Wydział Matematyki i Nauk Informacyjnych

Politechnika Warszawska

Algorytmy i podstawy programowania - ćwiczenia

Marek Gągolewski

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

PROGRAM ROZWOJOWY
POLITECHNIKI WARSZAWSKIEJ

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Zadanie 1.1. Dany jest algorytm Euklidesa znajdywania największego wspólnego dzielnika (NWD) dwóch liczb $a, b \in \mathbb{Z}$, $0 \leqslant a < b$.

```
//    Wejście:  0 ⩽ a < b
//    Wyjście:  NWD(a,b)

niech c ∈ ℕ;
dopóki (a ≠ 0)
{
    c = reszta z dzielenia b przez a;
    b = a;
    a = c;
}
zwróć b jako wynik;
```

Prześledź działanie algorytmu Euklidesa (jakie wartości przyjmują zmienne a, b, c w każdym kroku) znajdywania największego wspólnego dzielnika dla następujących par liczb

a) 42, 56,
b) 192, 348,
c) 199, 544,
d) 2166, 6099.

Zadanie 1.2. Pokaż, w jaki sposób za pomocą ciągu przypisań można przestawić wartości dwóch zmiennych (a, b), by otrzymać (b, a).

Zadanie 1.3. Pokaż, w jaki sposób za pomocą ciągu przypisań można przestawić wartości trzech zmiennych (a, b, c), by otrzymać (c, a, b).

Zadanie 1.4. Pokaż, w jaki sposób za pomocą ciągu przypisań można przestawić wartości czterech zmiennych (a, b, c, d), by otrzymać (c, d, b, a).

Zadanie 1.5. Dany jest ciąg n liczb rzeczywistych $\mathbf{x} = (x[0], x[1], \ldots, x[n-1])$ (umawiamy się, że elementy ciągów numerujemy od 0). Rozważmy ich średnią arytmetyczną, określoną jako

$$\frac{1}{n}\sum_{i=0}^{n-1} x[i] = \frac{1}{n}\left(x[0] + x[1] + \cdots + x[n-1]\right).$$

Rozważmy następujący algorytm służący do jej wyznaczania.

```
// Wejście: n > 0 oraz x[0], x[1], ..., x[n−1] ∈ ℝ
// Wyjście: średnia arytmetyczna elementów x[0], x[1], ..., x[n−1])
niech sredniaarytm ∈ ℝ;
niech i ∈ ℕ;
sredniaarytm = 0;
dla (i=0,1,...,n−1)
    sredniaarytm = sredniaarytm + x[i];
sredniaarytm = sredniaarytm / n;
zwróć sredniaarytm jako wynik;
```

Wyznacz za pomocą powyższego algorytmu wartość średniej arytmetycznej dla ciągów $(1, -1, 2, 0, -2)$ oraz $(34, 2, -3, 4, 3{,}5)$.

Zadanie 1.6. Dany jest ciąg n dodatnich liczb rzeczywistych $\mathbf{x} = (x[0], x[1], \ldots, x[n-1])$. Napisz algorytm, który wyznaczy ich średnią harmoniczną, określoną jako

$$\frac{n}{\sum_{i=0}^{n-1} \frac{1}{x[i]}} = \frac{n}{1/x[0] + 1/x[1] + \cdots + 1/x[n-1]}.$$

Wyznacz za pomocą tego algorytmu wartość średniej harmonicznej dla ciągów $(1, 4, 2, 3, 1)$ oraz $(10, 2, 3, 4)$.

$\star$ **Zadanie 1.7.** Dany jest ciąg n liczb rzeczywistych $\mathbf{x} = (x[0], x[1], \ldots, x[n-1])$. Rozważmy tzw. sumę kwadratów odchyleń elementów od ich średniej arytmetycznej, określoną jako

$$\mathrm{SKO}(\mathbf{x}) = \sum_{i=0}^{n-1} \left(x[i] - \left(\frac{1}{n} \sum_{j=0}^{n-1} x[j] \right) \right)^2.$$

Rozważmy następujący algorytm służący do wyznaczania SKO.

```
// Wejście: n > 0 oraz x[0], x[1], ..., x[n − 1] ∈ ℝ
// Wyjście: SKO(x[0], x[1], ..., x[n − 1])
niech sko, sredniaarytm ∈ ℝ;
niech i, j ∈ ℕ;

sko = 0;
dla (i=0,1,...,n−1)
{
   sredniaarytm = 0;
   dla (j=0,1,...,n−1)
   {
      sredniaarytm = sredniaarytm + x[j];
   }
   sredniaarytm = sredniaarytm / n;

   sko = sko + (x[i] − sredniaarytm)*(x[i] − sredniaarytm);
}
zwróć sko jako wynik;
```

a) Wyznacz za pomocą powyższego algorytmu wartość SKO dla $\mathbf{x} = (5, 3, -1, 7, -2)$.

b) Policz, ile łącznie operacji arytmetycznych $(+, -, *, /)$ potrzebnych jest do wyznaczenia SKO dla ciągów wejściowych o $n = 5, 10, 100, 1000, 10000$ elementach. Wyraź tę liczbę jako funkcję długości ciągu wejściowego n.

c) Zastanów się, jak usprawnić powyższy algorytm, by nie wykonywać wielokrotnie zbędnych obliczeń. Ile teraz będzie potrzebnych operacji arytmetycznych?

Zadanie 2.1. Przedstaw następujące liczby całkowite nieujemne w postaci binarnej, dziesiętnej i szesnastkowej: $10_{10}, 18_{10}, 18_{16}, 101_{2}, 101_{10}, 101_{16}, \mathrm{ABCDEF}_{16}, 64135312_{10}, 110101111001_{2}$, $\mathrm{FFFFF0C2}_{16}$.

