



Politechnika Białostocka  
Wydział Elektryczny  
Katedra Elektrotechniki, Energoelektroniki i Elektroenergetyki

Instrukcja  
do pracowni specjalistycznej z przedmiotu

## **Programowanie C**

Kod przedmiotu: **CP1S01005**

(studia stacjonarne)

# **JĘZYK C - OPERATOR WARUNKOWY, INSTRUKCJA SWITCH**

Numer ćwiczenia

**PRC\_04**

Autor:  
dr inż. Jarosław Forenc

Białystok 2023

# Spis treści

<b>1. Opis stanowiska .....</b>	<b>3</b>
1.1. Stosowana aparatura .....	3
1.2. Oprogramowanie.....	3
<b>2. Wiadomości teoretyczne.....</b>	<b>3</b>
2.1. Operator warunkowy .....	3
2.2. Instrukcja wyboru wielowariantowego - switch.....	5
2.3. Stałe zadeklarowane jako const .....	10
<b>3. Przebieg ćwiczenia.....</b>	<b>10</b>
<b>4. Literatura.....</b>	<b>12</b>
<b>5. Pytania kontrolne .....</b>	<b>13</b>
<b>6. Wymagania BHP .....</b>	<b>13</b>

---

**Materiały dydaktyczne przeznaczone dla studentów Wydziału Elektrycznego PB.**

© Wydział Elektryczny, Politechnika Białostocka, 2023 (wersja 1.0)

Wszelkie prawa zastrzeżone. Żadna część tej publikacji nie może być kopiowana i odtwarzana w jakiegokolwiek formie i przy użyciu jakichkolwiek środków bez zgody posiadacza praw autorskich.

# 1. Opis stanowiska

## 1.1. Stosowana aparatura

Podczas zajęć wykorzystywany jest komputer klasy PC z systemem operacyjnym Microsoft Windows 10/11.

## 1.2. Oprogramowanie

Na komputerach zainstalowany jest edytor kodu źródłowego Visual Studio Code 1.81 (lub nowszy) wraz z odpowiednimi rozszerzeniami (C/C++, Code Runner, Polish Language Pack for Visual Studio Code) oraz MinGW - zestaw kompilatorów różnych języków programowania (m.in. C, C++, Fortran, Java).

# 2. Wiadomości teoretyczne

## 2.1. Operator warunkowy

Operator warunkowy składa się z dwóch symboli (? - znak zapytania, : - dwukropek) i wymaga trzech operandów (wyrażeń **w1**, **w2**, **w3**). Stosując operator warunkowy otrzymujemy następującą postać wyrażenia:

$$\mathbf{w1 \ ? \ w2 \ : \ w3}$$

Wyrażenie warunkowe obliczane jest w następujący sposób: najpierw obliczane jest wyrażenie **w1**. Jeśli jego wartość jest różna od zera, to obliczane jest wyrażenie **w2** i jego wartość staje się wartością całego wyrażenia warunkowego. W przeciwnym razie **w2** jest ignorowane, a wartością wyrażenia warunkowego staje się wartość wyrażenia **w3** (po wcześniejszym jego obliczeniu).

Wyrażenia warunkowe stosowane są najczęściej wtedy, gdy pewnej zmiennej nadawana jest jedna z dwóch możliwych wartości. Mogą one zastępować proste instrukcje **if ... else**. Przykładowo, obliczanie wartości bezwzględnej zmiennej **x** może mieć następującą postać:

```
if (x < 0)
    y = -x;
else
    y = x;
```

można zastąpić  
przez:

```
y = (x < 0) ? -x : x;
```

Podobnie postępujemy z wyznaczeniem większej z dwóch zmiennych **a** i **b**:

```
if (a > b)
    max = a;
else
    max = b;
```

można zastąpić  
przez:

```
max = (a > b) ? a : b;
```

Zastosowanie wyrażania warunkowego upraszcza kod programu i może dawać w wyniku kompilacji bardziej zoptymalizowany kod wykonywalny. W poniższym przykładzie operator warunkowy został zastosowany bezpośrednio w instrukcji **printf()** do sprawdzenia czy liczba jest parzysta/nieparzysta i dodatnia/ujemna.

