

INFORMATYKA

C

WYDZIAŁ ELEKTROTECHNIKI I
INFORMATYKI
ELEKTROTECHNIKA
DR INŻ. MICHAŁ ŁANCZONT

V

Podjmowanie decyzji

1. Warunek

- *Warunki proste i złożone*
- *Zagnieżdżanie*

2. Wybór

3. Zmienna logiczna

4. Operator warunkowy

Decyzja

- Podejmowanie decyzji przez program
- Analiza danych wejściowy decyduje o wykonaniu określonego kodu
- W języku C dostępne są dwie konstrukcje umożliwiające wykonanie określonego kodu w zależności od spełnienia określonego warunku oraz operator wyboru
 - Instrukcja warunkowa **IF**
 - Instrukcja wyboru **SWITCH**
 - Operator wyboru **?**

Zmienna logiczna

- Typ logiczny **bool** standardowo nie jest dostępny
- Podpięcie do kodu programu biblioteki **<stdbool.h>** dostarcza narzędzi umożliwiających korzystanie ze zmiennej logicznej **bool**

```
1  #include <stdio.h>
2  #include <stdbool.h>
3
4  int main()
5  {
6      bool x = true;
7      bool y = 0;
8      bool z = 12;
9      printf("Zmienna logiczna x ma wartosc %i\n",x);
10     printf("Zmienna logiczna y ma wartosc %i\n",y);
11     printf("Zmienna logiczna z ma wartosc %i\n",z);
12 }
```

C:\WINDOWS\system32\cmd.exe

```
Zmienna logiczna x ma wartosc 1
Zmienna logiczna y ma wartosc 0
Zmienna logiczna z ma wartosc 1

Press any key to continue . . .
```

Instrukcja warunkowa

- W języku C podstawową metodą podejmowania decyzji jest użycie ogólnej postaci instrukcji **if**

```
if (warunek)  
    instrukcja;
```

- Jeżeli warunek zapisany w nawiasie zwróci wartość **true** to zostanie wykonana linia kodu zapisana bezpośrednio po warunku do znaku ;
- Jeżeli warunek zwróci wartość **false** to linia kodu po warunku zostanie pominięta

Przykład

```
1  #include <stdio.h>
2
3  int main()
4  {
5      printf("Podaj liczbe calkowita: ");
6      int x;
7      scanf("%i",&x);
8      if(x%2==0)
9          printf("Podana liczba jest liczba parzysta.");
10 }
```

 C:\Windows\system32\cmd.exe

```
Podaj liczbe calkowita: 34
Podana liczba jest liczba parzysta.
Press any key to continue . . .
```

- Decyzja w instrukcji warunkowej może być podejmowano w oparciu o:
 1. Prosty warunek logiczny – operatory relacji (**==**, **!=**, **<**, **<=**, **>**, **>=**)
 2. Złożony warunek logiczny – operatory logiczne (**&&** - **AND**, **||** - **OR**, **!** - **NOT**)
 3. Wynik działania funkcji typu **bool**
 4. Kombinacji wymienionych powyżej

Przykład

C:\Windows\system32\cmd.exe

Zdefiniuj zakres liczbowy (-A,A), A=50
Wylosowana liczba 75 jest poza zadany zakres.

Press any key to continue . . .

```
10     printf("Zdefiniuj zakres liczbowy (-A,A), A=");
11     int A;
12     scanf("%i",&A);
13     srand(time(0));
14     int x = -2*A+rand()%(4*A+1);
15     if(test(A,x))
16         printf("Wylosowana liczba %i miesci sie w zadany zakresie.\n",x);
17     if(!test(A,x))
18         printf("Wylosowana liczba %i jest poza zadany zakresem.\n",x);
19 }
20 bool test(int A,int x)
21 {
22     if(x>=-A && x<=A)
23         return true;
24     return false;
25 }
```

Warunek dwustopniowy

- Stosując podstawową formę instrukcji warunkowej do zaprogramowania reakcji kodu na obie odpowiedzi warunku (**true** i **false**) wymaga zaprogramowania osobnych linii warunkowych dla każdej z odpowiedzi.
- W języku C dostępna jest rozbudowana wersja instrukcji warunkowej o słowo kluczowe **else**

```
if (warunek)
```

```
    instrukcja gdy TRUE;
```

```
else
```

```
    instrukcja gdy FALSE;
```