Zadanie 2.2. Dana jest n cyfrowa liczba w systemie U2 zapisana jako ciąg cyfr $b_{n-1}b_{n-2}\ldots b_1b_0$, gdzie $b_i \in \{0,1\}$ dla $i = 0, 1, \ldots, n-1$. Pokaż, że powielenie pierwszej cyfry dowolną liczbę razy nie zmienia wartości danej liczby, tzn. $b_{n-1}b_{n-2}\ldots b_1b_0 = b_{n-1}b_{n-1}\ldots b_{n-1}b_{n-2}\ldots b_1b_0$.

Zadanie 2.3. Przedstaw następujące liczby dane w notacji U2 jako liczby dziesiętne: 10111100, 00111001, 1000000111001011.

$\star$ **Zadanie 2.4.** Przedstaw następujące liczby dziesiętne w notacji U2 (do zapisu użyj 8, 16 lub 32 bitów): $-12, 54, -128, -129, 53263, -32000, -56321, -3263411$.

$\star$ **Zadanie 2.5.** Rozważmy operacje dodawania i odejmowania liczb nieujemnych w reprezentacji binarnej. Oblicz wartość następujących wyrażeń korzystając z metody analogicznej do sposobu „szkolnego" (dodawania i odejmowanie słupkami). Pamiętaj jednak, że np. $1_2 + 1_2 = 10_2$.

a) $1000_2 + 111_2$,
b) $1000_2 + 1111_2$,
c) $1110110_2 + 11100111_2$,
d) $11111_2 + 11111111_2$,
e) $1111_2 - 0001_2$,
f) $1000_2 - 0001_2$,
g) $10101010_2 - 11101_2$,
h) $11001101_2 - 10010111_2$.

$\star$ **Zadanie 2.6.** Okazuje się, że liczby w systemie U2 można dodawać i odejmować tą samą metodą, co liczby nieujemne w reprezentacji binarnej. Oblicz wartość następujących wyrażeń i sprawdź otrzymane wyniki, przekształcając je do postaci dziesiętnej. Uwaga: operacji dokonuj na dwóch liczbach n bitowych, a wynik podaj również jako n bitowy.

a) $1000_{\mathrm{U2}} + 0111_{\mathrm{U2}}$,
b) $10110110_{\mathrm{U2}} + 11100111_{\mathrm{U2}}$,
c) $10101010_{\mathrm{U2}} - 00011101_{\mathrm{U2}}$,
d) $11001101_{\mathrm{U2}} - 10010111_{\mathrm{U2}}$.
e) $10001101_{\mathrm{U2}} - 01010111_{\mathrm{U2}}$.

$\star$ **Zadanie 2.7.** Niech dana będzie liczba w postaci U2. Pokaż, że aby uzyskać liczbę do niej przeciwną, należy odwrócić wartości jej bitów (dokonać ich negacji, tzn. zamienić zera na jedynki i odwrotnie) i dodać do wyniku wartość 1. Ile wynoszą wartości $0101_{\mathrm{U2}}, 1001_{\mathrm{U2}}$ i 0111_{U2} po dokonaniu tych operacji? Sprawdź uzyskane rezultaty za pomocą konwersji tych liczb do systemu dziesiętnego.

Zadanie 3.1. Wyraź następujące liczby zmiennoprzecinkowe dane w notacji naukowej w zwykłej postaci dziesiętnej.

a) $4.21e6$,
b) $1.95323e2$,
c) $2.314e-4$,
d) $4.235532e-2$.

Zadanie 3.2. Wyznacz wartość następujących wyrażeń. Ponadto określ typ zwracany przez każde z nich.

a) 10.0+15.0/2+4.3,
b) 10.0+15/2+4.3,
c) 3.0*4/6+6,
d) 20.0−2/6+3,
e) 10+17*3+4,
f) 10+17/3.0+4,
g) 3*4%6+6,
h) 3.0*4%6+6,
i) 10+17%3+4.

Zadanie 3.3. Dane są 3 zmienne zadeklarowane w sposób następujący.

```
double a = 10.6, b = 13.9, c = -3.42;
```

Oblicz wartość poniższych wyrażeń.

a) int(a),
b) int(c),
c) int(a+b),
d) int(a)+b+c,
e) int(a+b)*c,
f) double(int(a+b)/int(c)),
g) double(int(a))/c.

Zadanie 3.4. Korzystając z przekształceń logicznych (np. praw De Morgana, praw rozdzielności) uprość następujące wyrażenia (zakładamy, że a,b,c,d są typu **double**, a p,q,r typu **bool**).

a) !(!p),
b) !p && !q ,
c) !(!p || !q || !r),
d) !(b>a && b<c),
e) !(a>=b && b>=c && a>=c),
f) (a>b && a<c) || (a<c && a>d),
g) p || !p .

Zadanie 3.5. Jaki wynik dadzą następujące operacje bitowe wykonane na danych typu **short**?

a) 0x0FCD | 0xFFFF,
b) 364 & 0x323,
c) ~163,
d) 0xFC93 ^ 0x201D,
e) 14 << 4,
f) 0xf5a3 >> 8,
g) 0x3a9f >> 7.

Zadanie 3.6. Niech dana będzie zmienna typu **int**. W jaki sposób dokonać zmiany znaku wartości tej zmiennej na przeciwny nie używając operatora −?

