

Marek Luliński & Gniewomir Sarbicki

**PYTHON**

**C++**

**JAVASCRIPT**

**ZADANIA Z PROGRAMOWANIA**



**Helion** 

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Opieka redakcyjna: Ewelina Burska

Projekt okładki: Studio Gravite/Olsztyn

Obarek, Pokoński, Pazdrijowski, Zaprucki

Materiały graficzne na okładce zostały wykorzystane za zgodą Shutterstock.

Wydawnictwo HELION

ul. Kościuszki 1c, 44-100 GLIWICE

tel. 32 231 22 19, 32 230 98 63

e-mail: [helion@helion.pl](mailto:helion@helion.pl)

WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/pycysz>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-283-3527-1

Copyright © Helion 2018

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

# Spis treści

<b>Wstęp</b> .....	<b>5</b>
<b>Wprowadzenie</b> .....	<b>7</b>
<b>Rozdział 1. Zadania</b> .....	<b>31</b>
<b>Rozdział 2. Rozwiązania</b> .....	<b>41</b>
<b>Rozdział 3. Dodatek</b> .....	<b>117</b>
T-komputer .....	117
<b>Rozdział 4. Trochę historii</b> .....	<b>121</b>
Programowanie z „myszką” .....	121
Zastosowanie tablic .....	123
Języki programowania .....	124
<b>Literatura</b> .....	<b>126</b>
<b>Słowniczek informatyczny</b> .....	<b>127</b>
<b>Skorowidz</b> .....	<b>135</b>



## C++

Język C++ został zaprojektowany przez Bjarne'a Stroustrupa jako następcę C, rozbudowany głównie o obiektowość. W fazie tworzenia język nazywał się roboczo „C z klasami”. W latach 90. zyskał dużą popularność zarówno w programowaniu systemowym, jak i użytkowym. C++ jest uznawany za jeden z najlepszych języków do programowania obiektowego. Programiści wybierają język C++, jeżeli wymagana jest najwyższa wydajność, bezpośredni dostęp do systemu i niezawodność. C++ wpłynął znacząco na późniejsze języki, jak Java czy PHP, które mają wiele wspólnych cech syntaktycznych.

## Struktura programu źródłowego

Plik źródłowy programu jest plikiem tekstowym z poleceniami i instrukcjami języka.

Plik z nadanym rozszerzeniem *c* zawiera program napisany w języku C, a plik z rozszerzeniem *cpp* to program źródłowy C++.

W wyniku kompilacji powstaje kod pośredni w pliku *o* lub *obj*, a następnie po konsolidacji plik wykonywalny.

```
#include .....: — dołączenie bibliotek
main()          — program główny
{              — początek programu
.....:        — instrukcje
}              — koniec programu
```

## Przykład programu

```
#include <iostream> // dołączenie biblioteki wejścia/wyjścia — „input/output”
using namespace std; // użycie przestrzeni nazw std — „stream display”
main() // program główny
{ // początek programu
    int a, b, m, n; // deklaracje zmiennych typu całkowitego
    system("CLS"); // Clear Screen — „czyszczenie” ekranu
    cout << " I liczba = "; // wyświetlenie tekstu na ekranie
    cin >> a; // wczytanie I liczby z klawiatury do zmiennej
    m = a; // podstawienie
    cout << " II liczba = "; // wyświetlenie tekstu na ekranie
    cin >> b; // wczytanie II liczby z klawiatury do zmiennej
    n = b; // podstawienie
    while(a != b) // powtarzanie czynności, „pętla dopóki”
        if (a > b) a = a - b; else b = b - a; // instrukcja warunkowa „jeżeli”
    cout << " NWD(" << m << ", " << n << ") = " << a << endl; // wyprowadzenie wyniku
    system("PAUSE"); // „przytrzymanie” ekranu — oczekiwanie na klawisz
} // koniec programu
```

## Przydatne biblioteki

- <iostream> — konieczna! operacje wejścia/wyjścia
- <cstring> — obsługa tekstów
- <cmath> — funkcje matematyczne
- <fstream> — operacje plikowe
- <sstream> — obsługa strumieni
- <iomanip> — umożliwia m.in. ustalanie w liczbach liczby miejsc po przecinku
- <cstdlib> — zawiera m.in. funkcje konwersji typów

