

Zadania

1. Napisz program, który wyświetla na ekranie komunikat *Witaj*.
2. Napisz program przeliczający wielkość kąta wyrażoną w stopniach na radiany według wzoru

$$Rad = St \cdot \frac{\pi}{180}.$$

Liczba π ma w języku C++ nazwę **M_PI**. Aby skorzystać z tej stałej, należy użyć dyrektyw

```
#define _USE_MATH_DEFINES
#include <math.h>
```

3. Napisz program przeliczający temperaturę wyrażoną w stopniach Celsjusza na temperaturę wyrażoną w skali Fahrenheita według wzoru

$$F = 32 + 1.8 \cdot C.$$

4. Napisz program wyznaczający wartość funkcji

$$f(x) = x^5 + 3 \cos x + e^{4x-7}$$

w zadanym punkcie.

5. Napisz program wyznaczający wartość funkcji

$$f(x) = (ax + b)(c + d)$$

w zadanym punkcie. Wartości współczynników a, b, c, d należy wczytać z klawiatury.

6. Napisz program, który pobiera liczbę całkowitą i oblicza jej pierwiastek. Wykorzystaj funkcję `sqrt()`.

7. Napisz program, który pobiera dwie liczby całkowite i sumuje je tylko w przypadku, gdy obie są dodatnie.

8. Napisz program, który pobiera liczbę rzeczywistą i wyświetla jedną z poniższych informacji:

- Wprowadzona liczba jest dodatnia.
- Wprowadzona liczba jest ujemna.
- Wprowadzona liczba jest równa zero.

9. Napisz program znajdujący pierwiastki trójmianu kwadratowego $ax^2 + bx + c = 0$ dla zadanych parametrów a, b, c .

10. Napisz program, który wyznacza rozwiązanie układu równań liniowych

$$\begin{cases} ax + by = e \\ cx + dy = f. \end{cases}$$

Skorzystaj ze wzorów Cramera. Wartości współczynników a, b, c, d, e, f należy wczytać z klawiatury.

11. Napisz program, który spośród trzech różnych liczb a, b, c wybiera największą.

12. Napisz program, który wczytuje trzy liczby całkowite a następnie sprawdza, czy stanowią one trójkę pitagorejską, to znaczy czy suma kwadratów dwóch mniejszych liczb jest równa kwadratowi liczby największej.

Przykład: liczby 3, 4, 5 stanowią trójkę pitagorejską, ponieważ $3^2 + 4^2 = 5^2$.

Uwaga: W programie należy założyć, że użytkownik wpisze liczby w dowolnej kolejności, np. 5, 3, 4.

13. Napisz program, który wprowadzoną ocenę wypisuje słownie:

2 - niedostateczny,

3 - dostateczny,

4 - dobry,

5 - bardzo dobry,

6 - celujący.

Wykorzystaj instrukcję switch.

14. Napisz program kalkulator, który wykonuje wybraną przez użytkownika operację arytmetyczną (dodawanie, odejmowanie, mnożenie, dzielenie) na dwóch wczytanych liczbach. Do wyboru działania użyj instrukcji switch.

15. Napisz program, który w zależności od wyboru użytkownika wczytuje wymiary: kwadratu, prostokąta lub trójkąta i wypisuje pole figury o wczytanych wymiarach. Do wyboru figury użyj instrukcji switch.

16. Napisz program obliczający tygodniowe zarobki brutto i netto pracownika, gdy znana jest jego kategoria zaszeregowania i liczba przepracowanych godzin w ciągu tygodnia. Dane te należy wczytać z klawiatury. Wykorzystaj instrukcję switch.

