



Wprowadzenie do informatyki

Jerzy Nawrocki
Wydział Informatyki
Politechnika Poznańska
jerzy.nawrocki@put.poznan.pl

Programowanie imperatywne i język C

Wprowadzenie do informatyki

Cel wykładu



Zaprezentować paradygmat programowania imperatywnego na przykładzie języka C.

Programowanie imperatywne i język C (2)

Wprowadzenie do informatyki

Imperatywność

łac. imperare
rozkazywać

Oktawian August
63 p.n.e – 14 n.e.
Pierwszy Imperator Rzymu

Programowanie imperatywne i język C (3)

Wprowadzenie do informatyki

Klasy języków programowania

Języki programowania

Imperatywne
Ciąg rozkazów

Deklaratywne
Opis problemu


Jak? Fortran, Algol, Pascal, C, C++, Java

Go? Prolog, ILOG

Programowanie imperatywne i język C (4)

Wprowadzenie do informatyki

Historia języka C

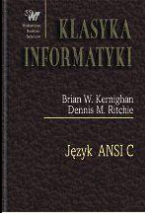


Dennis M. Ritchie

1966: Kompilator języka BCPL
1969: Język B
1972: Język C
Ken Thompson, UNIX

B.Kernighan, D.Ritchie,
The C Programming Language.


Lex, YACC, AWK
Stroustrup, C++, 1983



Programowanie imperatywne i język C (5)

Wprowadzenie do informatyki

Pisz „Bitwa pod Akcjum”.



- **Programy z Jedną instrukcją**
- **Koncepcja zmiennej**
- **Drukowanie wartości zmiennej**
- **Czytanie liczb**
- **Instrukcja warunkowa**
- **Uproszczona instrukcja warunkowa**
- **Instrukcja powtarzania 'while'**
- **Modularyzacja i funkcje**

Programowanie imperatywne i język C (6)

© Jerzy Nawrocki, Wprowadzenie do informatyki

Wprowadzenie do informatyki

Program z instrukcją drukowania

Język C

```
void main() {  
    printf ("Bitwa pod Akcjum\n");  
    return;  
}
```

Bitwa pod Akcjum

Pascal

```
begin  
    writeln ('Bitwa pod Akcjum')  
end.
```

Start
...
Stop

Programowanie imperatywne i język C (7)

Wprowadzenie do informatyki

Kompilacja programu i wykonanie obliczeń

1. Zapisanie programu w pliku Akc.c → 2. Kompilacja np. tcc Akc.c → Akc.exe

Dane wejściowe → 3. Wykonanie obliczeń przez Akc.exe → Wyniki Bitwa pod Akcjum

Programowanie imperatywne i język C (8)

Wprowadzenie do informatyki



Zapamiętaj sobie „31”.

- Programy z Jedną Instrukcją
- Koncepcja zmiennej
- Drukowanie wartości zmiennej
- Czytanie liczb
- Instrukcja warunkowa
- Uproszczona instrukcja warunkowa
- Instrukcja powtarzania 'while'
- Modularyzacja i funkcje

Programowanie imperatywne i język C (9)

Wprowadzenie do informatyki

Zmienna jako pamięć

Cel: Pamiętanie wartości (np. liczby)

Dwie operacje do wyboru:

- Zapisanie jakiejś wartości.
- Odczytanie wartości.

121 Zmienna

Zapisz 5
Zapisz 121
Odczytaj Jest 121

Programowanie imperatywne i język C (10)

Wprowadzenie do informatyki

Zmienna jako pamięć

W programie można używać wielu zmiennych. Dlatego trzeba je nazywać.

11	Zmienna X
22	Zmienna Y
33	Zmienna SUMA

Programowanie imperatywne i język C (11)

Wprowadzenie do informatyki

Zmienna jako pamięć

W programie można używać wielu zmiennych. Dlatego trzeba je nazywać.