## Przykład programu z funkcją

```
#include <iostream>
using namespace std;
// ..... PODPROGRAM .....
int euk(int a, int b)
{
    while(a != b)
        if (a > b) a = a - b; else b = b - a;
    return a;
}
// ..... PROGRAM GŁÓWNY .....
main()
{
    int m, n;
    system("CLS");
    cout<<" I liczba = ";
    cin>>m;
    cout<<" II liczba = ";
    cin>>n;
    cout<<" NWD(" << m << ", " << n << ") = " << euk(m, n) << endl;
    system("PAUSE");
}
```

## Kompilatory, edytory programistyczne i środowiska języka C++

### Windows

Dev-C++, C++ Builder, Code::Blocks, Turbo C++, Dev-C++, Eclipse, Watcom C++, NetBeans, Intel C++ Compiler, Magic C++, Ulitimate++, Borland C++, Microsoft Visual C++, Microsoft Visual Studio

### Linux

g++, Eclipse, Code::Blocks  
Kompilacja: g++ -o moje moje.cpp  
Uruchomienie: ./moje

## Typy liczbowe

### Liczby całkowite

Typ zmiennej	Zakres	Rozmiar pamięci [bajty]
char	0 – 255	1
wchar_t	0 – 65535	2
short int	-32768 – 32767	2
int	-2147483648 – 2147483647	2 lub 4
long int	-2147483648 – 2147483647	4
long long int	$-9,2 \cdot 10^{18}$ – $9,2 \cdot 10^{18}$	8

### Liczby rzeczywiste wymierne

Typ zmiennej	Zakres	Rozmiar pamięci [bajty]
float	$3,4 \cdot 10^{-38}$ – $3,4 \cdot 10^{38}$	4
double	$1,7 \cdot 10^{-308}$ – $1,7 \cdot 10^{308}$	8
long double	$3,4 \cdot 10^{-4932}$ – $1,1 \cdot 10^{4932}$	8, 10 lub 12

## Funkcje matematyczne

- abs — wartość bezwzględna
- fabs — wartość bezwzględna liczby zmiennoprzecinkowej
- sqrt — pierwiastek kwadratowy
- cbirt — pierwiastek trzeciego stopnia
- pow — potęgowanie
- exp — liczba  $e$  do potęgi
- log — logarytm naturalny
- log10 — logarytm o podstawie 10
- ceil — zaokrąglenie w górę
- floor — zaokrąglenie w dół
- round — zaokrąglenie matematyczne
- trunc — obcięcie części ułamkowej liczby
- sin — sinus
- cos — cosinus
- tan — tangens
- asin — arcus sinus
- acos — arcus cosinus
- atan — arcus tangens

## Instrukcja warunkowa

Podstawową instrukcją warunkową jest instrukcja `if-else`.

Instrukcja może mieć następujące składnie:

```
if(wyrażenie) instrukcja;
if(wyrażenie) instrukcja_1; else instrukcja_2;
if(wyrażenie) { instrukcje_1 } else { instrukcje_2 };
```

W wyrażeniach logicznych stosujemy operatory porównań `==`, `!=`, `<`, `>`, `<=`, `>=` oraz operatory logiczne `&&` (*and*) i `||` (*or*).

## Instrukcje iteracyjne

### 1. Pętla `for` („dla”)

```
for(wartość_początkowa: warunek: krok) {instrukcje};
```

Przebieg pętli odbywa się za pomocą zmiennej iteracyjnej. Najpierw jest jej nadawana wartość początkowa, następnie sprawdzany jest warunek logiczny i jeśli jest on prawdziwy, to wykonywana jest podana instrukcja, na koniec następuje zmiana zmiennej iteracyjnej (tzw. licznika pętli) i ponownie sprawdzany jest warunek itd.

### 2. Pętla `do-while` („wykonuj ... dopóki”)

```
do{instrukcje} while (warunek);
```

Stosujemy ją wówczas, gdy nie znamy dokładnej liczby powtórzeń, ale chcemy, aby podana instrukcja wykonała się chociaż jeden raz. Instrukcje są wykonywane tak długo, dopóki podany warunek jest prawdziwy.

