

Programowanie w C i C++

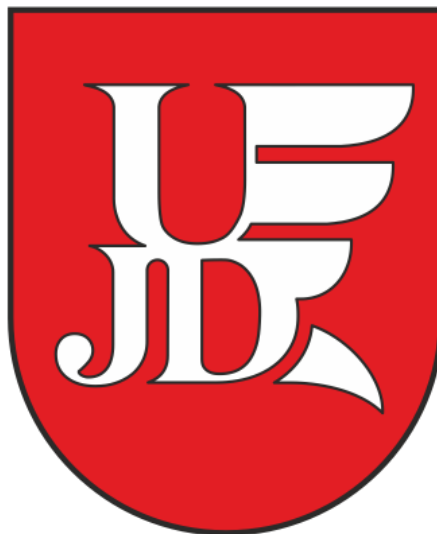
Oskar Purgal

Starosta Grupy Informatyków

17.03.2025

Informatyka – Inżynieria Oprogramowania (IO)

Laboratorium 2



Zadanie 1

Napisz program wyświetlający dwa napisy "Witaj świecie" oraz "Programowanie w C/C++" każdy w osobnej linii. Użyj do tego jednej instrukcji do wyświetlania ciągów. Kolejno wyświetl trzy dowolne linie tekstu a po każdej z nich informację "Naciśnij Enter aby kontynuować...". Po tym napisie program ma oczekiwać na naciśnięcie przez użytkownika dowolnego klawisza. Zadbaj o czyszczenie bufora klawiatury. Można użyć funkcji `getchar()` [stdio] a następnie `getche()` [conio] i zaobserwować różnicę podczas naciśnięcia klawisza np litery i samego klawisza enter.

Programiz
C++ Online Compiler

main.cpp

Share

Run

```
1 #include <stdio.h>
2
3 int main() {
4     printf("Witaj swiecie\n");
5     printf("Nacisnij Enter aby kontynuowac...\n");
6     getchar();
7
8     printf("Programowanie w C/C++\n");
9     printf("Nacisnij Enter aby kontynuowac...\n");
10    getchar();
11
12    printf("Tutaj Oskar Purgal Pozdrawiam\n");
13    printf("Nacisnij Enter aby kontynuowac...\n");
14    getchar();
15
16    return 0;
17 }
18
```

Output

Witaj swiecie
Nacisnij Enter aby kontynuowac...

Programowanie w C/C++
Nacisnij Enter aby kontynuowac...

Tutaj Oskar Purgal Pozdrawiam
Nacisnij Enter aby kontynuowac...

=== Code Execution Successful ===

Zadanie 2

Zadanie 2

Stworzyć nowy projekt, nazwać go Projekt1 i i zapisać. Wkleić poniższy kod

```
#include <iostream>
#include <stdio.h>

using namespace std;int i,j;int main(){for (j=1; j<=10; j++){for (i=1;j<=10;i++){k=i*j;printf("%4d",k);cout<<endl;}return 0;}
```

Przeprowadzić formatowanie ręczne zgodnie z dobrymi praktykami programowania tak aby każdy wiersz zaczynał się w kolumnie zero. Następnie wykonać autoformatowanie za pomocą wtyczki AStyle. Skompilować i sprawdzić co podaje kompilator w oknie statusu Logs & others. Sprawdzić wynik działania programu.

```
#include <iostream>
```

```
#include <stdio.h>
```

```
using namespace std;
```

```
int i, j;
```

```
int k;
```

```
int main() {
```

```
for (j = 1; j <= 10; j++) {
```

```
for (i = 1; i <= 10; i++) {
```

```
k = i * j;
```

```
printf("%4d", k);
```

```
}
```

```
cout << endl;
```

```
}  
  
return 0;  
  
}
```

main.cpp

Share

Run

```
1  #include <iostream>  
2  #include <stdio.h>  
3  
4  using namespace std;  
5  
6  int i, j;  
7  int k;  
8  
9  int main() {  
10     for (j = 1; j <= 10; j++) {  
11         for (i = 1; i <= 10; i++) {  
12             k = i * j;  
13             printf("%4d", k);  
14         }  
15         cout << endl;  
16     }  
17     return 0;  
18 }  
19
```

Output

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

=== Code Execution Successful ===

Zadanie 3

Zadanie 3

Stworzyć nowy projekt i wkleić poniższy kod:

```
#include <iostream>
#include <stdio.h>

using namespace std;

int i,j;
float K;

int main()
{
    for (j=1; j<=10; j++)
    {
        for (i=1; i<=10; i++)
        {
            k=i*j
            printf("%4d",k);
        }
        cout << endl;
    }
    //return 0;
}
```

Przeprowadzić proces kompilacji. Jeśli pojawią się błędy opisać je i poprawić.