Przykład

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main()
6  {
7      printf("Podaj zakres liczbowy (-A,A), A=");
8      int A;
9      scanf("%i",&A);
10     srand(time(0));
11     int x = -2*A+rand()%(4*A+1);
12     if(abs(x)<=A)
13         printf("Wylosowana liczba %i zawiera sie w przedziale (-%i,%i).\n",x,A,A);
14     else
15         printf("Wylosowana liczba %i nie zawiera sie w przedziale (-%i,%i).\n",x,A,A);
16 }
```

```
C:\WINDOWS\system32\cmd.exe

Podaj zakres liczbowy (-A,A), A=25
Wylosowana liczba 27 nie zawiera sie w przedziale (-25,25).

Press any key to continue . . .
```

Kaskada warunków

- Wykorzystując postać dwustopniową instrukcji warunkowej możliwe jest uzyskanie kaskady
- W linii kodu dla sytuacji niespełnienia warunku w instrukcji **if** można umieścić kolejną instrukcję warunkową **if** z innym warunkiem logicznym
- W efekcie można opracować warunek testujący szereg warunków i dzięki temu zrealizować złożony filtr decyzyjny

Kaskada warunków

- Ogólna postać kaskady warunkowej

```
if (warunek_1)
    instrukcja gdy TRUE;
else if (warunek_2)
    instrukcja gdy TRUE;
else if (warunek_3)
    instrukcja gdy TRUE;
else
    instrukcja gdy FALSE;
```

Przykład

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6  //printf("program wyznacza pierwiastki równania kwadratowego  $Ax^2+Bx+C=0$ \n");
7  printf("Wprowadz rownanie w podanej postaci: ");
8  float A,B,C;
9  scanf("%fx^2%fx%f",&A,&B,&C);
10 printf("A=%.2f, B=%.2f, C=%.2f",A,B,C);
11 return 0;
12 }
```

Trojan:Win32/Swrort.A

20.09.2021 09:02 (Aktywne)

Poważny

C:\WINDOWS\system32\cmd.exe

```
Wprowadz rownanie w podanej postaci: -36.23x^2-25.01x+36.14
A=-36.23, B=-25.01, C=36.14
Press any key to continue . . .
```

Dozwolone zagrożenia

Historia ochrony

Przykład

```
1 #include <stdio.h>
2 #include <stdlib.h>
```

C:\WINDOWS\system32\cmd.exe

Program oblicza pierwiastki rownania kwadratowego $Ax^2+Bx+C=0$.
Wprowadz rownanie w podanej postaci: $25x^2-12.5x+5$
Delta jest ujemna, wynosi -343.75 , rownanie nie ma pierwiastkow rzeczywistych.
Press any key to continue . . .

C:\WINDOWS\system32\cmd.exe

Program oblicza pierwiastki rownania kwadratowego $Ax^2+Bx+C=0$.
Wprowadz rownanie w podanej postaci: $1x^2+10x+25$
Rownanie ma jedno miejsce zerowe $x=-5.00$.
Press any key to continue . . .

C:\WINDOWS\system32\cmd.exe

Program oblicza pierwiastki rownania kwadratowego $Ax^2+Bx+C=0$.
Wprowadz rownanie w podanej postaci: $25x^2+12.5x-5$
Rownanie ma dwa miejsca zerowe $x_1=0.26$ i $x_2=-0.76$.
Press any key to continue . . .

Przykład

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4  #include <complex.h>
5  |
6  int main()
7  {
8  printf("Program oblicza pierwiastki rownania kwadratowego Ax^2+Bx+C=0.\n");
9  printf("Wprowadz rownanie w podanej postaci: ");
10 float A,B,C;
11 scanf("%fx^2%fx%f",&A,&B,&C);
12 float delta = B*B-4*A*C;
13 if(delta<0)
14 {
15     printf("Delta jest ujemna, wynosi %.2f, rownanie nie ma pierwiastkow rzeczywistych.\n",delta);
16     printf("Pierwiastek kwadratowy z delty wynosi: %.2fi.\n",cimag(csqrt(delta)));
17     complex float x1 = (-B+csqrt(delta))/(2*A);
18     complex float x2 = (-B-csqrt(delta))/(2*A);
19     printf("Rownanie ma pierwiastki zespolone:\n");
20     printf("x1=%.2f%+.2fi\n",creal(x1),cimag(x1));
21     printf("x2=%.2f%+.2fi\n",creal(x2),cimag(x2));
22 }
```