### 3. Pętla `while` („dopóki”)

```
while(warunek) {instrukcje}
```

Stosujemy ją wówczas, gdy nie znamy dokładnej liczby powtórzeń, ale możliwe, że podana instrukcja nie będzie wykonana ani razu. Instrukcje są wykonywane tak długo, dopóki warunek jest prawdziwy.

W przypadku zastosowania kilku funkcji w programie należy pamiętać o tym, że przy braku ich deklaracji (prototypów) i wywoływaniu jednej funkcji przez drugą bardzo ważna jest kolejność zapisu ich definicji.

Dlatego zaleca się tworzenie nagłówków funkcji, wówczas kolejność ich zapisu będzie nieistotna.

Wyjaśnia do poniższy przykład.

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

//funkcja
int fun1()
{
    fun2(); // błąd kompilacji! „fun2” niezadeklarowane
    return 0;
}
//funkcja
int fun2()
{
```



```

        return 0;
    }
    //funkcja główna
    main()
    {
        return 0;
    }

```

## Przykład programu z deklaracjami funkcji

```

#include <iostream>
using namespace std;

// nagłówki funkcji (na końcu średnik!)
int fun1(); // dzięki temu
int fun2(); // kompilacja bez błędów

// funkcja
int fun1()
{
    fun2();
    return 0;
}
// funkcja
int fun2()
{
    return 0;
}
// funkcja główna
main()
{
    return 0;
}

```

W języku C++ istnieje możliwość kontrolowania strumienia wejściowego przy użyciu metod `cin.good()` oraz `cin.fail()`.

Należy też pamiętać, że liczby zmiennoprzecinkowe w zależności od typu są zapamiętywane zawsze z pewnym przybliżeniem, co może prowadzić do błędów w obliczeniach.

W poniższym przykładzie może pojawić się komunikat, że liczby `y` i `z` nie są równe!

```

double x, y, z;

x = 5.0;
y = 4.0 * x - 25.8;
z = -5.8;