Pseudokod Po naprawie:

```
#include <iostream>
#include <stdio.h>

using namespace std;

int i, j, k;

int main()
{
    for (j = 1; j <= 10; j++)
    {
        for (i = 1; i <= 10; i++)
        {
            k = i * j;
            printf("%4d", k);
        }
    }
}
```

```

cout << endl;

}

return 0;

}

```

```

main.cpp
1  #include <iostream>
2  #include <stdio.h>
3
4  using namespace std;
5
6  int i, j, k;
7
8  int main()
9  {
10     for (j = 1; j <= 10; j++)
11     {
12         for (i = 1; i <= 10; i++)
13         {
14             k = i * j;
15             printf("%4d", k);
16         }
17         cout << endl;
18     }
19     return 0;
20 }
21

```

Output

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

=== Code Execution Successful ===

Zadanie 4

Napisz program dzielący dwie dowolne liczby całkowite przez siebie, wynik ma być liczbą rzeczywistą. Skorzystaj ze stałych lub literałów liczbowych. Przeanalizuj wyniki i sposób działania operacji dzielenia w C++.

Kod:

```

#include <iostream>

using namespace std;

int main() {

int a, b;

double wynik;

cout << "Podaj pierwsza liczbe calkowita (a): ";

cin >> a;

cout << "Podaj druga liczbe calkowita (b): ";

cin >> b;

wynik = static_cast<double>(a) / b;

cout << "Wynik dzielenia " << a << " przez " << b << " to: " << wynik << endl;

return 0;

}

```

```
main.cpp
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int a, b;
6     double wynik;
7
8     cout << "Podaj pierwsza liczbe calkowita (a): ";
9     cin >> a;
10    cout << "Podaj druga liczbe calkowita (b): ";
11    cin >> b;
12
13    wynik = static_cast<double>(a) / b;
14
15    cout << "Wynik dzielenia " << a << " przez " << b << " to: " << wynik << endl;
16
17    return 0;
18 }
19
```

Output

```
Podaj pierwsza liczbe calkowita (a): 5
Podaj druga liczbe calkowita (b): 6
Wynik dzielenia 5 przez 6 to: 0.833333

=== Code Execution Successful ===
```

Zadanie 5:

Zadanie 5

Napisać program wyświetlający ile bajtów zajmują dostępne w Twoim kompilatorze i wersji języka typy danych. Sprawdź typy: bool, char, unsigned char, int, unsigned int, long int, long long int, unsigned long long int, float, double, long double. Wynik powinien wyglądać mniej więcej tak:

Dlaczego w dwóch ostatnich wierszach ilość znaków napisu nie odpowiada podanemu wynikowi?

Zadanie 6:

Napisać program sprawdzający czy podana wartość znajduje się w przedziale:

<-10..20)

(0..10>

<-3..3>

Podana wartość może występować w programie jako stała lub zmienna z konkretną wartością. Sprawdzić wynik dla wartości wewnętrznych i granicznych: -10,20,0,10,-3,3. Odpowiedzią może być 0 lub 1.

```
#include <stdio.h>
```

```
int main() {
```

```
int value;
```

```
value = -10;
```

```
printf("%d\n", (value >= -10 && value < 20));
```

```
printf("%d\n", (value > 0 && value <= 10));
```

```
printf("%d\n", (value >= -3 && value <= 3));
```

```
return 0;
```

```
}
```

main.cpp	Output
<pre>1 #include <stdio.h> 2 3 int main() { 4 int value; 5 value = -10; 6 7 printf("%d\n", (value >= -10 && value < 20)); 8 printf("%d\n", (value > 0 && value <= 10)); 9 printf("%d\n", (value >= -3 && value <= 3)); 10 11 return 0; 12 } 13 </pre>	<pre>1 0 0 === Code Execution Successful ===</pre>

Zadanie 7:

Sprawdzić czy podana wartość występuje w którymś z przedziałów:

<-10..22) .. (30..40)

(0..3> .. <5..7>

Sprawdzić wynik dla wartości wewnętrznych i granicznych podanych przedziałów. Odpowiedzią może być 0 lub 1.