Dla poszczególnych kategorii zaszeregowania obowiązują następujące stawki:

Kategoria zaszeregowania	Stawka [zł/godz.]
A	15
B	25
C	30
D	35

Jeśli pracownik przepracuje więcej niż 40 godzin w ciągu tygodnia, zapłata za każdą nadgodzinę jest dwukrotnie wyższa od stawki pracownika. Podatek od zarobku jest obliczany według następującej tabeli:

Zarobek	Stopa procentowa podatku
≤ 700	15%
700 – 1200	20%
> 1200	25%

17. Napisz program obliczający wielkość wygranej przy założeniu, że gracz rzuca kostką dwukrotnie. Jeśli p i q oznaczają odpowiednio wyniki pierwszego i drugiego rzutu, wygrana jest określona według tabeli:

p	q	wygrana
parzyste	2 lub 4 lub 5	$p + 3q$
	1 lub 3 lub 6	$2q$
nieparzyste	1 lub 3 lub 6	$5p + 3$, gdy $p = q$ $2q + p$, gdy $p \neq q$
	2 lub 4 lub 5	$\min(p, q) + 4$

Ponadto, jeśli wynikami obu rzutów jest 5, gracz otrzymuje premię w postaci dodatkowych 5 punktów.

Program napisz w dwóch wersjach.

[1] W wersji pierwszej wyniki rzutów wczytuj z klawiatury.

[2] W wersji drugiej wyniki rzutów powinny być losowane. Skorzystaj z funkcji

- `srand(time(NULL))` – inicjowanie generatora liczb pseudolosowych
- `rand()%6 + 1` – wylosowanie liczby ze zbioru $\{1, 2, 3, 4, 5, 6\}$

oraz dołącz dyrektywy

```
#include <cstdlib>
```

```
#include <ctime>
```

Pętle

18. Napisz program wypisujący na ekranie wszystkie liczby:

1. dwucyfrowe;
2. trzycyfrowe podzielne przez 13.

19. Napisz program wypisujący wszystkie dzielniki liczby naturalnej n w trzech wersjach:

1. z pętlą *for*,
2. z pętlą *while*,
3. z pętlą *do...while*.

20. Napisz program wyznaczający sumę kwadratów liczb naturalnych od 1 do n . Liczbę n należy wczytać z klawiatury.

21. Napisz program sprawdzający, czy wczytana z klawiatury liczba n jest pierwsza.

22. Napisz program obliczający, ile różnych podzbiorów k -elementowych można utworzyć ze zbioru n -elementowego. Liczby k i n należy wczytać z klawiatury. Wykorzystaj wzór

$$\frac{n!}{k!(n-k)!}$$

23. Napisz program obliczający sumę n ułamków postaci $\frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}, \dots$

Liczbę n należy wczytać z klawiatury.

Zwróć uwagę na działanie operatora dzielenia dla różnych typów argumentów, np:

- `int i = 7; float j = i/2; //zmienna j ma wartość 3`
- `float i = 7; float j = i/2; //zmienna j ma wartość 3.5`

Przykład dla $n = 5$: $\frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \frac{4}{5} + \frac{5}{6} = 3.55$

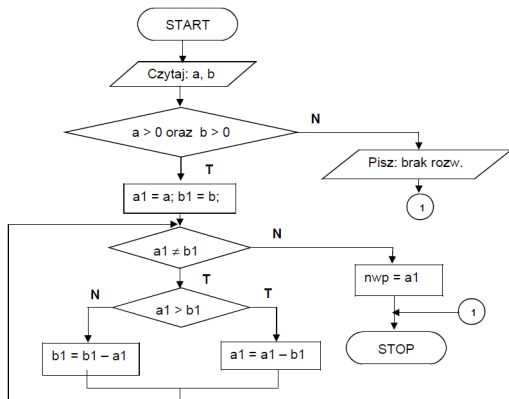
24. Napisz program sumujący losowe liczby naturalne z przedziału $[0, 20]$ do momentu, gdy kolejna wylosowana liczba będzie taka sama jak poprzednia. Zadanie zrealizuj w dwóch wersjach:

1. najpierw z pętlą *do...while*,
2. następnie z pętlą *while*.

Zastanów się nad różnicami między napisanymi wersjami.

25. Napisz program wyznaczający n -tą potęgę liczby x . Liczby n oraz x należy wczytać z klawiatury. Zadanie zrealizuj z wykorzystaniem wybranej pętli. Nie stosuj gotowych funkcji (np. `pow`).

26. Napisz program obliczający największy wspólny dzielnik dwóch liczb całkowitych dodatnich a i b . Liczby a i b należy wczytać z klawiatury. Przeanalizuj, a następnie wykorzystaj poniższy schemat blokowy.