if(z == y) cout << "z i y są równe";
else cout << "z oraz y NIE są równe!";

```



# Skorowidz

## A

algorytm Euklidesa, 51

## B

blok funkcji, 27

## C

C++, 12

edytory programistyczne, 13

funkcje matematyczne, 14

instrukcje iteracyjne, 15

instrukcja warunkowa, 15

kompilatory, 13

struktura programu źródłowego, 12

środowiska, 13

typy liczbowe, 14

ciąg Fibonacciego, 33, 60, 65

## D

deklaracje funkcji, 16

działania matematyczne, 21

## E

edytory programistyczne, 9

## F

formularz, 17

funkcja split(), 94

funkcje, 11

matematyczne, 10, 14, 19

rekurencyjne, 59, 60, 63

## I

instrukcja warunkowa, 10, 15, 26  
trójargumentowa, 26

instrukcje

bitowe, 25

blokowe, 25

iteracyjne, 11, 15

## J

JavaScript, 17

funkcje matematyczne, 19

język

C++, 12

FORTRAN, 124

JavaScript, 17

Pascal, 7

PLOCZR, 117

Python, 20

## K

karta perforowana, 122

krotki, 22

## L

liczba

B-super-pierwsza, 31, 44

doskonała, 31, 46

e, 35, 81

Mersenne'a, 32, 48

naturalna, 42, 43, 44

parzysta, 31, 42

pierwsza, 31, 42

podzielna, 31, 47

liczba  
 super-pierwsza, 31, 43  
 wyrazów w zdaniu, 39  
 $\pi$ , 35, 78

liczby  
 bliźniacze, 32, 50  
 zaprzyjaźnione, 32, 50

listy, 22  
 składane, 27

**Ł**

łańcuchy znaków, 23

**N**

największy wspólny dzielnik, 32, 51

**O**

obliczanie sumy, 67  
 operacje na listach, 23

**P**

palindromy, 39, 108  
 Pascal, 7  
 edytory programistyczne, 9  
 funkcje, 11  
 funkcje matematyczne, 10  
 instrukcja warunkowa, 10  
 instrukcje iteracyjne, 11  
 kompilatory, 9  
 struktura programu źródłowego, 8  
 typy liczbowe, 10  
 perforowanie kart, 122  
 permutacje, 32, 52  
 pętla  
 for, 25  
 while, 26  
 podzbiory, 33, 56  
 podział liczby naturalnej, 71  
 podziały, 34  
 pole figury, 37  
 porównywanie wydajności, 29  
 programowanie  
 funkcyjne, 28  
 komputerów, 5  
 punkty kratowe, 36, 88  
 Python, 20  
 blok funkcji, 27  
 działania matematyczne, 21  
 instrukcja warunkowa, 26  
 instrukcje bitowe, 25

instrukcje blokowe, 25  
 interpretery i edytory, 20  
 listy i krotki, 22  
 listy składane, 27  
 łańcuchy znaków, 23  
 typy liczbowe, 21  
 wyrażenia logiczne, 24

**R**

rozkład liczby naturalnej, 74  
 rozkłady, 34  
 rozwiązywanie trójkąta, 36, 83  
 równanie  
 kwadratowe, 38, 101  
 liniowe, 38, 100  
 rzut ukośny, 38, 97

**S**

schemat Hornera, 96  
 słowo kluczowe void, 72  
 struktura programu źródłowego  
 w języku C++, 12  
 w języku Pascal, 8  
 sumy, 33, 67  
 szyfr  
 Cezara, 39, 110  
 XOR, 39, 113

**T**

tablica, 123  
 taśma perforowana, 122  
 teoria informacji, 123  
 T-komputer, 117  
 trójkąt, 36  
 Pascala, 34, 75  
 typ boolean, 41  
 typy liczbowe, 10, 14, 21

**U**

układ równań, 39, 103  
 urządzenie do perforowania kart, 122

**W**

wyrażenia logiczne, 24  
 wyznacznik główny, 103  
 wzór Bineta, 65

# PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW  
w działający bankomat!

**Dowiedz się więcej i dołącz już dzisiaj!**

<http://program-partnerski.helion.pl>

GRUPA WYDAWNICZA

 **Helion SA**

**PYTHON****C++****JAVASCRIPT**

## Zadania z programowania? Zbiór programów do napisania!

C++, JavaScript, Python i Pascal — oto zbiór języków, które przećwiczysz dzięki zadaniom w tej książce. Każdy z nich ma nieco inne zastosowania, wszystkie zaś są lub będą w najbliższym czasie wykorzystywane na lekcjach informatyki i na egzaminach, a ich użycie w obliczu realnych problemów może wydatnie uprościć Ci życie. Jak najłatwiej i najszybciej nauczyć się programować w tych językach? Oczywiście dzięki ćwiczeniom! Z tą książką napiszesz konkretne programy rozwiązujące zadania numeryczne, logiczne i matematyczne — w każdym z powyższych języków.

Z praktycznym zestawem zawartych tu zadań spróbujesz znaleźć permutacje zbioru, zaszyfrować tekst szyfrem Cezara, zbudować trójkąt Pascala i wskazać punkty kratowe wewnątrz koła o promieniu  $r$ . Napiszesz według własnego pomysłu programy odpowiadające na postawione w zadaniach pytania. Jeśli jesteś uczniem, będziesz mógł w ten sposób ćwiczyć nie tylko umiejętności programistyczne, lecz także matematyczne. Jeśli jesteś nauczycielem, znajdziesz tu wiele inspiracji do pracy z uczniami. Gotowy program będziesz mógł później porównać z rozwiązaniami proponowanymi przez autorów. Dogadaj się z komputerem — w każdym z czterech języków!

**Helion**

księgarnia internetowa

<http://helion.pl>

zamówienia telefoniczne

**0 801 339900****0 601 339900**

Helion SA  
ul. Kościuszki 1c, 44-100 Gliwice  
tel.: 32 230 98 63  
e-mail: [helion@helion.pl](mailto:helion@helion.pl)  
<http://helion.pl>

Sprawdź najnowsze promocje:  
➤ <http://helion.pl/promocje>  
Książki najchętniej czytane:  
➤ <http://helion.pl/bestsellery>  
Zamów informacje o nowościach:  
➤ <http://helion.pl/nowosci>

sięgnij po **WIĘCEJ****KOD KORZYŚCI**

ISBN 978-83-283-3527-1



9 788328 335271

Informatyka w najlepszym wydaniu

cena: 29,90 zł