```
C:\WINDOWS\system32\cmd.exe
Program oblicza pierwiastki rownania kwadratowego Ax^2+Bx+C=0.
Wprowadz rownanie w podanej postaci: 25x^2-12.5x+5
Delta jest ujemna, wynosi -343.75, rownanie nie ma pierwiastkow rzeczywistych.
Pierwiastek kwadratowy z delty wynosi: 18.54i.
Rownanie ma pierwiastki zespolone:
x1=0.25+0.37i
x2=0.25-0.37i

Press any key to continue . . .
```

Warunki złożone

- Proste warunki budowane są w oparciu o operatory relacji
- Warunki złożone budowane są w oparciu o relacje proste operatory logiczne

```
if (x>0 && x<10)
```

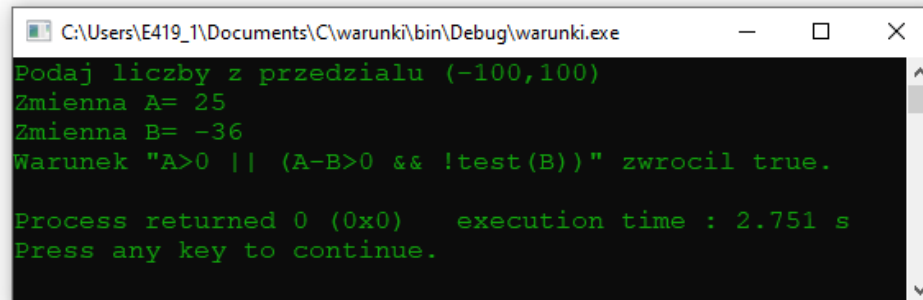
```
if (x+y>10 || (z<25 && z>12))
```

```
if (!test(x) || test(y))
```

A	B	A && B	A B
TRUE	TRUE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE
FALSE	TRUE	FALSE	TRUE
FALSE	FALSE	FALSE	FALSE

Przykład

```
1  #include <stdio.h>
2  #include <stdbool.h>
3
4  bool test(int);
5
6  int main()
7  {
8      int A, B;
9      printf("Podaj liczby z przedzialu (-100,100)\n");
10     printf("Zmienna A= ");
11     scanf("%i",&A);
12     printf("Zmienna B= ");
13     scanf("%i",&B);
14     if(A*B>0 && A>0)
15         printf("Warunek \'A*B>0 && A>0\' zwrocil true.\n");
16     else if(test(A*B))
17         printf("Funkcja test(A*B) zwrocila true.\n");
18     else if(A>0 || (A-B>0 && !test(B)))
19         printf("Warunek \'A>0 || (A-B>0 && !test(B))\' zwrocil true.\n");
20     else
21         printf("Warunek \'A>0 || (A-B>0 && !test(B))\' zwrocil true.\n");
22 }
23
24 bool test(int x)
25 {
26     if (x>=1)
27         return true;
28     return false;
29 }
```

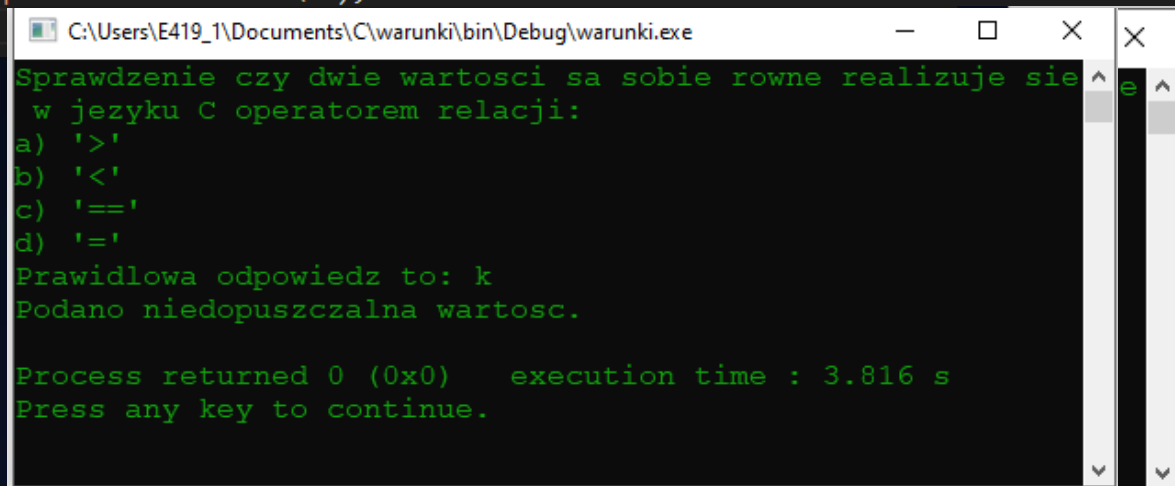