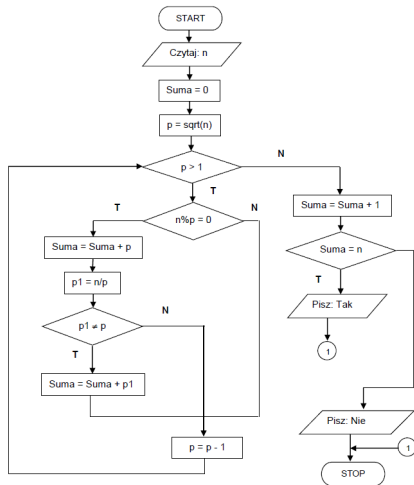


27. Napisz program obliczający najmniejszą wspólną wielokrotność dwóch liczb całkowitych dodatnich a i b . Liczby a i b należy wczytać z klawiatury.

28. Napisz program sprawdzający, czy wczytana z klawiatury dodatnia liczba całkowita n jest liczbą doskonałą. Liczba doskonała to taka, której suma dzielników (nie licząc samej liczby n) jest równa tej liczbie. Przykładami liczb doskonałych są liczby 6 i 28.

Przeanalizuj poniższy schemat blokowy i odpowiedz na pytania:

- W jaki sposób znajdowane są dzielniki danej liczby?
- Jakie zalety ma przedstawiony sposób znajdowania dzielników? Porównaj go ze swoim rozwiązaniem zadania 19.



29. Napisz program rysujący trójkąt, którego wysokość h należy wczytać z klawiatury. Przykład dla $h = 5$:

```
x
xxx
xxxxx
xxxxxxx
xxxxxxxxx
```

30. Napisz program wypisujący na ekranie liczby według poniższego przykładu dla $h = 5$. Wartość h należy wczytać z klawiatury.

```
1
2 4
3 6 9
4 8 12 16
5 10 15 20 25
```

31. Napisz program wyszukujący liczby pierwsze z przedziału $[2 \dots n]$. Wartość n wczytaj z klawiatury.

Przykład dla $n = 20$: 2 3 5 7 11 13 17 19.

Tablice

W każdym z poniższych zadań utwórz tablicę 100-elementową oraz pobierz z klawiatury rzeczywisty rozmiar $n \leq 100$. Wypełnij tablicę n losowymi liczbami typu `int` z przedziału $[a, b]$, gdzie liczby a i b podaje użytkownik.

Przykład 1: po wprowadzeniu $a = 0$, $b = 9$, $n = 10$ program powinien wypełnić tablicę 10 losowymi liczbami całkowitymi z przedziału $[0, 9]$, np. 6 1 3 0 3 2 2 5 3 0.

32. Napisz program, który oblicza średnią arytmetyczną elementów tablicy.

Przykład: Dla tablicy z przykładu 1 średnia jest równa 2.5.

33. Napisz program, który pobiera od użytkownika liczbę x typu `int` a następnie podaje, ile razy wystąpiła ona w tablicy.

Przykład: dla tablicy z przykładu 1 i $x = 0$ wynikiem działania programu jest 2, ponieważ zero w tablicy występuje dwukrotnie.

34. Napisz program, który podaje, ile liczb nieparzystych jest w tablicy oraz ich sumę.

Przykład: dla tablicy z przykładu 1 liczba elementów nieparzystych: 5, ich suma: 15.

35. Napisz program, który elementy tablicy zastępuje ich potęgami stopnia drugiego.

Przykład: dla tablicy z przykładu 1 wynikiem jest: 36 1 9 0 9 4 4 25 9 0.

36. Napisz program, który wypisuje elementy tablicy o wartościach z przedziału $[c, d]$. Parametry c, d podaje użytkownik.

Przykład: dla tablicy 6 1 3 0 3 2 2 5 3 0, $c = 3$, $d = 8$; wynik: 6 3 3 5 3.

37. Napisz program, który wypisuje elementy o indeksach podzielnych przez liczbę p typu int, wprowadzaną przez użytkownika.

Przykład: dla tablicy 6 1 3 0 3 2 2 5 3 0, $p = 3$; wynik: 0 2 0.