Sprawdzenie czy liczba jest parzysta/nieparzysta, dodatnia/ujemna.

```
#include <stdio.h>

int main(void)
{
    int x;

    printf("Podaj liczbę: ");
    scanf("%d",&x);

    if (x == 0)
        printf("Liczba: zero\n");
    else
    {
        printf("Liczba: %s\n", x>0 ? "dodatnia" : "ujemna");
        printf("Liczba: %s\n", x%2==0 ? "parzysta" :
            "nieparzysta");
    }
    return 0;
}
```

Przykładowe wyniki uruchomienia programu:

```
Podaj liczbe: 5
Liczba: dodatnia
Liczba: nieparzysta
```

```
Podaj liczbe: -6
Liczba: ujemna
Liczba: parzysta
```

```
Podaj liczbe: 0
Liczba: zero
```

## 2.2. Instrukcja wyboru wielowariantowego - switch

Instrukcja **switch** służy do podejmowania decyzji wielowariantowych. W instrukcji tej sprawdza się, czy wartość pewnego wyrażenia pasuje do jednej z kilku **całkowitych, stałych wartości** (**wyrażenie\_stale**). W przypadku stwierdzenia równości następuje przekazanie sterowania (skok) do odpowiedniego miejsca. W niektórych sytuacjach instrukcja **switch** może zastąpić wielokrotne instrukcje **if - else if**.

Ogólna postać instrukcji **switch** jest następująca:

```
switch (wyrażenie)
{
    case wyrażenie_stale: instrukcje;
    case wyrażenie_stale: instrukcje;
    case wyrażenie_stale: instrukcje;
    ...
    default: instrukcje;
}
```

Obliczane jest **wyrażenie** w nawiasach. Następnie jego wartość porównywana jest z wartościami **wyrażeń stałych** (zawartych w częściach oznaczanych przez etykiety **case**). Sterowanie jest przekazywane do tej instrukcji, którą poprzedza etykieta **case** z **wyrażeniem stałym** równym co do wartości **wyrażeniu** w nawiasach **switch**. Od tego miejsca wykonywane są wszystkie instrukcje znajdujące się po tej etykiecie oraz oznaczone przez inne etykiety, aż do końca instrukcji **switch**. Jeśli nie znajdzie się żadna etykieta „pasująca” do

wartości **wyrażenia** w nawiasach **switch**, to sterowanie jest przekazywane do części oznaczonej przez etykietę **default**. Jeśli nie ma etykiety **default**, to sterowanie przekazywane jest do instrukcji następującej po **switch**. Etykieta **default** może wystąpić tylko jeden raz. W składni instrukcji **switch** wszystkie instrukcje oraz etykieta **default** są opcjonalne.

**Wyrażania stałe** występujące po etykietach **case** nie mogą powtarzać się. Jeśli wystąpi taka sytuacja, to kompilator zasygnalizuje błąd. W jednej instrukcji **switch** może występować maksymalnie do 1023 etykiet **case**. **Wyrażenie stałe** musi mieć typ całkowity. Jego wartość powinna być znana w trakcie kompilacji i nie może zostać zmieniona w fazie wykonania programu. Jako **wyrażenie stałe** najczęściej stosuje się:

- liczby całkowite, np. **1, 2, 3, 0, -1, -2**;
- stałe zadeklarowane jako **const** lub przez dyrektywę preprocesora **#define**;
- znaki umieszczone w apostrofach, np. **'+', 'a'**.

W poniższym programie funkcja **getchar()** odczytuje wciśnięty klawisz i podstawia jego kod pod zmienną **key**. Następnie w instrukcji **switch** kod klawisza porównywany jest z wyrażeniami stałymi znajdującymi się po **case**. Jeśli wciśniętym klawiszem był **'+'**, to zmienne **x1** i **x2** dodawane są do siebie i wyświetlana jest ich suma. Jeśli wciśnięto **'-'**, to zmienne są odejmowane. Wprowadzenie innego znaku spowoduje wyświetlenie tekstu: **Nieznana operacja!**

Wybór arytmetycznego działania w zależności od wciśniętego klawisza.

```
#include <stdio.h>

int main(void)
{
    int    key;
    float  x1 = 10.0, x2 = 5.0, y;

    printf("Podaj dzialanie (+,-): ");
    key = getchar();
```

```

switch (key)
{
case '+':
    y = x1 + x2;
    printf("Dodawanie: y = %.2f \n",y);
    break;
case '-':
    y = x1 - x2;
    printf("Odejmowanie: y = %.2f \n",y);
    break;
default:
    printf("Nieznana operacja!\n");
}

return 0;
}

```

Przykładowe wyniki uruchomienia programu:

```

Podaj dzialanie: +
Dodawanie: y = 15.00

```

```

Podaj dzialanie: -
Odejmowanie: y = 5.00

```

```

Podaj dzialanie: *
Nieznana operacja!