Nie używamy instrukcji warunkowej

```
#include <stdio.h>
```

```
int main() {
```

```
int value = 5;
```

```
printf("%d\n", ((value >= -10 && value < 22) || (value >= 30 && value < 40)));
```

```
printf("%d\n", ((value > 0 && value <= 3) || (value >= 5 && value <= 7)));
```

```
return 0;}
```

main.cpp	Output
<pre>1 #include <stdio.h> 2 3 int main() { 4 int value = 5; 5 6 printf("%d\n", ((value >= -10 && value < 22) (value >= 30 && value < 40))); 7 printf("%d\n", ((value > 0 && value <= 3) (value >= 5 && value <= 7))); 8 9 return 0; 10 } 11 </pre>	<pre>1 1 === Code Execution Successful ===</pre>

Zadanie 8:

Napisz program obliczający sumę i średnią trzech liczb (literały, stałe). Uzasadnij dobór typów danych argumentów i wyniku. Rozbuduj program o operacje odejmowania na stałych i zmiennych typów int oraz unsigned int. Przeprowadź test dla odejmowania liczby większej od mniejszej zwłaszcza dla typów unsigned oraz mieszanych signed i unsigned. Skorzystaj ze stałych definiowanych za pomocą notacji C (stałe symboliczne, dyrektywy preprocesora) oraz C++ a także modyfikatorów typów tych stałych.

main.cpp	Output
<pre>1 #include <stdio.h> 2 3 #define NUM1 10 4 #define NUM2 20 5 #define NUM3 30 6 7 const int const_num = 100; 8 9 int main() { 10 int sum = NUM1 + NUM2 + NUM3; 11 float avg = sum / 3.0; 12 13 printf("Suma: %d\n", sum); 14 printf("Średnia: %.2f\n", avg); 15 16 int a = 10, b = 20; 17 printf("Odejmowanie (int): %d\n", a - b); 18 19 unsigned int c = 10, d = 20; 20 printf("Odejmowanie (unsigned int): %u\n", c - d); 21 22 unsigned int e = 10; 23 int f = 20; 24 printf("Odejmowanie mieszane: %u\n", e - f); 25 26 printf("Odejmowanie (stałe): %d\n", const_num - NUM1); 27 28 return 0; 29 } 30</pre>	<pre>Suma: 60 Średnia: 20.00 Odejmowanie (int): -10 Odejmowanie (unsigned int): 4294967286 Odejmowanie mieszane: 4294967286 Odejmowanie (stałe): 90 === Code Execution Successful ===</pre>

Zadanie 9

Napisz program obliczający tygodniowe całkowite wynagrodzenie. Uwzględnij godziny i stawki standardowe oraz nadliczbowe. Zadeklaruj zmienne, stałe oraz dobierz ich typy danych. Program ma podać ile pracownik zarobił w godzinach podstawowych, ile w nadliczbowych, ile w sumie oraz jaki procent godzin standardowych stanowią godziny nadliczbowe. Zakładamy, że stawka nadliczbowa jest o 50% większa od standardowej.

```
#include <stdio.h>
```

```
int main() {
```

```
    float regular_hours, overtime_hours;
```

```
    scanf("%f %f", &regular_hours, &overtime_hours);
```

```
    printf("%.2f %.2f %.2f %.2f%%\n",
```

```
        regular_hours * 15,
```

```
        overtime_hours * 22.5,
```

```
        regular_hours * 15 + overtime_hours * 22.5,
```

```
        (overtime_hours / (regular_hours + overtime_hours)) * 100);
```

```
    return 0;
```

```
}
```


Zadanie 10

Napisać program wyświetlający znak 'A' za pomocą zmiennej typu int oraz wartość 97 za pomocą zmiennej typu char.

Programiz

C++ Online Compiler

main.cpp

Share

Run

```
1 #include <stdio.h>
2
3 int main() {
4     int a = 'A';
5     char b = 97;
6
7     printf("%c\n", a);
8     printf("%d\n", b);
9
10    return 0;
11 }
12
```

A

97

=== Code Execution Successful ===




Zadanie 11

Zdefiniuj typy zmiennych dla następujących wartości:

- 3999999999
- 'a'
- 23
- 0.3
- 999e999

Napisz program wyświetlający wartości takich zmiennych.