38. Napisz program, który podaje element minimalny tablicy oraz jego indeks.

Przykład: dla tablicy 6 1 3 0 3 2 2 5 3 0 element minimalny: 0, jego indeks: 3 (pierwsze wystąpienie w tablicy).

39. Napisz program, który wyznacza parę sąsiednich elementów tablicy o największej sumie.

Przykład: dla tablicy 6 1 3 0 3 2 2 5 3 0 parą taką są elementy 5 i 3.

Tablice 2-wymiarowe

W każdym z zadań utwórz tablicę 10×10 oraz pobierz z klawiatury rzeczywisty wymiar wierszowy $n \leq 10$ i kolumnowy $m \leq 10$. Wypełnij tablicę losowymi liczbami typu int z przedziału $[0, 9]$ oraz wyświetl macierz.

Przykład 2: po wprowadzeniu $n = 5$, $m = 4$ program powinien wypełnić tablicę 5×4 losowymi liczbami całkowitymi z przedziału $[0, 9]$, np:

```
5 1 8 3
7 0 3 0
8 4 6 2
0 7 2 1
4 3 5 5
```

40. Napisz program, który znajdzie element maksymalny macierzy oraz jego położenie (indeks wierszowy i kolumnowy).

Przykład: dla tablicy z przykładu 2:

element maksymalny = 8, położenie: wiersz = 0, kolumna = 2 (pierwsze wystąpienie).

41. Napisz program, który sprawdzi, czy dana macierz kwadratowa jest trójkątna dolna, tzn. czy wszystkie jej elementy poniżej przekątnej głównej są równe zero, np.

```
5 1 8 3
0 7 3 2
0 0 6 2
0 0 0 1
```

42. Napisz program, który zamieni w macierzy wiersz k -ty z wierszem p -tym. Indeksy k i p należy wczytać z klawiatury.

Przykład: dla tablicy z przykładu 2, $k = 1$ i $p = 3$:

```
5 1 8 3
0 7 2 1
8 4 6 2
7 0 3 0
4 3 5 5
```

43. Napisz program, który w tablicy kwadratowej (tzn. $n = m$) zamieni elementy leżące na przekątnych, np:

```
5 1 8 3    3 1 8 5
7 0 3 0    7 3 0 0
8 4 6 2    8 6 4 2
0 7 2 1    1 7 2 0
```

Dodatkowe 1. Napisz program, który wyznacza drugi co do wielkości element tablicy.

Przykład: dla tablicy 6 1 3 0 3 2 2 5 3 0 takim elementem jest 0 lub 2.

Dodatkowe 2. Napisz program, który wyznacza element najczęściej występujący w tablicy.

Przykład: dla tablicy 6 1 3 0 3 2 2 5 3 0 takim elementem jest 3.

Dodatkowe 3. Napisz program, który oblicza iloczyn skalarny dwóch wektorów składających się z elementów całkowitych. Maksymalny rozmiar wektorów wynosi 10, rzeczywisty rozmiar n ($n \leq 10$) wczytaj z klawiatury.

Przykład: dla $n = 3$, $x = (1, 2, 3)$, $y = (4, 5, 6)$ iloczyn skalarny wektorów x i y jest równy $(1 \cdot 4) + (2 \cdot 5) + (3 \cdot 6) = 32$.

Dodatkowe 4. Napisz program, który dla macierzy $A_{n \times m}$ tworzy macierz transponowaną $A_{m \times n}^T$.

Przykład: Dla macierzy

5 1 8 3

7 0 3 0

macierzą transponowaną jest

5 7

1 0

8 3

3 0

Funkcje

44. Dla zadanej liczby rzeczywistej x poniższy program oblicza wartość wyrażenia

$$\sqrt[4]{x^2 + (x + 3)^2} + 5\sqrt[4]{(3x)^2 + 100}.$$

```
#include <iostream>
using namespace std;
int main()
{
    float x, wynik;
    cout << "Podaj x: ";
    cin >> x;
    wynik = sqrt(sqrt(x*x + (x + 3)*(x + 3)))
            + 5 * sqrt(sqrt((3 * x)*(3 * x) + 100));
    cout << "Wartosc wyrazenia to " << wynik;
}
```

Zmodyfikuj program, wprowadzając funkcję

```
float F(float a, float b)
{
    float p = sqrt(sqrt(a*a + b*b));
    return p;
}
```

W funkcji `main` wykorzystaj funkcję `F` do obliczenia wartości podanego wyrażenia.

