osobie jednoznaczny numer PESEL. Podczas pierwszego uruchomienia lista numerów PESEL jest pusta. Po uruchomieniu programu dodany będzie mógł być kolejny wpis (lub wpisy). Program będzie pytał kolejno o:

1. czterocyfrowy rok urodzenia: „Podaj czterocyfrowy rok urodzenia, np. 1993.” Po wpisaniu złego roku program poinformuje: „Podałeś złą formę roku, wciśnij p jeśli chcesz podać ponownie rok lub wciśnij inny klawisz jeśli chcesz zakończyć”. Wciśnięcie p powoduje powrót do początku podpunktu a), inny przejście do podpunktu f);
2. miesiąc: „Podaj miesiąc urodzenia, np. czerwiec albo 6 .” Po wpisaniu złego miesiąca program poinformuje: „Podałeś złą formę miesiąca, wciśnij p jeśli chcesz podać ponownie miesiąc lub wciśnij inny klawisz jeśli chcesz zakończyć”. Wciśnięcie p powoduje powrót do początku podpunktu b), inny przejście do podpunktu f);
3. dzień miesiąca: „Podaj numer dnia miesiąca urodzenia, np. 23.” Po wpisaniu złego numeru program poinformuje: „Podałeś zły numer, wciśnij p jeśli chcesz podać ponownie numer dnia lub wciśnij inny klawisz jeśli chcesz zakończyć”. Wciśnięcie p powoduje powrót do początku podpunktu c), inny przejście do podpunktu f);
4. płeć: „Podaj płeć: wciśnij k jeśli jesteś kobietą albo m jeśli jesteś mężczyzną.” Po wpisaniu złego klawisza program poinformuje: „Podałeś niepoprawną formę płci, wciśnij p jeśli chcesz podać ponownie płeć lub wciśnij inny klawisz jeśli chcesz zakończyć.” Wciśnięcie p powoduje powrót do początku podpunktu d), inny przejście do podpunktu f);
5. weryfikację: „Czy chcesz dokonać wpisu 1993, 6, 23, k? Klawisz t – tak, pozostałe – nie.” Wybór klawisza innego niż t – przechodzimy do podpunktu f).
6. „Czy chcesz dokonać kolejnego wpisu? Klawisz t – tak, pozostałe – nie.” Wybór klawisza innego niż t – koniec działania programu.

Po każdym poprawnym i kompletnym zebraniu danych z podpunktów a)—d) i wyborze klawisza t w podpunkcie e) program dopisuje do listy numerów PESEL (plik pesel.txt) kolejne osoby (w rzeczywistości zbierane są inne dane osobowe jak imię, nazwisko itd. – my nie zajmujemy się tym przypadkiem) przypisując im kolejne liczby porządkowe (cyfry siódma, ósma i dziewiąta) i wyznaczając cyfrę kontrolną (dziesiąta cyfra).

Śledząc poprawne wpisy z podpunktów a)—d), tzn. sczytując kolejno: 1993; czerwiec; 23; k wpis (jeśli byłby on pierwszym dla osoby tej płci urodzonej w tym dniu) byłby postaci: 93062300007.

Lista powinna być posortowana od osób najstarszych do najmłodszych, w przypadku osób o tym samym wieku po kolejnych wpisach, a w przypadku tego samego wpisu po płci (najpierw kobiety).

Zad. 2. Często rozwiązywanie układów równań liniowych (szczególnie w zadaniach stawianych przez życie) sprawiać może spore kłopoty i metody szkolne wówczas zawodzą (powodem może być rozmiar układu lub to, że ma on inną ilość równań i niewiadomych).

Jeśli założymy, że układ równań liniowych przedstawiony jest w postaci: , gdzie jest dwuwskaźnikową tablicą współczynników, jest tablicą niewiadomych, a tablicą wyrazów wolnych, to taki układ równań rozwiązać można pewną metodą przybliżoną (nie daje ona gwarancji otrzymania dokładnego wyniku, ale gwarantuje to, że wykonując odpowiednio dużo kroków będziemy w stanie zbliżyć się do rozwiązania dokładnego dowolnie blisko). Algorytm jednej z takich metod przedstawić możemy w następujących krokach:

1. wybór rozwiązania startowego (jego długość jest oczywiście równa długości tablicy ),
2. kolejne rozwiązania wyznacza się z następującej zależności:

gdzie: jest -tym wierszem współczynników tablicy , jest -tym elementem tablicy , oznacza iloczyn skalarny wektorów i , tzn. jeśli obydwa wektory mają długość , to

oznacza sumę kwadratów elementów wektora , , gdzie oznacza resztę z dzielenia liczby przez liczbę natomiast liczba oznacza ilość wierszy dwuwymiarowej tablicy . Mnożenie wektora przez liczbę definiujemy tradycyjnie, tzn. w wyniku otrzymujemy wektor, którego każdy element pomnożony został przez daną liczbę.

