
Zajęcia 3 – tablice statyczne i łańcuchy

1. Napisz program, który:

- utworzy tablicę 10 liczb rzeczywistych i wypełni ją wartościami losowymi z przedziału $[3.0, \dots, 9.0)$,
- wyznaczy średnią arytmetyczną elementów tablicy,
- wyznaczy ile elementów jest mniejszych, ile większych od średniej.

Na ekranie należy wypisać wylosowane liczby w oryginalnej oraz odwrotnej kolejności, a także wszystkie wyznaczone wartości.

Przykład:

```
6.0 6.4 8.3 7.3 7.3 6.2 3.5 6.4 7.7 3.4
3.4 7.7 6.4 3.5 6.2 7.3 7.3 8.3 6.4 6.0
Średnia: 6.3
Większych: 6
Mniejszych: 4
```

2. Napisz program, który wczytuje od użytkownika liczbę całkowitą, a następnie wyświetla jej reprezentację w kodzie binarnym (ZM). Podczas konwersji liczby należy cyfry odpowiadające kolejnym jej bitom zapisywać w pomocniczej tablicy znaków. Maksymalna liczba cyfr powinna zależeć od rozmiaru zmiennej typu `int`, np. dla 4 bajtów jest to 32. Konwersji należy dokonać korzystając z operacji dzielenia całkowitego oraz operacji modulo.

Przykład:

Wejście:

-75 (liczba podana przez użytkownika)

Wynik:

Liczba -75 binarnie: 1.1001011

3. Napisz program, który:

- stworzy tablicę (macierz) 5×5 liczb całkowitych, a następnie wypełnij ją losowymi wartościami ze zbioru $\{-5, -4, \dots, 5\}$,
- dla każdej kolumny wyznacz minimum,

- wyznaczy największą wartość w komórkach leżących na przekątnych macierzy (po jednej wartości dla każdej z dwóch przekątnych).

Program ma wyświetlać tablicę wypełnioną liczbami oraz tablice z minimami oraz maksimami.

4. Napisz program, który wczytuje od użytkownika tekst znak po znaku aż do naciśnięcia sekwencji CTRL+Z oznaczającej koniec ciągu (EOF). Program powinien wyświetlić:
 - liczbę białych znaków w tekście,
 - liczbę liter w tekście,
 - liczbę samogłosek w tekście (tylko z alfabetu angielskiego),
 - sumę wszystkich cyfr w podanym tekście.

Przykład:

Podaj tekst, aby zakończyć naciśnij CTRL+Z (windows) lub CTRL+D (linux):

Ala ma 15 rybek
i 2 koty.

Białe znaki: 6
Literey: 15
Samogłoski: 8
Suma cyfr: 8

5. Napisz program, który pobiera od użytkownika łańcuch znaków o maks. długości 40 znaków i sprawdza, czy jest on palindromem. W przypadku gdy podany tekst nie jest palindromem program powinien wyświetlić minimalną liczbę zmian znaków potrzebną, by zmienić podany ciąg w palindrom.

Przykład 1.:

Podaj tekst: robot
Najmniejsza liczba zmian liter konieczna, by przekształcić słowo "robot" w palindrom wynosi: 1

Przykład 2.:

Podaj tekst: Alaala
Podany ciąg jest palindromem.

6. Napisz program, który sprawdza, czy zadana dwuwymiarowa tablica 9x9 liczb całkowitych ze zbioru $\{1, 2, \dots, 9\}$ jest poprawnym rozwiązaniem łamigłówki Sudoku. Rozwiązanie jest poprawne, jeżeli:

- w każdym wierszu tabeli każda z cyfr $\{1, 2, \dots, 9\}$ jest wpisana dokładnie raz,
- w każdej kolumnie tabeli każda z cyfr $\{1, 2, \dots, 9\}$ jest wpisana dokładnie raz,
- w każdym z 9 kwadratów 3×3 każda z cyfr $\{1, 2, \dots, 9\}$ jest wpisana dokładnie raz.

Program powinien wyświetlać informację o tym, czy rozwiązanie jest poprawne, a w przeciwnym razie numery wierszy, kolumn oraz współrzędne kwadratów 3×3 , w których napotkano konfigurację sprzeczną z zasadami.

Przykład 1. Poniższa konfiguracja przedstawia *poprawne* rozwiązanie:

```
int sudoku1[][N] = {
    {1, 2, 3, 4, 5, 6, 7, 8, 9},
    {4, 5, 6, 7, 8, 9, 1, 2, 3},
    {7, 8, 9, 1, 2, 3, 4, 5, 6},
    {2, 1, 4, 3, 6, 5, 8, 9, 7},
    {3, 6, 5, 8, 9, 7, 2, 1, 4},
    {8, 9, 7, 2, 1, 4, 3, 6, 5},
    {5, 3, 1, 6, 4, 2, 9, 7, 8},
    {6, 4, 2, 9, 7, 8, 5, 3, 1},
    {9, 7, 8, 5, 3, 1, 6, 4, 2}
};
```

Przykład 2. Poniższa konfiguracja przedstawia *niepoprawne* rozwiązanie:

```
int sudoku2[][N] = {
    {1, 2, 2, 4, 5, 6, 7, 8, 9},
    {4, 5, 6, 7, 8, 9, 1, 2, 3},
    {7, 8, 9, 1, 2, 3, 4, 5, 6},
    {2, 1, 4, 3, 6, 5, 8, 9, 7},
    {3, 6, 5, 8, 9, 7, 2, 1, 4},
    {8, 9, 7, 2, 1, 4, 3, 6, 5},
    {5, 3, 1, 4, 4, 2, 9, 7, 8},
    {6, 4, 2, 9, 7, 8, 5, 3, 1},
    {9, 7, 8, 5, 3, 1, 6, 4, 2}
};
```

Wydruk wygenerowany przez program:

```
Row 1 is not ok
Row 7 is not ok
Column 3 is not ok
Column 4 is not ok
Square (1, 1) is not ok
Square (3, 2) is not ok
```