18	Zmienna X
2	Zmienna Y
20	Zmienna SUMA

Zapisz w X 18
Zapisz w Y 2
Zapisz w SUMA sumę X+Y

Ukryty odczyt

Programowanie imperatywne i język C (12)

Wprowadzenie do informatyki

Zmienna jako pamięć

Język C

18	Zmienna X
2	Zmienna Y
20	Zmienna SUMA

```
void main() {
    ...
    X = 18;
    Y = 2;
    SUMA = X + Y;
    ...
}
```

Zapisz w X 18
Zapisz w Y 2
Zapisz w SUMA sumę X+Y

Programowanie imperatywne i język C (13)

Wprowadzenie do informatyki

Zmienna jako pamięć

Pascal

18	Zmienna X
2	Zmienna Y
20	Zmienna SUMA

```
begin
    ...
    X := 18;
    Y := 2;
    SUMA := X + Y;
    ...
end.
```

Zapisz w X 18
Zapisz w Y 2
Zapisz w SUMA sumę X+Y

Programowanie imperatywne i język C (14)

Wprowadzenie do informatyki

Typy zmiennych

Komputer inaczej przetwarza

- Liczby całkowite (...,-2, -1, 0, 1, 2, ..)
- Liczby rzeczywiste (-1.5, 0.5, 2.3, ..)
- Pojedyncze znaki ('a', 'b', 'z', 'A', 'Z', '+', '-', '*', '\$', '0', '9', ..)
- Ciągi znaków ("on", "rok 1956", ..)

Zmienna: **typ + nazwa**

Przykładowe typy zmiennych w języku C:

- int** – liczby całkowite (ang. integer)
- float – liczby rzeczywiste (ang. floating point)
- char – znaki (ang. character)

Pascal

integer
real
char

Programowanie imperatywne i język C (15)

Wprowadzenie do informatyki

Prosty program ze zmiennymi

```
void main() {
    ...
    return;
}
```

Deklaracje zmiennych

Instrukcje

Zmienna: **typ + nazwa**

Deklaracje zmiennych:
typ nazwy_oddzielone_przecinkami ;

Programowanie imperatywne i język C (16)

Wprowadzenie do informatyki

Zmienna jako pamięć

Kompletny program

```
void main() {
    ...
    X = 18;
    Y = 2;
    SUMA = X + Y;
    ...
}
```

```
void main() {
    int X, Y, SUMA;
    X = 18;
    Y = 2;
    SUMA = X + Y;
    return;
}
```

Deklaracje zmiennych:
typ nazwy_oddzielone_przecinkami ;

Programowanie imperatywne i język C (17)

Wprowadzenie do informatyki

Zmienna jako pamięć

Pascal

```
void main() {
    int X, Y, SUMA;
    X = 18;
    Y = 2;
    SUMA = X + Y;
    return;
}
```

```
var
    X, Y, SUMA: integer;
begin
    X := 18;
    Y := 2;
    SUMA := X + Y;
end.
```

Deklaracje zmiennych:
nazwy_oddzielone_przecinkami : typ;

Programowanie imperatywne i język C (18)

Wprowadzenie do informatyki

Pisz co zapamiętałeś.

- Programy z Jedną Instrukcją
- Koncepcja zmiennej
- Drukowanie wartości zmiennej
- Czytanie liczb
- Instrukcja warunkowa
- Uproszczona instrukcja warunkowa
- Instrukcja powtarzania 'while'
- Modularyzacja i funkcje

Programowanie imperatywne i język C (19)

Wprowadzenie do informatyki

Drukowanie wartości zmiennej

Kompletny program

```
void main() {
    int X, Y, SUMA;
    X = 18;
    Y = 2;
    SUMA = X + Y;
    return;
}
```

Nie widać wyników!

Programowanie imperatywne i język C (20)

Wprowadzenie do informatyki

Drukowanie wartości zmiennej

Ogólniejsza postać printf:

```
printf(" format ", zmienne);
```

%d oznacza int

```
printf("X = %d \n", X);
```

X = 18

Kompletny program

```
void main() {
    int X, Y, SUMA;
    X = 18;
    Y = 2;
    SUMA = X + Y;
    printf("%d \n", SUMA);
    return;
}
```

Programowanie imperatywne i język C (21)

Wprowadzenie do informatyki

Drukowanie wartości zmiennej

Pascal

```
var
    X, Y, SUMA: integer;
begin
    X := 18;
    Y := 2;
    SUMA := X + Y;
    writeln(SUMA);
end.
```

20

Programowanie imperatywne i język C (22)

Wprowadzenie do informatyki

Czytaj, co pisze Livia Drusilla

- Programy z Jedną Instrukcją
- Koncepcja zmiennej
- Drukowanie wartości zmiennej
- Czytanie liczb
- Instrukcja warunkowa
- Uproszczona instrukcja warunkowa
- Instrukcja powtarzania 'while'
- Modularyzacja i funkcje

Programowanie imperatywne i język C (23)

Wprowadzenie do informatyki

Zadanie

Napisać program wyświetlający sumę dowolnych dwóch liczb całkowitych.