```
C:\Users\E419_1\Documents\C\warunki\bin\Debug\warunki.exe
Podaj liczby z przedzialu (-100,100)
Zmienna A= 25
Zmienna B= -36
Warunek "A>0 || (A-B>0 && !test(B))" zwrocil true.

Process returned 0 (0x0)   execution time : 2.751 s
Press any key to continue.
```

Wybór

```
1  #include <stdio.h>
2
3  int main()
4  {
5      printf("Sprawdzenie czy dwie wartosci sa sobie rowne realizuje sie w jezyku C operatorem relacji:\n");
6      printf("a) '>'\n");
7      printf("b) '<'\n");
8      printf("c) '=='\n");
9      printf("d) '='\n");
10     char x;
11     printf("Prawidlowa odpowiedz to: ");
12     scanf("%c",&x);
13     if(x=='a')
14     |     printf("Odpowiedz bledna!\n");
15     else if(x=='b')
16     |     printf("Odpowiedz bledna!\n");
17     else if(x=='c')
18     |     printf("Odpowiedz prawidlowa!\n");
19     else if(x=='d')
20     |     printf("Odpowiedz bledna!\n");
21     else
22     |     printf("Podano niedopuszczalna wartosc.\n");
23 }
```



```
C:\Users\E419_1\Documents\C\warunki\bin\Debug\warunki.exe
Sprawdzenie czy dwie wartosci sa sobie rowne realizuje sie
w jezyku C operatorem relacji:
a) '>'
b) '<'
c) '=='
d) '='
Prawidlowa odpowiedz to: k
Podano niedopuszczalna wartosc.

Process returned 0 (0x0)    execution time : 3.816 s
Press any key to continue.
```

Instrukcja wyboru

- Instrukcja wyboru switch .. case umożliwia zaprogramowanie wielu reakcji kodu w zależności od wartości wyrażenia sterującego

```
switch (wyrażenie) {  
    case wartość_1:  
        kod  
    case wartość_2:  
        kod  
    default:  
        kod  
}
```

Instrukcja wyboru

- **switch** działa jak przełącznik który wymusza przeskoczenie z realizacją kodu do wiersza z **case** powiązanym z wartością jaką przyjmie wyrażenie w linii **switch**
- Jeżeli żadna z linii **case** nie jest równa wartości wyrażenia, „Focus” przeskakuje do linii **default**
- Wyrażenie w linii **switch** może być zmienną, działaniem na zmiennych lub wywołaniem funkcji
- Wartości przyjmowane przez wyrażenie w linii **switch** mogą być typu **int**, **char**, **bool** itp..

Instrukcja wyboru

- Po wykonaniu przeskoku z linii **switch** do właściwego wiersza **case** program wykonuje kod od tego miejsca
- Powoduje to że poza kodem powiązanim z daną linią **case** wykonane będą także pozostałe linie kodu z bloku **switch**
- Konieczne jest więc zapewnienie przerwania wykonywania kodu bloku **switch** .. **case** przed wykonaniem kodu przypisanego do kolejnej linii **case**

Instrukcja wyboru

- Zadanie to realizuje funkcja przerwania bloku – **break**
- W wyniku jej wywołania przerywane jest wykonywanie aktualnego bloku kodu ograniczonego **{ }**
- Program przeskakuje do pierwszego wiersza kodu po przerwanym bloku

Przykład

```
1 1 #include <stdio.h>
2 2
3 3 int main()
4 4 {
5 5     printf("Wykonanie wybranego kodu, wybierz (a,b,c lub d): ");
6 6     char x;
```

C:\Windows\system32\cmd.exe

Wykonanie wybranego kodu, wybierz (a,b,c lub d): a
Wybrano a.

Press any key to continue . . .