45. Napisz funkcję o nazwie `iloczyn`, która mnoży dwie liczby typu `int`, a następnie przetestuj jej działanie w funkcji `main`. Zadanie wykonaj w czterech wersjach (wspomóż się przykładami z wykładu):

1. z funkcją o prototypie `int iloczyn(int a, int b)`
2. z funkcją o prototypie `int iloczyn()`
3. z funkcją o prototypie `void iloczyn(int a, int b)`
4. z funkcją o prototypie `void iloczyn()`

46. Dany jest program, który pobiera od użytkownika liczby całkowite a , b , c i n , następnie oblicza wartość wyrażenia

$$\frac{a^n + b^{n+5}}{c^{2n+1}}.$$

Kod programu znajdziesz na następnej stronie.

```

#include <iostream>
using namespace std;
int main()
{
    float a, b, c, n;
    cout << "Podaj kolejno liczby a, b, c, n: ";
    cin >> a >> b >> c >> n;
    float a1 = 1;
    for (int i = 1; i <= n; i++)
    {
        a1 = a1 * a;
    }
    float b1 = 1;
    for (int i = 1; i <= n+5 ; i++)
    {
        b1 = b1 * b;
    }
    float c1 = 1;
    for (int i = 1; i <= 2*n+1; i++)
    {
        c1 = c1 * c;
    }
    float wynik = (a1 + b1)/c1;
    cout << "Wartosc wyrazenia to " << wynik;
}

```

Zmodyfikuj program, wprowadzając funkcję o nazwie *potega*, która podnosi daną liczbę do zadanej potęgi (nie używaj gotowych funkcji, np. *pow*). Napisaną funkcję wykorzystaj w funkcji *main* do obliczenia wartości podanego wyrażenia.

47. Napisz program wypisujący wszystkie liczby pierwsze z przedziału $[2, n]$. Liczbę n podaje użytkownik. W programie powinna znaleźć się funkcja

`bool pierwsza(int p),`

która zwróci w wyniku wartość *true*, jeżeli sprawdzana liczba p jest pierwsza i wartość *false* w przeciwnym przypadku.

48. Przepisz i przetestuj poniższe programy. Jaka jest różnica w ich działaniu i z czego ona wynika?

```
#include <iostream>
using namespace std;
void F(int &i);
int main()
{
    int n = 2;
    cout << n << endl;
    F(n);
    cout << n;
}
void F(int &i)
{
    i = i + 5;
}
```

```
#include <iostream>
using namespace std;
void F(int i);
int main()
{
    int n = 2;
    cout << n << endl;
    F(n);
    cout << n;
}
void F(int i)
{
    i = i + 5;
}
```

49. Napisz funkcję, która otrzymuje dwa parametry a , b typu *int* i pierwszy z nich zwiększa dwukrotnie, a do drugiego dodaje 100. Przetestuj jej działanie w funkcji `main`.

50. Napisz funkcję, której parametrami są liczby oznaczające długość, szerokość i wysokość prostopadłościanu a wynikiem jest pole powierzchni ścian oraz objętość.

Pamiętaj, że funkcja nie może zwrócić dwóch wartości jednocześnie (za pomocą *return*). Jeden z wyników (np. pole) zwróć zatem za pomocą *return* a drugi (objętość) w parametrze przekazywanym przez adres. Prototyp takiej funkcji:

```
float prost(float dlug, float szer, float wys, float &objetosc);
```

Przetestuj działanie tej funkcji w funkcji `main`.

Zmodyfikuj funkcję tak, aby dodatkowo zwracała -1 w przypadku, gdy długość, szerokość lub wysokość jest ujemna. W tym celu użyj dwukrotnie *return*;

51. Napisz funkcję `max2`, która zwraca większą z dwóch podanych jako jej parametry liczb, a następnie funkcję `max3`, zwracającą największą spośród trzech przekazanych w parametrach liczb. Funkcja `max3` ma do tego celu używać funkcji `max2`.