Np. dla układu równań:

dwuwskaźnikowa tablica *A* ma postać: i analogicznie dalsze elementy algorytmu: , , , , .

Napisz program, który rozwiązywał będzie zadany układ równań liniowych. Na wejściu podajemy: dwuwymiarową tablicę współczynników (*tablicaA.txt*), tablicę wyrazów wolnych (*tablicaB.txt*), tablicę (rozwiązanie) startowe (*tablicax0.txt*) oraz dokładność (o dokładność tę program będzie pytał). Dokładność ta jest niezbędna do zatrzymania algorytmu (jest to metoda przybliżona). Dokładniej algorytm pracował będzie dopóki otrzymane rozwiązanie nie zbliży się do dokładnego o mniej niż . Weryfikacja warunku zatrzymania odbywa się na podstawie formuły:

gdzie , jest wektorem (jednowymiarową tablicą), którego każdy -ty element otrzymujemy przez działanie , gdzie jest otrzymanym rozwiązaniem.

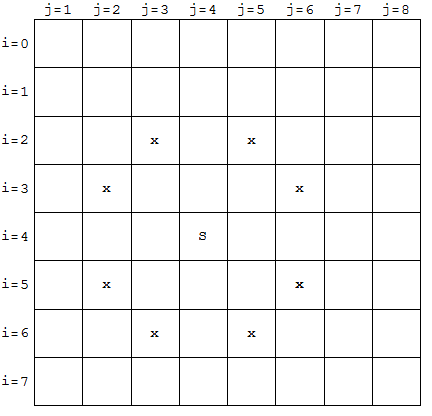
Program ma zwracać do pliku *tablicarozw.txt* rozwiązanie (pierwsze, które pozytywnie przeszło weryfikację).

Przykład. Podając , , (patrz odpowiednie pliki .txt) i otrzymujemy rozwiązanie: (rozwiązaniem dokładnym jest ).

Zad. 3. Napisz program weryfikujący, czy diagram sudoku został poprawnie wypełniony. Rozpatrz możliwie dużo różnych przypadków możliwości popełnienia błędów i różne rozmiary diagramów. Diagram sudoku jest dwuwymiarową tablicą zapisaną w pliku. Program po zapytaniu o nazwę pliku przeprowadzi weryfikację i poda odpowiedź. Np. dla pliku sudoku1.txt odpowiedź brzmi: *poprawnie,*  a dla pliku sudoku2.txt odpowiedź brzmi: *niepoprawnie*.

Zad. 4 Szachownica ma 64 pola. Niech jej pola będą definiowane jak elementy dwuwskaźnikowej tablicy. Na dole były początkowo ustawione bierki czarne, a na górze białe. Napisz program, który po zapytaniu o kolor (b – kolor biały, c – kolor czarny), pole szachownicy () i o bierkę (w — wieża, g – goniec, s – skoczek, k – król, h – hetman lub p – pion) zapisze do pliku szchy.txt położenie wybranej bierki i wszystkie dopuszczalne położenia tej bierki po wykonaniu ruchu (na szachownicy jest tylko wybrana bierka). Rozwiązanie ma być przedstawione graficznie – za pomocą znaków dostępnych z klawiatury narysowana na być szachownica, w wybranym jej miejscu umieszczone bierka (za pomocą odpowiedniej litery – wielka litera, to bierka biała, mała, to bierka czarna) natomiast możliwe ruch zaznaczone literą x.

Przykład. Jeżeli wybierzemy kolejno: b; 5; 4; s, to otrzymamy rozwiązanie jak na rysunku (przykład pokazuje rozmieszczenie bierek, nie należy zwracać uwagi na grafikę, którą pozostawiamy w gestii rozwiązującego):



Zad. 5. Na samym dole koła rowerowego o danym promieniu *R* siedzi mucha. Rower przejechał *n* metrów. Napisz program obliczający (po zapytaniu o *R* i *n*) długość drogi, jaką przebyła w tym czasie mucha (zakładamy, że poruszała się ona na kole nie zmieniając na nim swej pozycji).

Uwaga 1: Można skorzystać z równania cykloidy.

Uwaga 2: Domyślamy się, że rozwiązujący nie znają całek oznaczonych, więc rozwiązania korzystające wprost z tego pojęcia będą niemile widziane.

Wskazówka 1: Z równania cykloidy można uzyskać zależność: przejechane metry – położenie muchy.

Wskazówka 2: Długość krzywej można przybliżać długością łamanej, której wierzchołki znajdują się na tej krzywej (oczywiście im więcej wierzchołków, tym lepiej). Kierując się tą wskazówką, należy uwzględnić ilość wierzchołków jako argument programu.

Uwaga 3: Do rozwiązania (program) należy dołączyć opis rozwiązania problemu.

Na poniższym rysunku znajduje się ilustracja zadania (szarym kołem zaznaczone są początkowa, przejściowe i końcowa pozycja muchy, a pogrubiony niebieski łuk jest drogą przebyłą przez muchę).