18	2	→	18 + 2 = 20
0	5	→	0 + 5 = 5
-3	1	→	-3 + 1 = -2

Programowanie imperatywne i język C (24)

Wprowadzenie do informatyki

Czytanie liczb

Jak wczytać wartości X,Y?

```
void main() {
    int X, Y, SUMA;
    X = 18;
    Y = 2;
    SUMA = X + Y;
    printf("%d + %d = %d\n", X, Y, SUMA);
    return;
}
```

Programowanie imperatywne i język C (25)

Wprowadzenie do informatyki

Czytanie liczb

Jak wczytać wartości X,Y?

```
scanf("format", adresy_zmiennych);
```

```
scanf("%d", &X);
```

```
void main() {
    int X;
    scanf("%d", &X);
    printf("%d\n", X);
    return;
}
```

Programowanie imperatywne i język C (26)

Wprowadzenie do informatyki

Czytanie liczb

```
void main() {
    int X, Y, SUMA;
    X = 18;
    Y = 2;
    SUMA = X + Y;
    printf("%d + %d = %d\n", X, Y, SUMA);
    return;
}
```

Programowanie imperatywne i język C (27)

Wprowadzenie do informatyki

Czytanie liczb

```
0 5
```

```
void main() {
    int X, Y, SUMA;

    scanf("%d %d", &X, &Y);
    SUMA = X + Y;
    printf("%d + %d = %d\n", X, Y, SUMA);
    return;
}
```

```
0 + 5 = 5
```

Programowanie imperatywne i język C (28)

Wprowadzenie do informatyki

Czytanie liczb

```
0 5
```

```
var
    X, Y, SUMA: integer;
begin
    read(X, Y);
    SUMA := X + Y;
    writeln(X, ' + ', Y, ' = ', SUMA);
end.
```

Pascal

```
0 + 5 = 5
```

Programowanie imperatywne i język C (29)

Wprowadzenie do informatyki

Jeśli idzie > 1 legion, to wystaw 2; w przeciwnym razie wystaw 1.

- Programy z Jedną instrukcją
- Koncepcja zmiennej
- Drukowanie wartości zmiennej
- Czytanie liczb
- Instrukcja warunkowa
- Uproszczona instrukcja warunkowa
- Instrukcja powtarzania 'while'
- Modularyzacja i funkcje

Programowanie imperatywne i język C (30)

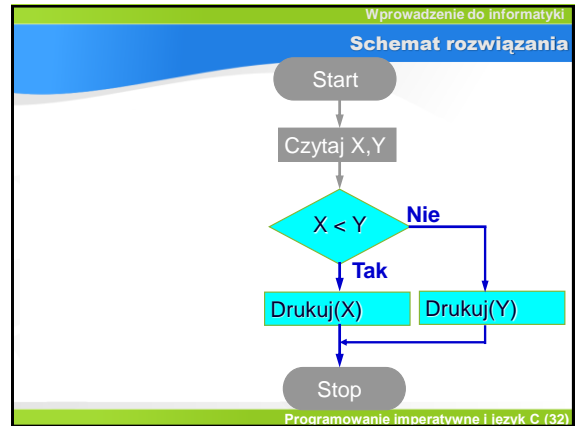
Wprowadzenie do informatyki

Zadanie

Napisać program wyświetlający minimum z dwóch liczb całkowitych.

18	2	→	min(18, 2) = 2
-3	1	→	min(-3, 1) = -3
5	5	→	min(5, 5) = 5

Programowanie imperatywne i język C (31)



Wprowadzenie do informatyki

Instrukcja warunkowa

```

if ( warunek ) {
    instrukcje_1
}
else {
    instrukcje_2
}

if ( ładna_pogoda ) {
    o_19tej_idziemy_na_spacer;
}
else {
    o_19tej_idziemy_do_kina;
}
  
```

Programowanie imperatywne i język C (33)

Wprowadzenie do informatyki

```

18 2
void main() {
    int X, Y;
    scanf("%d %d", &X, &Y);
    if ( X < Y ) {
        printf("min(%d, %d)= %d \n", X, Y, X);
    } else {
        printf("min(%d, %d)= %d \n", X, Y, Y);
    }
    return;
}
  
```

Programowanie imperatywne i język C (34)

Wprowadzenie do informatyki

```

18 2
var X, Y: integer;
begin
    read(X, Y);
    if ( X < Y ) then
        begin
            writeln('min(', X, ', ', Y, ') = ', X);
        end
    else
        begin
            writeln('min(', X, ', ', Y, ') = ', Y);
        end
    end.
min(18, 2) = 2
  
```

Pascal

Programowanie imperatywne i język C (35)

Wprowadzenie do informatyki



Jeśli Marek Antoniusz nie ustąpi, to wypowiemy wojnę Egipcjom.