```

Po instrukcjach każdego wariantu **case** występują instrukcje **break**. Powodują one natychmiastowe opuszczenie instrukcji **switch**. Ich brak spowodowałby wykonanie wszystkich instrukcji (do końca instrukcji **switch**) występujących po każdym **case**.

```

switch (key)
{
case '+':
    y = x1 + x2;
    printf("Dodawanie: y = %.2f \n",y);
case '-':
    y = x1 - x2;
    printf("Odejmowanie: y = %.2f \n",y);
}

```

```
default:
    printf("Nieznana operacja!\n");
}
```

Po wciśnięciu '+' wyświetlone zostałyby komunikaty:

```
Dodawanie: y = 15.00
Odejmowanie: y = 5.00
Nieznana operacja!
```

Po wciśnięciu '-' wyświetlone zostałyby komunikaty:

```
Odejmowanie: y = 5.00
Nieznana operacja!
```

Po wciśnięciu innego znaku wyświetlony zostałby komunikat:

```
Nieznana operacja!
```

Kolejny program wyświetla słownie ocenę wczytaną z klawiatury.

Program wyświetlający słownie ocenę wprowadzoną jako liczba.

```
#include <stdio.h>

int main(void)
{
    int ocena;

    printf("Podaj ocene: ");
    scanf("%d",&ocena);
    switch (ocena)
    {
        case 5:
            printf("Twoja ocena: bardzo dobry\n");
            break;
        case 4:
            printf("Twoja ocena: dobry\n");
            break;
        case 3:
            printf("Twoja ocena: dostateczny\n");
            break;
    }
}
```



```

        case 2:
            printf("Twoja ocena: niedostateczny\n");
            break;
        default:
            printf("Bledna ocena\n");
    }

    return 0;
}

```

Przykłady uruchomienia programu:

```

Podaj ocene: 4
Twoja ocena: dobry

```

```

Podaj ocene: 2
Twoja ocena: niedostateczny

```

```

Podaj ocene: 0
Bledna ocena

```

Z każdym wariantem może być związane jedno lub kilka wyrażeń stałych.

Program wyświetlający informację o wprowadzonej ocenie.

```

#include <stdio.h>

int main(void)
{
    int ocena;

    printf("Podaj ocene: ");
    scanf("%d",&ocena);
    switch (ocena)
    {
        case 5: case 4: case 3:
            printf("Ocena pozytywna\n");
            break;
        case 2:
            printf("Ocena negatywna\n");
            break;
    }
}

```

```
        default:
            printf("Bledna ocena\n");
    }

    return 0;
}
```

Przykłady uruchomienia programu:

```
Podaj ocene: 4
Ocena pozytywna
```

```
Podaj ocene: 3
Ocena pozytywna
```

```
Podaj ocene: 2
Ocena negatywna
```

```
Podaj ocene: 0
Bledna ocena
```

### 2.3. Stałe zadeklarowane jako const

Stałą specyfikuje się wstawiając przed nazwą typu słowo **const**. Tak zadeklarowana "zmienna" musi otrzymać wartość w trakcie inicjalizacji i nie może być później zmieniana.

```
const int x = 10;
```

## 3. Przebieg ćwiczenia

Na pracowni specjalistycznej należy wykonać wybrane zadania wskazane przez prowadzącego zajęcia. W różnych grupach mogą być wykonywane różne zadania.

1. Napisz program obliczający i wyświetlający liczbę bajtów potrzebną do zapisania wprowadzonej z klawiatury liczby bitów. Przyjmij, że 1 bajt to 8 bitów. Zastosuj operator warunkowy.

2. Napisz program, w którym użytkownik wprowadza z klawiatury liczbę całkowitą. Jeśli liczba jest ujemna, to przypisz zmiennej **znak** wartość **-1**. Jeśli liczba jest równa zero, to przypisz zmiennej **znak** wartość **0**. Jeśli liczba jest dodatnia, to przypisz zmiennej **znak** wartość **1**. Zastosuj operator warunkowy. Wyświetl wartość zmiennej **znak**.
3. Wskaźniki zadziałania wkładek bezpiecznikowych oznacza się odpowiednimi kolorami zależnie od ich prądu znamionowego (Tabela 1).

Tabela 1. Wybrane kolory wskaźników zadziałania wkładek bezpiecznikowych

Barwa wskaźnika	Prąd znamionowy wkładki
zielona	6
czerwona	10
szara	16
niebieska	20

Napisz program, w którym po wprowadzeniu przez użytkownika prądu znamionowego wkładki, wyświetlana jest barwa odpowiadającego jej wskaźnika zadziałania. W przypadku błędnej wartości prądu wyświetl odpowiedni komunikat. Zastosuj instrukcję **switch**.

4. Napisz program, w którym użytkownik wprowadza numer dnia tygodnia. Program powinien wypisać tekst:
  - „zwykły dzień” - dla dni od poniedziałku do piątku;
  - „weekend” - dla soboty i niedzieli;
  - „błędny numer dnia” - dla pozostałych wartości.

Przyjmij: 1 - poniedziałek, 2 - wtorek, 3 - środa, itd. Zastosuj instrukcję **switch**.