Zadanie 3.7. Policyjny fotoradar emituje fale elektromagnetyczne o częstotliwości f_e Hz. Wiązka tych fal jest odbijana od nadjeżdżającego z prędkością v samochodu i, jako że auto jest w ruchu, powraca do urządzenia ze zmienioną częstotliwością f_o Hz. Polscy funkcjonariusze testują właśnie najnowszy model brytyjskiej „suszarki". Związek pomiędzy omawianymi zmiennymi można wyrazić równaniem

$$v = 6{,}685 \times 10^8 \, \frac{f_o - f_e}{f_o + f_e}.$$

Prędkość jednak jest podawana w milach na godzinę. Wiedząc, że 1 mila to 1609,344 m, napisz fragment kodu w języku C++, który dla danego f_e i f_o poda prędkość nadjeżdżającego samochodu w km/h. Ponadto wypisz na ekran wartość logiczną mówiącą, czy została przekroczona dopuszczalna prędkość, wynosząca w tym miejscu 50 km/h.

Jaki będzie wynik działania tej procedury dla $f_e = 2 \times 10^{10}$ Hz i $f_o = 2.000004 \times 10^{10}$ Hz?

$\star$ **Zadanie 3.8.** Danych jest 6 zmiennych ax,ay,bx,by,cx,cy typu **double**, reprezentujących współrzędne 3 punktów w $\mathbb{R}^2$: $\mathbf{a} = (ax,ay)$, $\mathbf{b} = (bx,by)$, $\mathbf{c} = (cx,cy)$. Napisz fragment kodu, który wyznaczy kwadrat promienia okręgu przechodzącego przez $\mathbf{a}$, $\mathbf{b}$ i $\mathbf{c}$. Dany jest on wzorem

$$r^2 = \frac{|\mathbf{a}-\mathbf{c}|^2 \, |\mathbf{b}-\mathbf{c}|^2 \, |\mathbf{a}-\mathbf{b}|^2}{4\,|(\mathbf{a}-\mathbf{c}) \times (\mathbf{b}-\mathbf{c})|^2},$$

gdzie np. $|\mathbf{a}-\mathbf{b}| = \sqrt{(\mathrm{ax}-\mathrm{bx})^2 + (\mathrm{ay}-\mathrm{by})^2}$ oraz $\mathbf{a} \times \mathbf{b} = \mathrm{axby} - \mathrm{aybx}$.

Zadanie 3.9. Dane są $a, b, c, d, e, f \in \mathbb{R}$ takie, że układ dwu równań liniowych względem niewiadomych $x, y \in \mathbb{R}$:

$$\begin{cases} ax + by &= c, \\ dx + ey &= f. \end{cases}$$

jest oznaczony. Zaproponuj fragment kodu w języku C++, który wyznaczy jego rozwiązanie, tzn. obliczy wartość zmiennych x,y na podstawie pewnych wartości zmiennych a,b,c,d,e,f. Do reprezentacji liczb rzeczywistych użyj typu **double**.

Zadanie 4.1. Wyraź następujące pętle dane w sposób opisowy za pomocą instrukcji **for**.

a) **dla** $i = 0, 1, \ldots, n-1$ wypisz i (dla pewnego $n \in \mathbb{N}$),
b) **dla** $i = n, n-1, \ldots, 0$ wypisz i (dla pewnego $n \in \mathbb{N}$),
c) **dla** $j = 1, 3, \ldots, 2k-1$ wypisz j (dla pewnego $k \in \mathbb{N}$),
d) **dla** $i = 1, 2, 4, 7, \ldots, n$ wypisz i (dla pewnego $n \in \mathbb{N}$),
e) **dla** $j = 1, 2, 4, 8, 16, \ldots n$ wypisz j (dla pewnego $n \in \mathbb{N}$),
f) **dla** $j = 1, 2, 4, 8, 16, \ldots, 2^k$ wypisz j (dla pewnego $k \in \mathbb{N}$),
g) **dla** $x = a, a+\delta, a+2\delta, \ldots, b$ wypisz x (dla pewnych $a, b, \delta \in \mathbb{R}, a < b, \delta > 0$).

Zadanie 4.2. Zaprogramuj w języku C++ algorytm Euklidesa do wyznaczania największego wspólnego dzielnika dwóch liczb (zob. zestaw zadań nr 1).

Zadanie 4.3. Spośród liczb $1, 2, \ldots, 100$ wypisz na ekran wszystkie podzielne przez 7, tzn. 7,14,21,. . . .

Zadanie 4.4. Spośród liczb $1, 2, \ldots, 100$ wypisz na ekran wszystkie podzielne przez 2 lecz niepodzielne przez 5, tzn. 2,4,6,8,12, ...

Zadanie 4.5. Spośród liczb $1, 2, \ldots, 100$ wypisz na ekran co drugą podzielną przez 5 lub podzielną przez 7, tzn. 5, 10, 15, 21, 28, ...

⋆ **Zadanie 4.6.** Napisz fragment kodu, który sprawdzi, czy następujące liczby są pierwsze: 7,93,97,6687,6689, 6691.

Zadanie 4.7. Napisz fragment kodu, który znajduje minimum z danych liczb całkowitych dodatnich. Liczby odczytuj z klawiatury, póki użytkownik nie wprowadzi 0.

Zadanie 4.8. Napisz fragment kodu, który znajduje różnicę między maksimum a minimum z danych liczb rzeczywistych nieujemnych. Liczby odczytuj z klawiatury, póki użytkownik nie wprowadzi liczby ujemnej.

Zadanie 4.9. Napisz fragment kodu, który aproksymuje wartość liczby π za pomocą wzoru

$$\pi \simeq 4\left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \ldots\right).$$

Wypisz kolejne przybliżenia korzystając z $1, 2, 3, \ldots, 25$ początkowych wyrazów tego szeregu.

Zadanie 4.10. Napisz fragment kodu, który aproksymuje wartość liczby e za pomocą wzoru

$$e \simeq 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \ldots,$$

gdzie $n! = 1 \times 2 \times \cdots \times n$. Wypisz wynik dopiero wtedy, gdy różnica pomiędzy kolejnymi wyrazami szeregu będzie mniejsza niż 10^{-9}.

