
Zajęcia 3 – tablice statyczne i łańcuchy

1. Wartości pomiaru temperatury powietrza w 20 stacjach pomiarowych na terenie Polski w dniu 16 marca 2022 r. o godz. 19:00 wyniosły:

4.2, 5.7, 3.3, 7.7, -1, 1.8, 6.6, 1.5, 2.2, 7.9, -6.1, 4.3, 6.3, 3.3, 3.2, 7, 2.9, 4.2, 4.8, 4.7

Napisz program, który na podstawie podanych wartości:

- wyznaczy i wydrukuje na konsoli średnią wartość temperatury;
- wyznaczy liczbę miejsc, w których temp. była wyższa od średniej;
- wyznaczy i wypisze odchylenie standardowe temperatury wg wzoru: $\sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N}}$, gdzie N to liczba pomiarów, a \bar{x} to średnia;
- wypisze temp. minimalną;
- wydrukuje temperatury w kolejności malejącej.

Srednia temp.: 3.725

Odch. standardowe: 3.13686

11 miejsc z temp. wyzsza od sredniej

Temp. minimalna: -6.1

Temp. malejaco:

7.9	7.7	7	6.6	6.3
5.7	4.8	4.7	4.3	4.2
4.2	3.3	3.3	3.2	2.9
2.2	1.8	1.5	-1	-6.1

Wskazówki:

- funkcja `std::sqrt` wymaga nagłówka `<cmath>`
- wart. temp. można posortować za pomocą `std::sort(temp, temp + 20)`, funkcja `sort` zdefiniowana jest w nagłówku `<algorithm>`
- wydruk liczb na określonej liczbie miejsc można uzyskać np. za pomocą `std::cout << std::setw(5) << x`, które uzupełni wydruk wart. x , tak aby zajmował co najmniej 5 znaków (nagłówek `<iomanip>`).

2. Napisz program, który utworzy tablicę dwuwymiarową 4×4 liczb całkowitych i wypełni ją podanymi wartościami, a następnie:

- wydrukuje tablicę;
- wypisze sumy wartości w każdym wierszu;
- wypisze sumy wartości w każdej kolumnie;
- wypisze sumy wartości dla przekątnych, tj. z lewa na prawo i z prawa na lewo;
- sprawdzi czy podana macierz liczb tworzy tzw. *magiczny kwadrat*, tj. taki w którym powyższe sumy są identyczne.
- *czy następujący ciąg liczb tworzy magiczny kwadrat 5×5 : 17 24 1 8 15
23 5 7 14 16 4 6 13 20 22 10 12 19 21 3 11 18 25 2 9

Przykład 1. Dla tablicy wypełnionej kolejno wartościami: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 17 program powinien wydrukować:

```

1  2  3  4
5  6  7  8
9 10 11 12
13 14 15 17
Sumy dla wierszy: 10 26 42 59
Sumy dla kolumn: 28 32 36 41
Suma dla pierwszej przekatnej: 35
Suma dla drugiej przekatnej: 34
Kwadrat magiczny: nie
```

Przykład 2. Dla tablicy wypełnionej kolejno wartościami: 1 14 14 4 11 7 6 9 8 10 10 5 13 2 3 15 program powinien wydrukować:

```

1  14  14  4
11  7  6  9
8  10  10  5
13  2  3  15
Sumy dla wierszy: 33 33 33 33
Sumy dla kolumn: 33 33 33 33
Suma dla pierwszej przekatnej: 33
Suma dla drugiej przekatnej: 33
Kwadrat magiczny: tak
```

Wskazówka: tablicę można wypełnić przy pomocy klasy `std::istringstream` zdefiniowanej w nagłówku `<sstream>` jak pokazano poniżej:

```

const char s[] = "1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 17";
std::istringstream iss(s);
// Teraz iss można używać jak std::cin, czyli
for (...) {
    iss >> tab[...];
}
```

3. 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: AlaA

Podany ciąg jest palindromem.

4. 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

5. Napisz program, który sprawdza, czy zadana dwuwymiarowa tablica 9×9 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

```