```
Nie 14         break;
Pre 15     case 'c':
16         printf("Wybrano c.\n");
17         break;
18     case 'd':
19         printf("Wybrano d.\n");
20         break;
21     default:
22         printf("Nie wybrano zadnej z podanych liter.\n");
23 }
24 }
```

Instrukcja wyboru

```
1  #include <stdio.h>  
2
```

C:\Windows\system32\cmd.exe

Podaj wybrana litere alfabetu (a-z): f
Wybrano spolgloske.

Press any key to continue . . .

C:\Windows\system32\cmd.exe

Podaj wybrana litere alfabetu (a-z): u
Wybrano samogloske

Press any key to continue . . .

Operator warunkowy

- W języku C poza funkcjami warunkową i wyboru decyzję w kodzie można zaprogramować za pomocą operatora warunkowego ?
- Funkcjonalnie działa on tak samo jak instrukcja warunkowa **if**
- Sprawdzany jest warunek realizowany jest kod gry jest on spełniony lub nie
- Jest on stosowany wtedy gdy kod wykonywalny jest ograniczony do krótkiej, pojedynczej linii kodu

Operator warunkowy

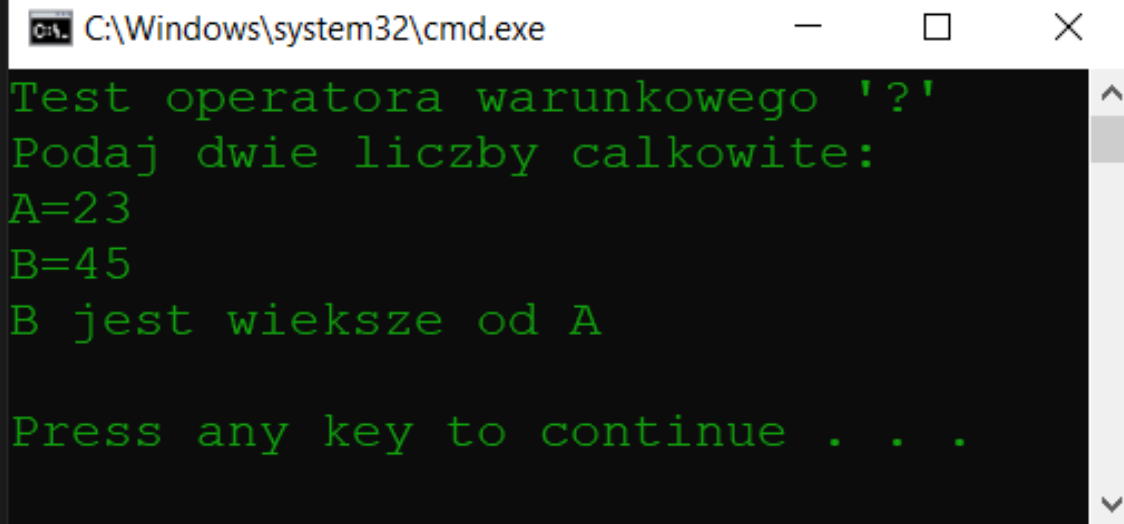
```
warunek ? Wyrażenie_TRUE:  
        Wyrażenie_FALSE;
```

Podobnie jak w przypadku instrukcji **if** możliwe jest zbudowanie kaskady warunków