5. Napisz program, w którym użytkownik wprowadza rok oraz numer miesiąca, a program wyświetla liczbę dni w podanym miesiącu. Uwzględnij obecność lat przestępnych. Zastosuj instrukcję **switch**. Uwaga: rok przestępny (liczbowo) spełnia następujące warunki: jest podzielny przez 4, ale nie jest podzielny przez 100 lub jest podzielny przez 400.

6. Napisz program, w którym użytkownik wprowadza numer miesiąca, a program wyświetla nazwy wszystkich miesięcy, które pozostały od tego miesiąca do końca roku. Wyświetl odpowiedni komunikat w przypadku błędnego numeru miesiąca. Zastosuj instrukcję **switch**.

Przykładowe wywołanie programu:

```
Podaj numer miesiaca:  9
-----
wrzesien
pazdziernik
listopad
grudzien
```

7. Napisz program, w którym użytkownik wprowadza dodatnią, całkowitą liczbę dwucyfrową. Program powinien wypisać tę liczbę słownie. Zastosuj instrukcję **switch**.

Przykładowe wywołanie programu:

```
Liczba:  37
Słownie: trzydziesci siedem
```

## 4. Literatura

- [1] Prata S.: Język C. Szkoła programowania. Wydanie VI. Helion, Gliwice, 2016.
- [2] Kernighan B.W., Ritchie D.M.: Język ANSI C. Programowanie. Wydanie II. Helion, Gliwice, 2010.
- [3] Deitel P.J., Deitel H.: Język C. Solidna wiedza w praktyce. Wydanie VIII. Helion, Gliwice, 2020.
- [4] Kochan S.G.: Język C. Kompendium wiedzy. Wydanie IV. Helion, Gliwice, 2015.
- [5] King K.N.: Język C. Nowoczesne programowanie. Wydanie II. Helion, Gliwice, 2011.
- [6] Wileczek R.: Microsoft Visual C++ 2008. Tworzenie aplikacji dla Windows. Helion, Gliwice, 2009.
- [7] <http://www.cplusplus.com/reference/clibrary> - C library - C++ Reference

[8] <https://cpp0x.pl/dokumentacja/standard-C/1> - Standard C

[9] <https://code.visualstudio.com/> - Visual Studio Code

[10] <https://sourceforge.net/projects/mingw/> - MinGW

## 5. Pytania kontrolne

1. Omów zasadę działania operatora warunkowego `? :`. W jaki sposób operator warunkowy może zastępować instrukcję **if-else**?
2. Omów składnię i zasadę działania instrukcji wyboru wielowariantowego **switch**.
3. Wyjaśnij, w jakim celu w instrukcji **switch** stosowane są instrukcje **break**?

## 6. Wymagania BHP

Warunkiem przystąpienia do praktycznej realizacji ćwiczenia jest zapoznanie się z instrukcją BHP i instrukcją przeciw pożarową oraz przestrzeganie zasad w nich zawartych.

W trakcie zajęć laboratoryjnych należy przestrzegać następujących zasad.

- Sprawdzić, czy urządzenia dostępne na stanowisku laboratoryjnym są w stanie kompletnym, nie wskazującym na fizyczne uszkodzenie.
- Jeżeli istnieje taka możliwość, należy dostosować warunki stanowiska do własnych potrzeb, ze względu na ergonomię. Monitor komputera ustawić w sposób zapewniający stałą i wygodną obserwację dla wszystkich członków zespołu.
- Sprawdzić prawidłowość połączeń urządzeń.
- Załączenie komputera może nastąpić po wyrażeniu zgody przez prowadzącego.
- W trakcie pracy z komputerem zabronione jest spożywanie posiłków i picie napojów.

- W przypadku zakończenia pracy należy zakończyć sesję przez wydanie polecenia wylogowania. Zamknięcie systemu operacyjnego może się odbywać tylko na wyraźne polecenie prowadzącego.
- Zabronione jest dokonywanie jakichkolwiek przełączeń oraz wymiana elementów składowych stanowiska.
- Zabroniona jest zmiana konfiguracji komputera, w tym systemu operacyjnego i programów użytkowych, która nie wynika z programu zajęć i nie jest wykonywana w porozumieniu z prowadzącym zajęcia.
- W przypadku zaniku napięcia zasilającego należy niezwłocznie wyłączyć wszystkie urządzenia.
- Stwierdzone wszelkie braki w wyposażeniu stanowiska oraz nieprawidłowości w funkcjonowaniu sprzętu należy przekazywać prowadzącemu zajęcia.
- Zabrania się samodzielnego włączania, manipulowania i korzystania z urządzeń nie należących do danego ćwiczenia.
- W przypadku wystąpienia porażenia prądem elektrycznym należy niezwłocznie wyłączyć zasilanie stanowiska. Przed odłączeniem napięcia nie dotykać porażonego.