- Programy z Jedną instrukcją
- Koncepcja zmiennej
- Drukowanie wartości zmiennej
- Czytanie liczb
- Instrukcja warunkowa
- Uproszczona instrukcja warunkowa
- Instrukcja powtarzania 'while'
- Modularyzacja i funkcje

Programowanie imperatywne i język C (36)

Wprowadzenie do informatyki

Zadanie

Napisać program wyświetlający wartość bezwzględną podanej liczby całkowitej.

3 → 3
-3 → 3

Programowanie imperatywne i język C (37)

Wprowadzenie do informatyki

Schemat rozwiązania

```

    graph TD
      Start(( )) --> Read[Czytaj X]
      Read --> Cond{X < 0}
      Cond -- Tak --> Negate[X ← -X]
      Negate --> Print[Drukuj X]
      Cond -- Nie --> Print
      Print --> End(( ))
    
```

Programowanie imperatywne i język C (38)

Wprowadzenie do informatyki

Uproszczona instrukcja warunkowa

```

    if (warunek) {
        instrukcje
    }

    if (pożar) {
        dzwonPoStrazPozarna;
    }
    
```

Programowanie imperatywne i język C (39)

Wprowadzenie do informatyki

Uproszczona instrukcja warunkowa

X

```

    3
    void main() {
        int X;
        scanf("%d", &X);
        if (X < 0) {
            X = -X;
        }
        printf("%d\n", X);
        return;
    }
    
```

Programowanie imperatywne i język C (40)

Wprowadzenie do informatyki

Relacje w języku C

1 5

```

    void main() {
        int a, b;
        scanf("%d %d", &a, &b);
        if (a < b) {printf("%d mniejsze niz %d\n", a, b);}
        if (a <= b) {printf("%d mniejsze lub rowne %d\n", a, b);}
        if (a == b) {printf("%d rowne %d\n", a, b);}
        if (a != b) {printf("%d rozne od %d\n", a, b);}
        if (a >= b) {printf("%d wieksze lub rowne %d\n", a, b);}
        if (a > b) {printf("%d wieksze niz %d\n", a, b);}
        return;
    }
    
```

Programowanie imperatywne i język C (41)

Wprowadzenie do informatyki

Jak długo Marek Antoniusz jest z Kleopatrami, będziemy walczyć.

- Programy z jedną instrukcją
- Koncepcja zmiennej
- Drukowanie wartości zmiennej
- Czytanie liczb
- Instrukcja warunkowa
- Uproszczona instrukcja warunkowa
- Instrukcja powtarzania 'while'
- Modularyzacja i funkcje

Programowanie imperatywne i język C (42)

Wprowadzenie do informatyki

Zadanie

Napisać program obliczający liczbę cyfr dziesiętnych podanej liczby naturalnej (0, 1, 2, ..).

0	→	0 ma cyfr: 1
9	→	9 ma cyfr: 1
10	→	10 ma cyfr: 2
99	→	99 ma cyfr: 2
100	→	100 ma cyfr: 3

Programowanie imperatywne i język C (43)

Wprowadzenie do informatyki

0, .. 9, 1 cyf. $X < 10$	10, .. 99, 2 cyf. $X < 100$	100, .. 999, 3 cyfry $X < 1000$	1 000, .. 9 999, 4 cyfry $X < 10000$	10 000, ..
--------------------------------	-----------------------------------	---------------------------------------	--	------------

Programowanie imperatywne i język C (44)

Wprowadzenie do informatyki

Przykładowe obliczenie

Jak to sprawdzić?