Zadanie 4.11. Napisz fragmenty kodu, które posłużą do wyznaczenia wartości następujących wyrażeń.

a) 2^n dla pewnego $n \in \mathbb{N}$,
b) $\sum_{i=1}^{10} i$,
c) $\sum_{i=1}^{100} \frac{1}{i!}$,
d) $\prod_{i=1}^{5} \frac{i}{i+1}$,
e) $e^x \simeq \sum_{n=0}^{100} \frac{x^n}{n!}$ dla pewnego $x \in \mathbb{R}$,
f) $\ln(1+x) \simeq \sum_{n=1}^{100} \frac{(-1)^{n+1}}{n} x^n$ dla pewnego $x \in [-1, 1]$,
g) $\sin x \simeq \sum_{n=0}^{100} \frac{(-1)^n}{(2n+1)!} x^{2n+1}$ dla pewnego $x \in \mathbb{R}$,
h) $\cos x \simeq \sum_{n=0}^{100} \frac{(-1)^n}{(2n)!} x^{2n}$ dla pewnego $x \in \mathbb{R}$,
i) $\arcsin x \simeq \sum_{n=0}^{100} \frac{(2n)!}{4^n (n!)^2 (2n+1)} x^{2n+1}$ dla pewnego $x \in (-1, 1)$.

Przypomnijmy, że zgodnie z umową np. $\sum_{i=1}^{10} i = 1 + 2 + \cdots + 10$ oraz np. $\prod_{i=1}^{5} \frac{i}{i+1} = \frac{1}{2} \times \frac{2}{3} \times \cdots \times \frac{5}{6}$.

Zadanie 4.12. Korzystając ze wzoru na przybliżoną wartość funkcji $\sin$ podanego w zad. 4.11, utwórz kod, który wydrukuje tablice przybliżonych wartości $\sin x$ dla $x = \frac{k}{n}\pi$, $k = 0, 1, \ldots, n$ i pewnego n, np. $n = 10$. Wynik niech będzie postaci podobnej do poniższej.

```
  x          sin(x)
0.0000000  0.0000000
0.3141593  0.3090170
...
3.1415927  0.0000000
```

Zadanie 4.13. Pobierz wartości zmiennych *a*,*b* typu **double** z klawiatury. Będą one definiować równanie względem niewiadomej $x \in \mathbb{R}$ postaci *a* x + *b* = 0. Zaproponuj fragment kodu w języku C++, który wyznaczy jego rozwiązanie. Poprawnie identyfikuj przypadek, gdy dane równanie nie jest równaniem liniowym.

Zadanie 4.14. Dla danych $a, b, c, d, e, f \in \mathbb{R}$ zaproponuj kod w języku C++ do rozwiązywania układu dwóch równań liniowych względem niewiadomych $x, y \in \mathbb{R}$:

$$\begin{cases} ax + by &= c, \\ dx + ey &= f. \end{cases}$$

Algorytm ten powinien poprawnie identyfikować przypadki (np. wypisując stosowny komunikat na ekranie), w których dany układ jest sprzeczny bądź nieoznaczony. Współczynniki układu pobierz z klawiatury. Do reprezentacji zbioru $\mathbb{R}$ użyj typu **double**.

⋆ **Zadanie 4.15.** Napisz fragmenty kodu, które sprawdzą, czy następujące zdania logiczne są tautologiami.

a) $p \wedge \neg p$,
b) $\neg(\neg p) \Leftrightarrow p$,
c) $p \vee q \Leftrightarrow p \vee q$,
d) $\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$,
e) $p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$,
f) $(p \Leftrightarrow q) \Leftrightarrow (p \vee q) \wedge (\neg p \vee \neg q)$.

⋆ **Zadanie 4.16.** Dla danej zmiennej typu **int** napisz program, który wypisze na ekran jej wartość w postaci binarnej.

Zadanie 5.1. Niech dana będzie tablica zadeklarowana jako **int** *tab*[*n*], dla pewnego *n*. Napisz kod, który przesunie każdy element o indeksie > 0 o jedną komórkę w lewo, a element pierwszy wstawi na miejsce ostatniego.

Zadanie 5.2. Za pomocą tylko jednej pętli **for** znajdź w tablicy **double** *tab*[*n*] element najmniejszy i największy.

Zadanie 5.3. Napisz fragment kodu, który w tablicy **bool** *tab*[*n*] zliczy, ile razy występuje wartość **true** oraz dokona negacji wszystkich elementów.

Zadanie 5.4. Napisz kod, który znajduje najmniejszy element w tablicy **int** *tab*[*n*]. Następnie wypełnij tym elementem wszystkie komórki o parzystych indeksach oraz elementem przeciwnym do niego komórki o indeksach nieparzystych.

Zadanie 5.5. Dla danej tablicy liczb rzeczywistych *t* rozmiaru *n* napisz kod, który wyznaczy wartość średniej arytmetycznej wszystkich elementów, danej wzorem $\frac{1}{n}\sum_{i=0}^{n-1} t[i]$.

Zadanie 5.6. Niech dany będzie n-elementowy ciąg liczb rzeczywistych $\mathbf{a} = (a_1, a_2, \ldots, a_n)$. Średnią geometryczną nazywamy wartość

$$\mathsf{GM}(\mathbf{a}) = \sqrt[n]{\prod_{i=1}^{n} a_i}.$$

Napisz program, który wczyta do tablicy $n = 8$ wartości z klawiatury oraz następnie policzy wartość ich średniej geometrycznej.