main.cpp



Run

Output

Clear

```
1 #include <stdio.h>
2
3 int main() {
4     long long long_long_num = 3999999999;
5     char char_num = 'a';
6     int int_num = 23;
7     float float_num = 0.3;
8     double double_num = 1.7e308;
9
10    printf("%lld\n", long_long_num);
11    printf("%c\n", char_num);
12    printf("%d\n", int_num);
13    printf("%.1f\n", float_num);
14    printf("%f\n", double_num);
15
16    return 0;
17 }
18
```

```
3999999999
a
23
0.3
16999999999999999388307957886599817433346074304075874502773119193537729178160565864330091787584707
98857226246798318891916991610559335717426836996206247363529647463651566046493566304068495784430
35243678150285532727122989863863108286445132123539211232533116754998568756505124374154292179946
23324794855339589632.000000

=== Code Execution Successful ===
```

Zadanie 12

Napisać program przedstawiający efekty wyjścia poza zakres zmiennych typów w stronę -inf jak i +inf (użyj też modyfikatora unsigned):

- int
- bool (0-falsz, wszystko inne prawda)
- long
- char
- float

Wykorzystaj stałe takie jak INT_MAX, INT_MIN dostępne w Twoim kompilatorze (np biblioteka limits.h, float.h) lub sprawdź samodzielnie dostępne zakresy.

Do zmiennej typu int wpisz wartość maksymalną, dodaj jeden, wyświetl wartość tej zmiennej. Efekt? Wnioski?

main.cpp	Output
<pre>1 #include <stdio.h> 2 #include <limits.h> 3 #include <float.h> 4 5 int main() { 6 printf("%lld\n", (long long)INT_MAX); 7 printf("%llu\n", (unsigned long long)UINT_MAX); 8 printf("%d\n", 2); 9 printf("%lld\n", (long long)LONG_MAX); 10 printf("%d\n", (char)(CHAR_MAX)); 11 printf("%f\n", FLT_MAX); 12 13 return 0; 14 } 15</pre>	<pre>2147483647 4294967295 2 9223372036854775807 127 340282346638528859811704183484516925440.000000 === Code Execution Successful ===</pre>

Zadanie 12a

Napisz program pokazujący efekty użycia operatora unarnego pre i post inkrementacji użytego w wyrażeniu. Przykład: zmienna a ma wartość 10. Napisz kod (jedna instrukcja) wyświetlający wartość tej zmiennej oraz zmiennej potraktowanej operatorem preinkrementacji np:

cout << a << " " << ++a << endl;

Wnioski? Jak zatem wygląda kolejność przetwarzania zmiennych i wykonywania instrukcji.

main.cpp	Output
<pre>1 #include <iostream> 2 using namespace std; 3 4 int main() { 5 int a = 10; 6 cout << a << " " << ++a << endl; 7 return 0; 8 } 9</pre>	<pre>10 11 === Code Execution Successful ===</pre>

Zadanie 13

Napisać program obliczający kwotę netto z podanej przez użytkownika kwoty brutto. Wartość podatku wyrażoną procentowo podaje także użytkownik. (zmienne rzeczywiste) Wynikiem ma być informacja typu: "Przy podatku 23% z kwoty brutto 123PLN otrzymujemy kwotę netto równą 100PLN. Wartość podatku to 23PLN". Wartości mają być wyświetlane z dokładnością do drugiego miejsca po przecinku. Zadbaj o odpowiednie komunikaty.

main.cpp	Output
<pre>1 #include <iostream> 2 #include <iomanip> 3 using namespace std; 4 5 int main() { 6 double brutto, podatek, netto, wartosc_podatku; 7 8 cout << "Podaj kwotę brutto: "; 9 cin >> brutto; 10 11 cout << "Podaj wartość podatku w procentach: "; 12 cin >> podatek; 13 14 wartosc_podatku = brutto * (podatek / 100); 15 netto = brutto - wartosc_podatku; 16 17 cout << fixed << setprecision(2); 18 cout << "Przy podatku " << podatek << "% z kwoty brutto " << brutto << " PLN otrzymujemy kwotę netto równą " << netto << " PLN. Wartość podatku to " << wartosc_podatku << " PLN." << endl; 19 20 return 0; 21 } 22</pre>	<pre>Podaj kwotę brutto: 1523 Podaj wartość podatku w procentach: 23% Przy podatku 23.00% z kwoty brutto 1523.00 PLN otrzymujemy kwotę netto równą 1172.71 PLN. Wartość podatku to 350.29 PLN. === Code Execution Successful ===</pre>

Zadanie 14

Napisz program wczytujący z klawiatury liczbę rzeczywistą i wyświetlający jej wartość bezwzględną. Użyj funkcji standardowej umieszczonej w zewnętrznej bibliotece swojego środowiska IDE.