```
warunek_1 ? W_TRUE: warunek_2 ?  
            W_TRUE: W_FALSE
```

Przykład

```
1  #include <stdio.h>
2
3  int main()
4  {
5      printf("Test operatora warunkowego '?'\n");
6      printf("Podaj dwie liczby calkowite:\n");
7      int a,b;
8      printf("A=");
9      scanf("%i",&a);
10     printf("B=");
11     scanf("%i",&b);
12     a>b ? printf("A jest wieksze od B\n") : printf("B jest wieksze od A\n");
13 }
```



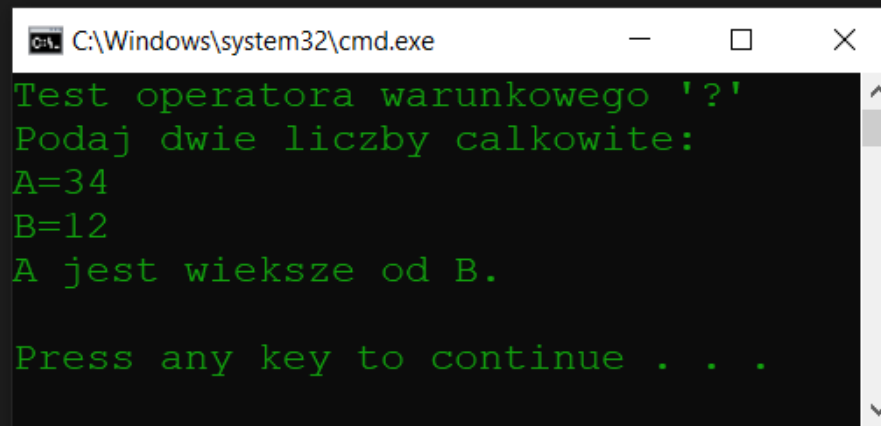
The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The output of the program is displayed in green text on a black background. The output matches the program's logic: it prints a test message, prompts for two integers, reads the values 23 and 45, and then prints "B jest wieksze od A" because 45 is greater than 23. The prompt "Press any key to continue . . ." is visible at the bottom.

```
C:\Windows\system32\cmd.exe
Test operatora warunkowego '?'
Podaj dwie liczby calkowite:
A=23
B=45
B jest wieksze od A

Press any key to continue . . .
```

Przykład

```
1  #include <stdio.h>
2
3  int main()
4  {
5      printf("Test operatora warunkowego '?'\n");
6      printf("Podaj dwie liczby calkowite:\n");
7      int a,b;
8      printf("A=");
9      scanf("%i",&a);
10     printf("B=");
11     scanf("%i",&b);
12     printf("%c jest wieksze od %c.\n",a>b?'A':'B',a>b?'B':'A');
13 }
```



```
C:\Windows\system32\cmd.exe
Test operatora warunkowego '?'
Podaj dwie liczby calkowite:
A=34
B=12
A jest wieksze od B.

Press any key to continue . . .
```

Zadanie

- Napisać program liczący silnię dla podanej liczby całkowitej

$$0! = 1$$

$$1! = 1$$

$$2! = 2$$

$$3! = 6$$

$$4! = 24 \dots$$

$$n! = \prod_{k=1}^n k \quad \forall n \in \mathbb{N}.$$

- Kod oprzeć na funkcji rekurencyjnej
- Funkcja rekurencyjna to funkcja która wywołuje samą siebie
- Silnię jakiej największej liczby można policzyć?

Zadanie

```
1  #include <stdio.h>
2  unsigned long long silnia(int);
3  int main()
4  {
5      printf("Program liczy silne dla podanej liczby.\n");
6      printf("Podaj liczbe: ");
7      int x;
8      scanf("%i",&x);
9      printf("%i!=%llu\n",x,silnia(x));
10 }
11 unsigned long long silnia(int x)
12 {
13     switch (x)
14     {
15         case 0:
16             return 1;
17         default:
18             return x*silnia(x-1);
19     }
20 }
```

Zadanie

```
C:\Windows\system32\cmd.exe
41!=15551764317513711616
42!=7538058755741581312
43!=10541877243825618944
44!=2673996885588443136
45!=9649395409222631424
46!=1150331055211806720
47!=17172071447535812608
48!=12602690238498734080
49!=8789267254022766592
50!=15188249005818642432
51!=18284192274659147776
52!=9994050523088551936
53!=13175843659825807360
54!=10519282829630636032
155!=6711489344688881664
156!=6908521828386340864
157!=6404118670120845312
158!=2504001392817995776
159!=162129586585337856
160!=9727775195120271360
161!=3098476543630901248
162!=7638104968020361216
163!=1585267068834414592
164!=9223372036854775808
165!=9223372036854775808
166!=0
167!=0
268!=0
Press any key to continue . . .
```