Zadanie 5.7. Niech dany będzie n-elementowy ciąg liczb rzeczywistych $\mathbf{a} = (a_1, a_2, \ldots, a_n)$. Średnią ważoną względem wektora wag $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ o elementach nieujemnych oraz takiego, że $\sum_{i=1}^{n} w_i = 1$, nazywamy wartość

$$\mathsf{WM}_{\mathbf{w}}(\mathbf{a}) = \sum_{i=1}^{n} a_i w_i.$$

Napisz program, który dla $n = 5$ wczyta z klawiatury wektor wag $\mathbf{w}$ i sprawdzi, czy spełnia on postawione wyżej założenia oraz wyznaczy wartość średniej ważonej ciągu $(-2, -1, 0, 1, 2)$.

Zadanie 5.8. Niech dany będzie n-elementowy ciąg liczb rzeczywistych $\mathbf{a} = (a_1, a_2, \ldots, a_n)$. Operatorem maks-min względem wektora wag $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ składającego się z wartości rzeczywistych, nazywamy wartość

$$\mathsf{MaxMin}_{\mathbf{w}}(\mathbf{a}) = \max_{i=1,2,\ldots,n} \left(\min\{a_i, w_i\}\right).$$

Napisz program, który dla $n = 5$ wczyta z klawiatury ciąg $\mathbf{a}$, następnie wyznaczy wartość operatora maks-min względem wektora wag $(1, 2, \ldots, n)$.

Zadanie 5.9. Niech dany będzie wektor o elementach rzeczywistych $\mathbf{x} = (x_1, \ldots, x_n)$. Napisz program, który wczyta wartości jego elementów z klawiatury (dla $n = 9$) oraz policzy wartość jego normy euklidesowej, wg wzoru

$$|\mathbf{x}| = \sqrt{\sum_{i=1}^{n} x_i^2}.$$

Zadanie 5.10. Niech dane będą dwa n-elementowe wektory o elementach rzeczywistych $\mathbf{x} = (x_1, \ldots, x_n)$ i $\mathbf{y} = (y_1, \ldots, y_n)$. Napisz program, który wczyta wartości ich elementów z klawiatury (dla $n = 5$) oraz policzy wartość ich odległości w metryce supremum, wg wzoru

$$d_m(\mathbf{x}, \mathbf{y}) = \max_{i=1,2,\ldots,n} |x_i - y_i|.$$

Zadanie 5.11. Niech dany będzie wielomian rzeczywisty stopnia n, $w(x) = \text{w[0]}x^0 + \text{w[1]}x^1 + \ldots \text{w[n]}x^n$, $w[n] \neq 0$, którego współczynniki przechowywane są w $n+1$ wymiarowej tablicy o elementach typu **double**. Napisz program, który wczytuje wartość współczynników dla $n = 3$ oraz wartość x i wyznacza wartość $w(x)$ wg powyższego wzoru.

⋆ **Zadanie 5.12.** Zmodyfikuj program z zad. 5.11 tak, by korzystał z bardziej efektywnego obliczeniowo wzoru na wartość $w(x)$, zwanego schematem Hornera:

$$w(x) = (\cdots(((\text{w[n]}x + \text{w[n}-\text{1]})\, x + \text{w[n}-\text{2]})\, x + \text{w[n}-\text{3]}) \cdots)\, x + \text{w[0]}.$$

⋆ **Zadanie 5.13.** Niech dane będą wielomiany $w(x)$ i $v(x)$ stopnia, odpowiednio, n i m. Napisz program, który wyznaczy wartości współczynników wielomianu $u(x)$ stopnia $n+m$, będącego iloczynem wielomianów w i v. Dokonaj obliczeń dla $w(x) = x^4 + 4x^2 - x + 2$ oraz $v(x) = x^4 + x^3 + 10$.

Zadanie 5.14. Zaimplementuj algorytm sortowania przez wybór dla danej tablicy o *n* elementach typu **int**. Oblicz, ile jest potrzebnych operacji porównań oraz przestawień elementów w zależności od *n*.

Zadanie 5.15. Zaimplementuj algorytm sortowania przez wstawianie dla danej tablicy o *n* elementach typu **int**. Oblicz, ile jest potrzebnych operacji porównań oraz przestawień elementów w zależności od *n* dla tablicy już posortowanej oraz dla tablicy posortowanej w kolejności odwrotnej.

Zadanie 5.16. Zaimplementuj algorytm sortowania bąbelkowego dla danej tablicy o *n* elementach typu **int**. Oblicz, ile jest potrzebnych operacji porównań oraz przestawień elementów w zależności od *n* dla tablicy już posortowanej oraz dla tablicy posortowanej w kolejności odwrotnej.

⋆ **Zadanie 5.17.** Dana jest tablica o *n*>1 elementach typu **double**. Napisz funkcję, która obliczy wariancję jej elementów używając tylko jednej pętli **for**. Wariancja elementów ciągu $\mathbf{x} = (x_1, \ldots, x_n)$ dana jest wzorem

$$s^2(\mathbf{x}) = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{\mathbf{x}})^2, \tag{1}$$

gdzie $\overline{\mathbf{x}}$ jest średnią arytmetyczną ciągu $\mathbf{x}$.

Zadanie 6.1. Napisz funkcję parzysta, która sprawdza czy dany argument typu **int** jest liczbą parzystą czy nieparzystą. Zwróć wynik typu **bool**.

Zadanie 6.2. Napisz funkcję silnia, która dla danego $n \in \mathbb{N}$ zwraca wartość $1 \times 2 \times \cdots \times n$.

Zadanie 6.3. Napisz funkcję max, która dla danych $a, b, c \in \mathbb{Z}$ zwraca ich maksimum.

Zadanie 6.4. Napisz funkcję med, która znajduje medianę (wartość środkową) trzech liczb rzeczywistych, np. $\mathrm{med}(4, 2, 7) = 4$ i $\mathrm{med}(1, 2, 3) = 2$.

Zadanie 6.5. Napisz funkcję nwd zwracającą największy wspólny dzielnik dwóch liczb naturalnych.