Programiz
C++ Online Compiler

main.cpp

Share

Run

```
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 int main() {
6     double liczba;
7
8     cout << "Podaj liczbę: ";
9     cin >> liczba;
10
11     cout << "Wartość bezwzględna liczby: " << abs(liczba) << endl;
12
13     return 0;
14 }
15
```

Output

Podaj liczbę: -5
Wartość bezwzględna liczby: 5

=== Code Execution Successful ===

Zadanie 15

Napisz program, który wczytuje ze standardowego wejścia liczbę całkowitą i wypisuje na standardowym wyjściu liczbę o jeden większą. Zastosować i omówić trzy sposoby inkrementacji w C++. Przedyskutować sposoby inkrementowania przedrostkowego i przyrostkowego w wyrażeniach.

Programiz
C++ Online Compiler

main.cpp

Share

Run

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int liczba;
6
7     cout << "Podaj liczbę całkowitą: ";
8     cin >> liczba;
9
10    cout << ++liczba << endl;
11    cout << liczba++ << endl;
12    cout << liczba + 1 << endl;
13
14    return 0;
15 }
16
```

Output

Podaj liczbę całkowitą: 5
6
6
8

=== Code Execution Successful ===

Zadanie 16

Napisać program przeliczający odległość podaną w km na mile. Zakładamy, że 1 mila to 1.6 km.

Programiz
C++ Online Compiler

main.cpp

Share

Run

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     double km;
6
7     cout << "Podaj odległość w kilometrach: ";
8     cin >> km;
9
10    double mile = km / 1.6;
11
12    cout << "Odległość w milach: " << mile << endl;
13
14    return 0;
15 }
```

Output

Podaj odległość w kilometrach: 512
Odległość w milach: 320

=== Code Execution Successful ===

Zadanie 17

Napisać program przyjmujący dane: długość trasy w km oraz ilość spalonego paliwa. Program jako wynik ma podać średnie zużycie paliwa na 100km. Zadbać o odpowiednie informacje dla użytkownika.