Programowanie imperatywne i język C (45)

Wprowadzenie do informatyki

Instrukcja powtarzania

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

```
while (X >= G) {
    C = C + 1;
    G = G * 10;
}
```

Programowanie imperatywne i język C (46)

Wprowadzenie do informatyki

Program w C

```
void main() {
    int X, C, G;
    scanf("%d", &X);
    C = 1; G = 10;

    Kompilacja

    return;
}
```

Programowanie imperatywne i język C (47)

Wprowadzenie do informatyki

Program w C

```
void main() {
    int X, C, G;
    scanf("%d", &X);
    C = 1; G = 10;
    while (X >= G) {
        C = C + 1;
        G = G * 10;
    }
    printf("%d ma cyfr: %d\n", X, C);
    return;
}
```

Programowanie imperatywne i język C (48)

Wprowadzenie do informatyki

87

```

var
  X, C, G: integer;
  read(X);
  C:= 1; G:= 10;
  while (X >= G) do
  begin
    C= C + 1;
    G= G * 10;
  end;
  writeln(X, ' ma cyfr: ', C);
end.
    
```

Pascal

87 ma cyfr: 2

Programowanie imperatywne i język C (49)

Wprowadzenie do informatyki


Plan wykładu

- Programy z Jedną Instrukcją
- Koncepcja zmiennej
- Drukowanie wartości zmiennej
- Czytanie liczb
- Instrukcja warunkowa
- Uproszczona instrukcja warunkowa
- Instrukcja powtarzania 'while'
- Modularyzacja i funkcje

Programowanie imperatywne i język C (50)

Wprowadzenie do informatyki

Koncepcja modularyzacji



```

graph TD
    A[System wahadłowca] --> B[Rakieta ET]
    A --> C[Discovery]
    C --> D[Monitor]
    
```

Start wahadłowca Discovery (26.VII.2005)

Programowanie imperatywne i język C (51)

Wprowadzenie do informatyki

Koncepcja modularyzacji

```

graph TD
    A[tg(x)] --> B[sin(x)]
    A --> C[cos(x)]
    
```

$tg(x) = \sin(x) / \cos(x)$

Programowanie imperatywne i język C (52)

Wprowadzenie do informatyki

Funkcje

$\cos(\pi/2) = 0$

```

graph TD
    A["cos(π/2) = 0"] --> B[Nazwa funkcji]
    A --> C["Argument (parametr aktualny)"]
    A --> D[Wynik]
    
```

Programowanie imperatywne i język C (53)

Wprowadzenie do informatyki

Funkcje

Napisać program wyświetlający liczbę o 1 większą od podanej.

```

3
void main() {
  int X, Y;
  scanf("%d", &X);
  Y= X + 1;
  printf("%d\n", Y);
  return;
}
4
    
```

$nast(3) = 4$

```

int nast(int n) {
  return n+1;
}

void main() {
  int X, Y;
  scanf("%d", &X);
  Y= nast(X);
  printf("%d\n", Y);
  return;
}
    
```

Programowanie imperatywne i język C (54)

Wprowadzenie do informatyki

Funkcje

Deklaracja f.

```
int nast(int n) {
    return n+1;
}
```

Wywołanie f.

```
void main() {
    int X, Y;
    scanf("%d", X);
    Y= nast(X);
    printf("%d\n", Y);
    return;
}
```

Programowanie imperatywne i język C (55)

Wprowadzenie do informatyki

Funkcje

Parametr formalny

```
int nast(int n) {
    return n+1;
}
```

Parametr aktualny

```
void main() {
    int X, Y;
    scanf("%d", X);
    Y= nast(X);
    printf("%d\n", Y);
    return;
}
```

Programowanie imperatywne i język C (56)

Wprowadzenie do informatyki

Funkcje

```
int nast(int n) {
    return n+1;
}
```

```
void main() {
    int X, Y;
    scanf("%d", X);
    Y= nast(X);
    printf("%d\n", Y);
    return;
}
```

Programowanie imperatywne i język C (57)

Wprowadzenie do informatyki

Funkcje

```
int nast( int n){
    return n+1 ;
}
```

TypWyniku NazwaFun (Param){
Deklaracje
Instrukcje
return Wynik;
}

Programowanie imperatywne i język C (58)

Wprowadzenie do informatyki

Funkcje

Program = zbiór funkcji
 Wykonanie zaczyna się od funkcji main

```
int nast( int n ){
    return n+1 ;
}
```

```
void main() {
    int X, Y;
    scanf("%d", X);
    Y= nast(X);
    printf("%d\n", Y);
    return;
}
```

Programowanie imperatywne i język C (59)

Wprowadzenie do informatyki

Funkcje

Program = zbiór funkcji
 Wykonanie zaczyna się od funkcji main

Żaden

Brak parametrów

```
void main() {
    printf("Fly by LotIn");
    return;
}
```

Brak wyniku

Programowanie imperatywne i język C (60)