Zadanie 6.6. Napisz funkcję nww zwracającą najmniejszą wspólną wielokrotność dwóch liczb naturalnych.

Zadanie 6.7. Napisz funkcję o nazwie bmi, która jako argument przyjmuje wzrost (w m) i wagę (w kg) pacjenta, a jako wynik zwraca jego wskaźnik masy ciała (BMI), określony jako $\mathrm{BMI} = \mathrm{waga}/\mathrm{wzrost}^2$. (Ciekawostka: wg WHO BMI od $18{,}5$ do $25{,}0$ jest uznawana za wartość prawidłową.)

Zadanie 6.8. Napisz funkcję odl, która przyjmuje współrzędne rzeczywiste dwóch punktów *x1*, *y1*, *x2*, *y2* i zwraca ich odległość euklidesową daną wzorem $|\mathbf{x} - \mathbf{y}| = \sqrt{(x1 - y1)^2 + (x2 - y2)^2}$.

Zadanie 6.9. Napisz funkcję odlsup, która przyjmuje współrzędne rzeczywiste dwóch punktów *x1*,*y1*,*x2*,*y2* i zwraca ich odległość w metryce supremum, tj.

$$d_m(\mathbf{x}, \mathbf{y}) = \max\{|x1 - y1|, |x2 - y2|\}.$$

Zadanie 6.10. Napisz funkcję odlLp, która przyjmuje współrzędne rzeczywiste dwóch punktów *x1*,*y1*,*x2*,*y2* i zwraca ich odległość w metryce L^p, gdzie $p \in [1, \infty)$ jest także parametrem funkcji, wg wzoru

$$||\mathbf{x} - \mathbf{y}||_p = \sqrt[p]{|x1 - y1|^p + |x2 - y2|^p}.$$

Zadanie 6.11. $\sum_{i=1}^{\infty} \frac{(-1)^{i+1}}{i} x^i$ jest rozwinięciem funkcji $\ln(x+1)$ dla $(-1, 1]$ w szereg Taylora. Napisz funkcję lognat02, która dla danego $x \in (0, 2]$ zwraca przybliżenie wartości $\ln x$, a dla $x \notin (0, 2]$ wartość NaN.

Zadanie 6.12. Wiemy, że szereg $\sum_{i=0}^{\infty} \frac{(-1)^i}{(1-2i)} \frac{(2i)!}{(i!)^2} \frac{x^i}{(4^i)}$ dla $|x| < 1$ jest zbieżny i jego suma równa jest $\sqrt{1 + x}$. Napisz funkcję pierw02, która dla danej liczby rzeczywistej $x \in [0, 2]$ znajduje przybliżenie jej pierwiastka na podstawie podanego wzoru, a dla liczb $x \notin [0, 2]$ zwraca NaN.

Zadanie 6.13. Dane są rozwinięcia następujących funkcji w szereg Taylora.

a) $e^x = \sum_{i=0}^{\infty} \frac{x^n}{n!}$ dla pewnego $x \in \mathbb{R}$,

b) $\sin x = \sum_{\infty}^{100} \frac{(-1)^n}{(2n+1)!} x^{2n+1}$ dla pewnego $x \in \mathbb{R}$,

c) $\cos x = \sum_{\infty}^{100} \frac{(-1)^n}{(2n)!} x^{2n}$ dla pewnego $x \in \mathbb{R}$,

d) $\arcsin x = \sum_{\infty}^{100} \frac{(2n)!}{4^n (n!)^2 (2n+1)} x^{2n+1}$ dla pewnego $x \in (-1, 1)$.

Napisz funkcje w C++, które przybliżają wartości powyższych sum bądź zwracają NaN dla argumentów poza obszarem zbieżności.

Zadanie 6.14. Napisz funkcję zaokr, która dla liczby $x \in \mathbb{R}$ wyznacza jej zaokrąglenie dziesiętne — z dokładnością do podanej liczby cyfr dziesiętnych k — jako $\lfloor x\, 10^k + 0{,}5 \rfloor / 10^k$, gdzie $\lfloor u \rfloor$ jest funkcją „podłoga".

Zadanie 6.15. Niech dane będą a,b,c typu **double**. Zmienne te definiują równanie względem niewiadomej $x \in \mathbb{R}$ postaci

$$ax^2 + bx + c = 0.$$

Zaproponuj funkcję w języku C++, która wyznaczy jego rozwiązanie i wypisze wynik na ekran. Poprawnie identyfikuj przypadki, gdy dane równanie nie ma rozwiązań w $\mathbb{R}$, a także, gdy nie jest ono równaniem kwadratowym.

Zadanie 6.16. Napisz funkcję implementującą grę w „Zgadulę". Losuje ona liczbę całkowitą z zakresu od 1 do 100. Użytkownik próbuje odgadnąć liczbę wprowadzając swe typy z klawiatury, póki jej nie zgadnie. Za każdym razem otrzymuje komunikat zwrotny, np. „za mało" bądź „za dużo".

Zadanie 6.17. Napisz funkcję implementującą grę w „Zgadulę" zawierającą elementy „sztucznej inteligencji". Losuje ona liczbę całkowitą z zakresu od 1 do 100. Następnie komputer sam próbuje odgadnąć tę liczbę, przy okazji wypisując swe typy na ekranie. Zaproponuj prosty algorytm, który (nie oszukując!) znajdzie poprawne rozwiązanie w średnio jak najmniejszej liczbie kroków.

