

Trójkot



```
#double #float
#spójnik_logiczny #koniunkcja
#ampersand &&
#cmath #sqrt
#wzór_Herona
```

Oprócz liczb całkowitych czasami musimy zająć się liczbami posiadającymi część ułamkową. Takie dane najlepiej reprezentować przy użyciu typu `double`, który jest przykładem typu *zmiennoprzecinkowego* (ang. **floating point**, wymawiaj: *flotting point*). Zaraz, zaraz, skoro mamy typ `double`, to powinien być też typ danych w rodzaju `single`. No tak, mamy jeszcze typ danych `float`, taki trochę mniej wypasiony w porównaniu z `double`. Ze wszech miar zalecamy jednak użycie typu `double`, na którym obliczenia wykonywane są znacznie dokładniej.

Napiszemy program wczytujący trzy liczby rzeczywiste dodatnie a, b, c i sprawdzający, czy może istnieć trójkąt o takich właśnie długościach boków. Jeśli jest to możliwe, wtedy program powinien wypisać komunikat **Possible** (z ang. *możliwe*, wymawiaj: *posibl* z akcentem na o), a następnie obliczyć jego obwód (**perimeter**) oraz pole powierzchni (**area**, wymawiaj: *eria*, z akcentem na e). Jeśli utworzenie trójkąta jest niemożliwe, program powinien wypisać komunikat **Impossible**.

Jeśli trójkąt istnieje, wtedy jego długości boków muszą spełniać warunki: $a + b > c$, $a + c > b$ oraz $b + c > a$ (tak zwany *warunek trójkąta*). Wszystkie te warunki powinny być jednocześnie spełnione, zatem pomiędzy nimi musimy wstawić spójnik logiczny „i”. W języku logiki matematycznej takie połączenie wyrażeń logicznych nazywamy *koniunkcją*, zaś w języku C++ oznaczamy je symbolem `&&`. Pojedynczy znak `&` nazywany jest z angielska **ampersand** i nie ma sensownej polskiej nazwy.

Pełny warunek trójkąta powinien wyglądać następująco:

```
(a + b > c) && (a + c > b) && (b + c > a)
```

Nawiasy nie są konieczne, ale poprawiają czytelność zapisu.

Obliczenie obwodu nie przedstawia problemu ($a + b + c$), ale co z polem? Chciałoby się zastosować podręcznikowy wzór $P = \frac{1}{2}ah$, ale nie znamy wysokości trójkąta (h). Na szczęście mądrzy ludzie wymyślili mnóstwo różnych wzorów na pole trójkąta, wśród których wyróżnia się wzór Herona,* w którym występują tylko długości boków trójkąta:

$$P = \sqrt{p(p-a)(p-b)(p-c)} \quad \text{gdzie} \quad p = \frac{a+b+c}{2}.$$

*Heron z Aleksandrii, genialny grecki matematyk, fizyk i wynalazca, żył najprawdopodobniej w I w. n.e., a wśród jego wynalazków należy wymienić pierwowzór turbiny parowej czy rozmaite maszyny oblężnicze.

Jeszcze tylko musimy ogarnąć, jak obliczyć pierwiastek kwadratowy (ang. **square root**, wymawiaj: *skłter rut*) – do tego służy funkcja `sqrt()` z biblioteki `cmath` (wymawiaj: *si-maθ*, z twardym *s*[†]). Teraz możemy już zmontować cały program:

```
#include <iostream>
#include <cmath>
using namespace std;

int main()
{
    double a, b, c;
    cin >> a >> b >> c;
    if((a + b > c) && (a + c > b) && (b + c > a))
    {
        cout << "Possible" << endl;
        cout << "Perimeter: " << a + b + c << endl;
        double p = (a + b + c) / 2;
        cout << "Area: " <<
            sqrt(p * (p - a) * (p - b) * (p - c)) << endl;
    }
    else
        cout << "Impossible" << endl;
}
```

Po uruchomieniu programu i wpisaniu przykładowych liczb 5, 6, 7 program powinien wypisać:

```
5 6 7
Possible
Perimeter: 18
Area: 14.6969
```

Natomiast po wpisaniu liczb 1, 2, 3 program powinien wypisać:

```
1 2 3
Impossible
```

[†]Symbol θ oznacza „seplenione” *s*.