```
ej liczby.\n");
```

```
);
```

Zadanie

- Napisać program który określa czy podany rok jest przestępnym
- Rok jest przestępny jeżeli jest podzielny przez 4
- Rok nie jest przestępny jeżeli jest podzielny przez 100
- Rok jest przestępny jeżeli jest podzielny przez 400

Zadanie

```
C:\Windows\system32\cmd.exe

Program sprawdza czy podany rok jest rokiem przestepnym.

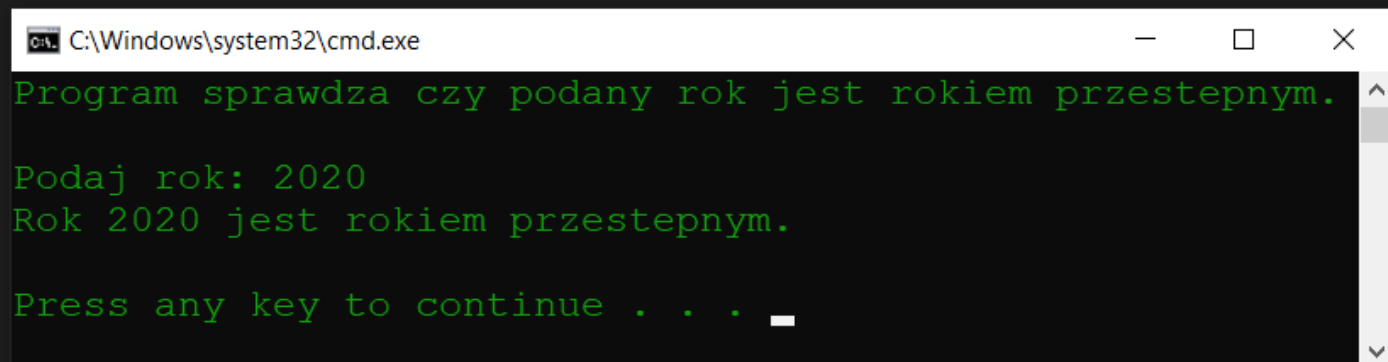
Podaj rok: 2022
Rok 2022 nie jest rokiem przestepnym.

Press any key to continue . . .
```

```
11         printf("Rok %i jest rokiem przestepnym.\n",data);
12     else if(data%100==0)
13         printf("Rok %i nie jest rokiem przestepnym.\n",data);
14     else
15         printf("Rok %i jest rokiem przestepnym.\n",data);
16 else
17     printf("Rok %i nie jest rokiem przestepnym.\n",data);
18 }
```

Zadanie

```
1  #include <stdio.h>
2
3  int main()
4  {
5      printf("Program sprawdza czy podany rok jest rokiem przestepnym.\n");
6      printf("Podaj rok: ");
7      int data;
8      scanf("%i",&data);
9      if((data%4==0 && data%100!=0) || data%400==0)
10         printf("Rok %i jest rokiem przestepnym.\n",data);
11     else
12         printf("Rok %i nie jest rokiem przestepnym.\n",data);
13 }
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The output of the program is displayed in green text on a black background. It shows the program's introduction, a prompt for a year, the user inputting "2020", and the program's response that 2020 is a leap year. The prompt "Press any key to continue . . ." is also visible.

```
C:\Windows\system32\cmd.exe
Program sprawdza czy podany rok jest rokiem przestepnym.

Podaj rok: 2020
Rok 2020 jest rokiem przestepnym.

Press any key to continue . . .
```



Zadanie

- Napisać program kalkulator
- Program pobiera ze standardowego wejścia dwie liczby i wykonuje na nich zadane działanie
- Wynik obliczeń wysyłany jest na standardowe wyjście
- Program umożliwia wykonanie działań – dodawania, odejmowania, mnożenia, dzielenia i potęgowania
- Program wykonuje działania na liczbach rzeczywistych
- Program sprawdza czy zadane działanie może zostać wykonane
- Napisać program w wersjach opartych na instrukcji warunkowej **if** i wyboru **switch..case**