Zadanie 7.1. Można pokazać, że $\sum_{i=0}^{\infty} \frac{x^i}{i!}$ jest rozwinięciem funkcji e^x w szereg Taylora. Napisz dwie wersje funkcji Exp, które dla danego $x \in \mathbb{R}$ znajdują przybliżenie jego eksponensu na podstawie podanego wzoru.

a) W jednej rozpatrz tylko n początkowych wyrazów szeregu, np. $n = 30$,
b) W drugiej przerwij obliczenia dopiero, gdy moduł kolejnego dodawanego wyrazu jest mniejszy niż założone δ, np. $\delta = 10^{-6}$.

Rada: Niech n i δ będą parametrami funkcji z wartościami domyślnymi.

Zadanie 7.2. Napisz kilka przeciążonych wersji funkcji swap, które przestawiają wartości dwóch argumentów wejściowych o typach **int**, **double** i **bool**.

Zadanie 7.3. Napisz nierekurencyjną funkcję służącą do znalezienie n-tej liczby Fibonacciego.

Zadanie 7.4. Napisz funkcję swap, która za pomocą ciągu przypisań przestawia wartości czterech zmiennych rzeczywistych (a, b, c, d), tak by na wyjściu otrzymać (c, b, d, a).

Zadanie 7.5. Napisz funkcję sort, która porządkuje niemalejąco wartości trzech argumentów wejściowych (liczby całkowite).

Zadanie 7.6. Zaproponuj funkcję wyznaczającej wartość tzw. funkcji 91 McCarthy'ego.

$$M(n) = \begin{cases} n - 10 & \text{dla } n > 100, \\ M(M(n + 11)) & \text{dla } n \leqslant 100. \end{cases}$$

Ciekawostka: okazuje się, że $M(n) = 91$ dla każdego $n \leqslant 101$ oraz $M(n) = M(n) - 10$ dla $n > 101$.

Zadanie 8.1. Zaimplementuj samodzielnie funkcje z biblioteki `<cstring>`: *strlen* (), *strcpy* (), *strncpy* (), *strcat* (), *strncat* (), *strcmp* (), *strstr* (), *strchr* (), *strrchr* ().

Zadanie 8.2. Napisz funkcję, która w danym łańcuchu znaków zamieni wszystkie małe litery alfabetu łacińskiego na wielkie.

Zadanie 8.3. Napisz funkcję, która usunie wszystkie znaki odstępów (spacje) z końca danego łańcucha znaków.

Zadanie 8.4. Napisz funkcję, która usunie wszystkie znaki odstępów (spacje) z początku danego łańcucha znaków.

Zadanie 8.5. Napisz funkcję, która usunie z danego napisu wszystkie znaki niebędące cyframi bądź kropką.

Zadanie 8.6. Napisz funkcję, która odwróci kolejność znaków w danym napisie.

Zadanie 8.7. Napisz funkcję, która jako parametr przyjmuje dwa łańcuchy znaków i zwraca nowy, dynamicznie alokowany napis będący ich połączeniem, np. dla "ala" i "ola" wynikiem powinno być "alaola".

Zadanie 8.8. Napisz funkcję, która oblicza, ile razy w danym napisie występuje dany znak.

Zadanie 8.9. Napisz funkcję, która oblicza, ile razy w danym napisie występuje dany inny łańcuch znaków, np. w "ababbababa" łańcuch "aba" występuje 3 razy.

Zadanie 8.10. Napisz funkcję, która dla danej liczby **int** zwróci dynamicznie alokowany łańcuch znaków, składający się z symboli 0 lub 1, przechowujący binarną reprezentację argumentu.

Zadanie 8.11. Napisz funkcję, która dla danego łańcucha znaków, składającego się z symboli 0 lub 1, reprezentującego pewną liczbę w postaci binarnej, zwróci jej wartość jako zmienną typu **int**.

Zadanie 8.12. Napisz funkcję, która dla danego łańcucha znaków, składającego się z symboli 0 lub 1, reprezentującego pewną liczbę w postaci binarnej, zwróci dynamicznie alokowany napis przechowujący jej szesnastkową reprezentację.

Zadanie 8.13. Palindrom[1] to ciąg liter, które są takie same niezależnie od tego, czy czytamy je od przodu czy od tyłu, np. *kobyłamamałybok*, *możejutrotadamasamadatortujeżom*, *ikarłapałraki*. Napisz funkcję sprawdzającą czy dany napis jest palindromem. Zwróć wartość typu **bool**.

[1]Zob. http://www.palindromy.pl.

Zadanie 9.1. Dana jest macierz A typu $n \times m$ o wartościach rzeczywistych oraz liczba $k \in \mathbb{R}$. Napisz funkcję, która wyznaczy wartość kA, czyli implementującą mnożenie macierzy przez skalar.

Zadanie 9.2. Dana jest macierz A typu $n \times m$ o wartościach rzeczywistych oraz wektor $\mathbf{b} \in \mathbb{R}^n$. Napisz funkcję, która zwróci macierz $[A|\mathbf{b}]$, czyli A rozszerzoną o nową kolumnę, której wartości pobrane są z $\mathbf{b}$.

Zadanie 9.3. Dana jest macierz A typu $n \times m$ o wartościach rzeczywistych oraz wektor $\mathbf{b} \in \mathbb{R}^m$. Napisz funkcję, która zwróci macierz A rozszerzoną o nowy wiersz, którego wartości pobrane są z $\mathbf{b}$.

Zadanie 9.4. Dana jest macierz A typu 2×2 o wartościach rzeczywistych. Napisz funkcję, która zwróci wyznacznik danej macierzy.

Zadanie 9.5. Dana jest macierz A typu 3×3 o wartościach rzeczywistych. Napisz funkcję, która zwróci wyznacznik danej macierzy.

$\star$ **Zadanie 9.6.** Dana jest macierz kwadratowa A o 4 wierszach i 4 kolumnach zawierająca wartości rzeczywiste. Napisz rekurencyjną funkcję, która zwróci wyznacznik danej macierzy. Skorzystaj wprost z definicji wyznacznika. Uwaga: taka metoda jest zbyt wolna, by korzystać z niej w praktyce.