52. Czy poniższy program działa poprawnie? Jeżeli nie, znajdź błąd i napraw go.

```
#include<iostream>
#include<ctime>
#include<conio.h>
using namespace std;
void dane(int n, int tab []); //wpisuje do tablicy losowe liczby int
void wyswietl(int n, int tab []); //wyswietla zawartosc tablicy
int main()
{
    int X[10], Y[10], n;
    srand(time(NULL));
    dane(n, X);
    wyswietl(n, X);
    dane(n, Y);
    wyswietl(n, Y);
    _getch();
}
void dane(int n, int tab [])
{
    cout << "Podaj rozmiar wektora (n<=10) ";
    cin >> n;
    for (int i = 0; i <= n - 1; i++)
        tab[i] = rand() % 10;
}
void wyswietl(int n, int tab [])
{
    for (int i = 0; i <= n - 1; i++)
    {
        cout << tab[i] << " ";
    }
    cout << endl;
}
```

53. Napisz funkcję, która oblicza iloczyn skalarny dwóch wektorów całkowitoliczbowych:

```
int skalarny(int n, int X[], int Y[]);
```

Przetestuj jej działanie w funkcji main. Do wypełniania i wypisywania wektorów użyj funkcji dane i wyswietl z poprzedniego zadania.

Przykład: dla $n = 3$, $X = (1, 2, 3)$, $Y = (4, 5, 6)$ iloczyn skalarny wektorów X i Y jest równy $(1 \cdot 4) + (2 \cdot 5) + (3 \cdot 6) = 32$.

54. Napisz funkcję o prototypie

```
int przekatna(int n, int tab[10][10]);
```

która oblicza sumę elementów na przekątnej głównej macierzy kwadratowej tab. Maksymalny rozmiar macierzy to 10x10, n oznacza rozmiar bieżący ($n \leq 10$).

Przykład: Dla $n = 4$ i poniższej macierzy suma jest równa 8.

```
5 1 2 1
4 0 8 8
7 5 2 4
0 3 7 1
```

W programie powinny znaleźć się również funkcje

- void dane(int n, int tab[10][10]) - wpisująca do macierzy losowe liczby całkowite,
- void wyswietl(int n, int tab[10][10]) - wyświetlająca zawartość macierzy.

Wskaźniki

55. Rozwiąż zadania z pliku *Wskaźniki_wstęp.pdf*.

56. W poniższym programie zdefiniuj pięć wskaźników i ustaw je odpowiednio na zmienne n , k , sn , sk i snk . Następnie zmodyfikuj kod tak, aby wszystkie odwołania do zmiennych odbywały za pomocą wskaźników.

```
#include <iostream>
using namespace std;
int main ()
{
    int n, k;
    cout << "Podaj kolejno n i k:" << endl;    cin >> n >> k;

    int sn = 1;
    for (int i = 2; i <= n; i++)
        sn = sn * i;

    int sk = 1;
    for (int i = 2; i <= k; i++)
        sk = sk * i;

    int snk = 1;
    for (int i = 2; i <= n-k; i++)
        snk = snk * i;

    cout << "Symbol Newtona n nad k wynosi " << sn / (sk * snk);
}
```

57. Przeanalizuj poniższy kod, a następnie uzupełnij go tak, aby wyświetlał najmniejszy element tablicy. Do elementów tablicy i zmiennej `min` odwołuj się przy pomocy wskaźników.

```
#include <iostream>
#include <ctime>
using namespace std;
int main()
{
    int tab[10];
    int n;
    int *wtab = tab;    //wskaznik na poczatek tablicy tab
    int *wn = &n;       //wskaznik na zmienna n
    cout << "Podaj rozmiar tablicy (n<=10): ";
    cin >> *wn;
    srand(time(NULL));
    for (int i = 0; i < *wn; i++)
    {
        *wtab = rand()%10;
        wtab++;        //przesuniecie wskaznika na nastepny element tablicy
    }
    wtab = tab;        //powrot wskaznika na poczatek tablicy
    for (int i = 0; i < *wn; i++)
        cout << *(wtab + i) << " ";    //tutaj wskaznika nie przesuwamy
    int *min;
    min = &tab[0];
    //uzupelnij
    cout << endl << "Minimalny element tablicy to " << *min;
}
```


58. Napisz program, który oblicza średnią arytmetyczną elementów leżących pomiędzy elementem minimalnym i maksymalnym w tablicy liczb całkowitych. Przyjmij, że maksymalny rozmiar tablicy to 20. Zadanie należy rozwiązać w oparciu o wskaźniki.