Programiz
C++ Online Compiler

main.cpp

Share

Run

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     double dlugosc_trasy, spalone_paliwo;
6
7     cout << "Podaj długość trasy w kilometrach: ";
8     cin >> dlugosc_trasy;
9
10    cout << "Podaj ilość spalonego paliwa w litrach: ";
11    cin >> spalone_paliwo;
12
13    double zuzycie_na_100km = (spalone_paliwo / dlugosc_trasy) * 100;
14
15    cout << "Średnie zużycie paliwa na 100 km wynosi: " << zuzycie_na_100km << " litrów." << endl;
16
17    return 0;
18 }
19
```

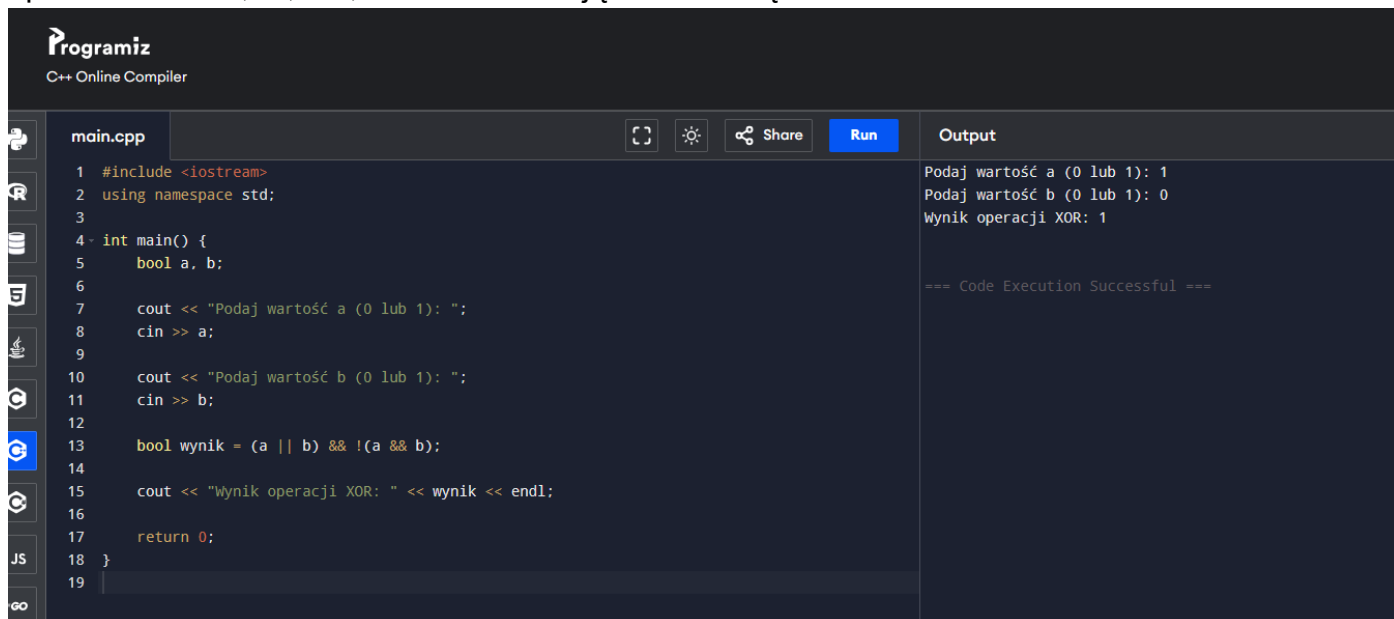
Output

Podaj długość trasy w kilometrach: 989
Podaj ilość spalonego paliwa w litrach: 120
Średnie zużycie paliwa na 100 km wynosi: 12.1335 litrów.

=== Code Execution Successful ===

Zadanie 19

Napisz program będący symulatorem operacji logicznej XOR. Na tym etapie nie dysponujesz operatorami and, or, xor, not ani instrukcją warunkową.



The screenshot shows the Programiz C++ Online Compiler interface. The code editor on the left contains a C++ program named `main.cpp` that simulates the XOR operation using only AND, OR, and NOT operators. The program prompts the user to enter two values, `a` and `b`, each being 0 or 1. It then calculates the XOR result using the expression `bool wynik = (a || b) && !(a && b);` and displays the result. The output panel on the right shows the program's execution: it prompts for `a` and `b`, receives inputs 1 and 0 respectively, and outputs the XOR result as 1. The execution is marked as successful.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     bool a, b;
6
7     cout << "Podaj wartość a (0 lub 1): ";
8     cin >> a;
9
10    cout << "Podaj wartość b (0 lub 1): ";
11    cin >> b;
12
13    bool wynik = (a || b) && !(a && b);
14
15    cout << "Wynik operacji XOR: " << wynik << endl;
16
17    return 0;
18 }
19
```

Output:

```
Podaj wartość a (0 lub 1): 1
Podaj wartość b (0 lub 1): 0
Wynik operacji XOR: 1

=== Code Execution Successful ===
```

Zadanie 21

Odpowiedz na pytania:

- Czym się różni znak '3' od cyfry 3 w sensie języka C++
- Co to jest typ danych char i do czego go używamy
- Co to jest typ int i jakie ma zastosowanie
- Podaj różnicę między typem int a float
- Jaki typ danych zastosujesz w przypadku wyznaczenia kwoty w PLN
- Jakie zastosowanie ma stała a jakie zmienna
- Jak można zadeklarować stałą
- Jak można zmienić wartość stałej
- Podaj różnice pomiędzy `==` a `=`
- Czym się różni typ od zmiennej
- Omów działanie znanych modyfikatorów zmiennych
- Które zmienne, ze znakiem czy bez znaku mogą przechowywać większą ilość danych? Podaj powody.
- Co to jest rzutowanie i jak się je przeprowadza?

Odpowiedzi:

- '3' to znak (char), 3 to liczba całkowita (int).
- char przechowuje pojedynczy znak.
- int to liczba całkowita, używana do przechowywania liczb całkowitych.
- int to liczba całkowita, float to liczba zmiennoprzecinkowa.
- float lub double do kwoty w PLN.
- Stała ma stałą wartość, zmienna zmienia wartość.

g) Stałą deklaruje się za pomocą `const`, np. `const int x = 10;`.

h) Stałej nie można zmienić.

i) `==` porównuje wartości, `=` przypisuje wartość.

j) Typ to kategoria danych, zmienna to uchwyt do danych.

k) `signed` – liczby dodatnie i ujemne, `unsigned` – tylko dodatnie, `long` – większy zakres, `short` – mniejszy zakres.

l) Zmienne `unsigned` mają większy zakres niż `signed`.

m) Rzutowanie to konwersja między typami, np. `(double)x`.