Zadanie 9.7. Napisz funkcję, która rozwiązuje układ 2 równań liniowych korzystając z metody Cramera. Poprawnie identyfikuj przypadki, w których dany układ nie jest oznaczony.

Zadanie 9.8. Napisz funkcję, która rozwiązuje układ 3 równań liniowych korzystając z metody Cramera. Poprawnie identyfikuj przypadki, w których dany układ nie jest oznaczony.

Zadanie 9.9. Dana jest macierz A o wartościach całkowitych. Napisz funkcję, która zwróci jej transpozycję.

Zadanie 9.10. Dana jest macierz A typu $n \times m$ o wartościach całkowitych. Napisz funkcję, która dla danego $0 \leqslant i < n$ i $0 \leqslant j < m$ zwróci podmacierz powstałą przez usunięcie z A i-tego wiersza i j-tej kolumny.

Zadanie 9.11. Dana jest kwadratowa macierz A o wartościach całkowitych. Napisz funkcję, która sprawdzi, czy macierz jest symetryczna. Zwróć wynik typu **bool**.

Zadanie 9.12. Dana jest macierz kwadratowa A o wartościach rzeczywistych typu $n \times n$. Napisz funkcję, która zwróci jej ślad, określony jako

$$\operatorname{tr}(A) = a_{11} + a_{22} + \cdots + a_{nn} = \sum_{i=1}^{n} a_{ii}.$$

Zadanie 9.13. Dla danej macierzy kwadratowej A napisz funkcję, która zwróci jej diagonalę w postaci tablicy jednowymiarowej.

Zadanie 9.14. Dla danej macierzy kwadratowej A napisz funkcję, która zwróci jej macierz diagonalną, czyli macierz z wyzerowanymi wszystkimi elementami poza przekątną.

Zadanie 9.15. Kwadratem łacińskim stopnia n nazywamy macierz kwadratową typu $n \times n$ o elementach ze zbioru $\{1, 2, \ldots, n\}$ taką, że żaden wiersz ani żadna kolumna nie zawierają dwóch takich samych wartości. Napisz funkcję, która sprawdza, czy dana macierz jest kwadratem łacińskim. Zwróć wynik typu **bool**.

Zadanie 9.16. Kwadratem magicznym stopnia n nazywamy macierz kwadratową typu $n \times n$ o elementach ze zbioru liczb naturalnych taką, że sumy elementów w każdym wierszu, w każdej kolumnie i na każdej z dwóch przekątnych są takie same. Napisz funkcję, która sprawdza, czy dana macierz jest kwadratem magicznym. Zwróć wynik typu **bool**.

Zadanie 10.1. Napisz samodzielnie pełny program w języku C++, który implementuje i testuje (w funkcji *main*()) następujące operacje na liście jednokierunkowej przechowującej wartości typu **double**:

a) Wyznaczenie sumy wartości wszystkich elementów.
b) Wyznaczenie sumy wartości co drugiego elementu.
c) Wyszukiwanie danego elementu.
d) Wstawienie elementu na początek listy.
e) Wstawienie elementu na koniec listy.
f) Wstawienie elementu na i-tą pozycję listy.
g) Usuwanie elementu z początku listy. Usuwany element jest zwracany przez funkcję.
h) Usuwanie elementu z końca listy. Usuwany element jest zwracany przez funkcję.
i) Usuwanie elementu o zadanej wartości. Zwracana jest wartość logiczna w zależności od tego, czy element znajdował się na liście, czy nie.
j) Usuwanie i-tego w kolejności elementu. Usuwany element jest zwracany przez funkcję.

Zadanie 10.2. Rozwiąż powyższe zadanie, implementując listę jednokierunkową, która dodatkowo przechowuje wskaźnik na ostatni element.

Zadanie 10.3. Rozwiąż powyższe zadanie, implementując listę dwukierunkową.

Zadanie 10.4. Dla danych dwóch list jednokierunkowych napisz funkcję, która je połączy, np. dla (1,2,5,4) oraz (3,2,5) sprawi, że I lista będzie postaci (1,2,5,4,3,2,5), a II zostanie skasowana.

Zadanie 10.5. Napisz samodzielnie pełny program w języku C++, który implementuje i testuje (w funkcji *main*()) stos (LIFO) zawierający dane typu **int**.

Zadanie 10.6. Napisz samodzielnie pełny program w języku C++, który implementuje i testuje (w funkcji *main*()) zwykłą kolejkę (FIFO) zawierającą dane typu **char**$*$ (napisy).

$\star$ **Zadanie 10.7.** Zaimplementuj operacje *enqueue*() i *dequeue*() zwykłej kolejki (FIFO) typu **int** korzystając tylko z dwóch gotowych stosów.

Zadanie 10.8. Napisz samodzielnie pełny program w języku C++, który implementuje i testuje (w funkcji *main*()) kolejkę priorytetową zawierającą dane typu **int**.

Zadanie 10.9. Napisz funkcję, która wykorzysta kolejkę priorytetową do posortowania danej tablicy o elementach typu **int**.

Zadanie 10.10. Napisz samodzielnie pełny program w języku C++, który implementuje i testuje (w funkcji *main*()) następujące operacje na drzewie binarnym przechowującym wartości typu **double**:

a) Wyszukiwanie danego elementu.
b) Zwrócenie elementu najmniejszego.
c) Zwrócenie elementu największego.
d) Wypisanie wszystkich elementów w kolejności od najmniejszego do największego.
e) Wstawianie danego elementu. Jeśli wstawiany element znajduje się już w drzewie nie należy wstawiać jego duplikatu.
f) Usuwanie danego elementu. Usuwany element jest zwracany przez funkcję.