Przykład 1: dla tablicy: 2 5 6 2 3 2 2 0 1 0

min = 0, jego indeks = 7

max = 6, jego indeks = 2

średnia arytmetyczna elementów: 6 2 3 2 2 0 jest równa $15/6 = 2.5$.

59. Zastanów się nad różnicą między funkcjami f i g, a następnie przetestuj program.

```
#include <iostream>
int f(int *a, int *b);
int* g(int *a, int *b);
int main()
{
    //uzupełnij
}
int f(int *a, int *b)
{
    if (*a < *b) return *a;
    else return *b;
}
int* g(int *a, int *b)
{
    if (*a < *b) return a;
    else return b;
}
```

60. Napisz funkcję, która oblicza iloczyn skalarny dwóch wektorów całkowitoliczbowych:

```
int skalarny(int *wsk_n, int *wsk_X, int *wsk_Y);
```

*wsk_n - wskaźnik na zmienną n, tzn na rozmiar tablicy

*wsk_X - wskaźnik na tablicę liczb całkowitych

*wsk_Y - wskaźnik na tablicę liczb całkowitych

Przyjmij, że maksymalny rozmiar wektorów to 10. W programie powinny znaleźć się również funkcje

- void dane(int *wsk_n, int *wsk_tab) - wczytująca bieżący rozmiar tablicy (≤ 10) i wypełniająca tablicę liczbami losowymi,
- void wyswietl(int *wsk_n, int *wsk_tab) - wypisująca zawartość tablicy.

61. Napisz funkcję odwracającą kolejność pierwszych n elementów w tablicy liczb typu int. Wartością zwracaną funkcji powinien być adres pierwszego elementu tablicy tak, aby możliwe było wywołanie

```
odwracanie(odwracanie(tablica,n),n).
```

Takie wywołanie powinno w efekcie dać tablicę z elementami ułożonymi w pierwotnej kolejności.

62. Zdefiniuj następujący typ strukturalny:

```
struct uczen
{
    string imie, nazwisko;
    float mat, fiz, inf;
};
```

Zdefiniuj tablicę 10-elementową o nazwie dane, przechowującą oceny uczniów z matematyki, fizyki i informatyki, oraz następujące funkcje

- void wczytaj_dane(int &n, uczen dane[10]);
wczytuje z klawiatury bieżący rozmiar tablicy ($n \leq 10$) i jej elementy,
- void wyswietl_dane(int n, uczen dane[10]);
wypisuje zawartość tablicy dane,
- void srednia(int n, uczen dane[10], float &sr_min, float &sr_max, int &i_min, int &i_max);
oblicza średnią ocen z przedmiotów dla każdego ucznia oraz znajduje osobę z najniższą i osobę najwyższą średnią,
- void wyswietl_sr(uczen dane[10], float sr_min, float sr_max, int i_min, int i_max);
wyświetla ucznia z najniższą i ucznia z najwyższą średnią.

Przetestuj napisane funkcje w funkcji main.

63. Zdefiniuj typ strukturalny o nazwie `trojkat` przechowujący długości boków trójkąta jako zmienne typu `float`. Następnie napisz funkcję, która otrzymuje jako argument zmienną typu `trojkat` i zwraca jako wartość obwód trójkąta przekazanego w argumentcie:

```
float obwod(trojkat t);
```

Przetestuj tę funkcję w funkcji `main`.

64. Zdefiniuj typ strukturalny o nazwie `ulamek` przechowujący licznik i mianownik ułamka (jako zmienne typu `int`) oraz wartość ułamka (jako zmienną typu `float`). Następnie napisz funkcję, która otrzymuje dwa argumenty typu `ulamek` i zwraca nową zmienną typu `ulamek`, będącą wynikiem mnożenia ułamków przekazanych w argumentach:

```
ulamek mnozenie(ulamek a, ulamek b);
```

Przetestuj tę funkcję w funkcji `main`.