Wprowadzenie do informatyki

```

void main() {
    int X, C, G;
    scanf("%d", &X);
    C= 1; G= 10;
    while (X >= G) {
        C= C + 1;
        G= G * 10;
    }
    printf("%d ma cyfr: %d ln", X, C);
    return;
}

void main() {
    int X, C;
    scanf("%d", &X);
    C= LiczbaCyfr(X);
    printf("%d ma cyfr: %d ln", X, C);
    return;
}

int LiczbaCyfr(int n) {
    int C, G;
    C= 1; G= 10;
    while (n >= G) {
        C= C + 1;
        G= G * 10;
    }
    return C;
}
    
```

Programowanie imperatywne i język C (62)

Wprowadzenie do informatyki

Wszystkie funkcje w jednym pliku

```

int LiczbaCyfr(int n) {
    int C, G;
    ...
    return C;
}

void main() {
    int X, C;
    scanf("%d", &X);
    C= LiczbaCyfr(X);
    printf("%d ma cyfr: %d ln", X, C);
    return;
}
    
```

Programowanie imperatywne i język C (62)

Wprowadzenie do informatyki

Funkcje w Pascalu

```

function LiczbaCyfr(n: integer): integer;
var C, G: integer;
begin
    ...
    LiczbaCyfr:= C;
end;

var X, C: integer;
begin
    read(X);
    C:= LiczbaCyfr(X);
    writeln(X, ' ma cyfr: ', C);
end.
    
```

Programowanie imperatywne i język C (63)

Wprowadzenie do informatyki

Plan wykładu



Podsumowanie

Programowanie imperatywne i język C (64)

Wprowadzenie do informatyki

Podsumowanie

Operacje wejścia-wyjścia	
scanf / printf	read / writeln
Instrukcje złożone	
if (...) { ... } else { ... }	if ... then ... else ...
while (...) { ... }	while ... do...
Wyrażenia logiczne	
a == b	a = b
a != b	a <> b
Instrukcja przypisania (zapamiętania)	
a = b;	a := b

Programowanie imperatywne i język C (65)

Wprowadzenie do informatyki

Podsumowanie

Podprogramy (funkcje) – Deklaracja	
int F (int x) { int LokalnyWynik; ... // instrukcje return LokalnyWynik; }	function F (x: integer): integer; var LokalnyWynik: integer; begin ... (* instrukcje *) F:= LokalnyWynik; end;
Podprogramy (funkcje) – Wywołanie	
Zmienna = 1 + F(2*a);	Zmienna := 1 + F(2*a);

Programowanie imperatywne i język C (66)

Wprowadzenie do informatyki

Programowanie sterowane testami

1. Testy (5, 1)
2. Program
3. Kompilacja
4. Wykonanie

```

5
void main() {
    int X, C, G;
    scanf("%d", &X);
    C= 1; G= 10;

    printf("%d ma cyfr: %d \n", X, C);
    return;
}
                    
```

5 ma cyfr: 1

Wprowadzenie do informatyki

Programowanie sterowane testami

1. Testy (87, 2)
2. Wykonanie

```

87
void main() {
    int X, C, G;
    scanf("%d", &X);
    C= 1; G= 10;

    printf("%d ma cyfr: %d \n", X, C);
    return;
}
                    
```

87 ma cyfr: 1

Wprowadzenie do informatyki

Programowanie sterowane testami

3. Program
4. Kompilacja
5. Wykonanie

```

87
void main() {
    int X, C, G;
    scanf("%d", X);
    C= 1; G= 10;
    while (X >= G) {
        C= C + 1;
        G= G * 10;
    }
    printf("%d ma cyfr: %d \n", X, C);
    return;
}
                    
```

87 ma cyfr: 1

Wprowadzenie do informatyki

Symboliczne wykonanie schematu

Jak to sprawdzić?

```

9
Czytaj X
C ← 1
G ← 10
X >= G?
    Nie → Drukuj C
    Tak → C ← C + 1, G ← G * 10 → X >= G?
                    
```

X	C	G
9		

Programowanie imperatywne i język C (70)

Wprowadzenie do informatyki

Symboliczne wykonanie programu

```

9
void main() {
    int X, C, G;
    scanf("%d", X);
    C= 1; G= 10;
    while (X >= G) {
        C= C + 1;
        G= G * 10;
    }
    printf("%d ma cyfr: %d \n", X, C);
    return;
}
                    
```

X	C	G
9		

Programowanie imperatywne i język C (71)

