

Tablice jako parametry funkcji

1. Przykład

Oceny punktowe ze sprawdzianu reprezentuje n -elementowy ciąg liczb dodatnich. Poniżej zapisano program, w którym wyznaczana jest liczba studentów posiadających ocenę powyżej średniej, zawierający funkcje:

- **printData()** – wyświetlanie oceny studentów,
- **readData()** – wprowadzanie liczby studentów i ich ocen,
- **average()** – wyznaczanie średniej ocen (wynik przekazywany przez parametr),
- **countAbove()** – określanie liczby ocen większych niż parametr (wynik zwracany przez funkcję).

```
#include <iostream>

using namespace std;

void readData(const char msg[], int *n, float tab[])
// void readData(const char *msg, int *n, float *tab)
{
    cout << msg << endl;
    cin >> *n;

    for(int i = 0; i < *n; i++)
    {
        cout << "Wartosc " << i + 1 << ": ";
        cin >> tab[i]; // cin >> *(tab + i);
    }
}

void printData(const char msg[], int n, float tab[])
{
    cout << msg << endl;

    for(int i = 0; i < n; i++)
        cout << tab[i] << endl;
}

void average(int n, float tab[], float *result)
{
    float sum = 0;
    for(int i = 0; i < n; i++)
        sum += tab[i];
    *result = sum / n;
}

int countAbove(int n, float tab[], float limit)
{

```

```
int cnt = 0;
for(int i = 0; i < n; i++)
    if(tab[i] > limit) cnt++;
return cnt;
}

int main()
{
    float grades[100], avg;
    int n;

    readData("Podaj liczbe ocen i oceny.", &n, grades);
    printData("Oceny punktowe studentow.", n, grades);

    average(n, grades, &avg);
    cout << "Srednia: " << avg << "." << endl;

    cout << "Powyzej sredniej: " << countAbove(n, grades, avg) << " osob." << endl;

    return 0;
}
```

- Przepisz i uruchom program.
- Przetestuj działanie programu po zmianie odwołań do elementów tablic na wskaźnikowe (np.: `x[i]` na `*(x + i)`).

2. Zadanie

Uzupełnij program z przykładu 1 o:

- funkcję obliczającą procent studentów z oceną w przedziale $\langle 0,3 w; 0,7 w \rangle$, gdzie w to parametr funkcji. Wywołaj funkcję z parametrem równym wartości oceny średniej,
- funkcję obliczającą minimalną i maksymalną ocenę studenta,
- funkcję wyznaczającą tablicę zawierającą stopnie studentów (skala punktów od 0 pkt. do 40 pkt.):
poniżej 20 pkt. – 2.0, 20 pkt. – 3.0, 24 pkt. – 3.5, 28 pkt. – 4.0, 32 pkt. – 4.5, 36 pkt. i więcej – 5.0.

3. Zadanie

Dwa n -elementowe ciągi liczb $\{x_i\}$ i $\{y_i\}$ reprezentują współrzędne punktów na płaszczyźnie. Opracuj program zawierający funkcję:

- do wczytywania liczby punktów i ich współrzędnych,
- do wyświetlania na ekranie współrzędnych punktów z wybranej ćwiartki układu współrzędnych (nr ćwiartki będzie parametrem funkcji),
- wyznaczającą odległości poszczególnych punktów od początku układu współrzędnych,
- wyznaczającą punkt najdalej oddalony od początku układu współrzędnych,
- wyznaczającą dodatkowe tablice ze współrzędnymi punktów położonych w okręgu o środku w początku układu współrzędnych i promieniu r .

Zadania dodatkowe

1. Uzupełnij program z zadania 1 o funkcję wyznaczającą sześćelementową tablicę zawierającą liczby studentów, którzy uzyskali poszczególne stopnie (pierwszy element tablicy – liczba stopni 2.0, drugi element tablicy – liczba stopni 3.0 itd.).
2. Dany jest wielomian $w(x) = a_n x^n + \dots + a_2 x^2 + a_1 x^1 + a_0 x^0 = a_n x^n + \dots + a_2 x^2 + a_1 x + a_0$. Napisz program (używając funkcji) do:
 - wyznaczania wartości wielomianu według schematu **Hornera**, w znanej postaci $w(x) = (\dots((a_n x + a_{n-1})x + \dots + a_2)x + a_1)x + a_0$, korzystając z podejścia rekurencyjnego i iteracyjnego (samodzielnie),

```
int hornerRec(int coef[], int deg, int x)
{
    if(deg == 0)
        return coef[0];

    return x * hornerRec(coef, deg - 1, x) + coef[deg];
}
```

```
int hornerIter(int coef[], int deg, int x)
{
    ...
}
```

- wyznaczania pochodnej wielomianu $\frac{d}{dx} w(x)$,
- wyznaczenie wartości pochodnej wielomianu.