

Programowanie w języku C (EAR1S02005)

Politechnika Białostocka - Wydział Elektryczny
Automatyka i Robotyka, semestr II, studia stacjonarne I stopnia
Rok akademicki 2020/2021

Zajęcia nr 4 (03.03.2021)

dr inż. Jarosław Forenc

Przykład: suma kolejnych 10 liczb

```
#include <stdio.h>

int main(void)
{
    int suma1;

    suma1 = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10;
    printf("Suma wynosi: %d\n", suma1);

    int suma2 = 0, i;

    for (i=1; i<=10; i=i+1)
        suma2 = suma2 + i;
    printf("Suma wynosi: %d\n", suma2);

    return 0;
}
```

Suma wynosi: 55
Suma wynosi: 55

Język C - pętla for

- Najczęściej stosowana postać pętli **for**

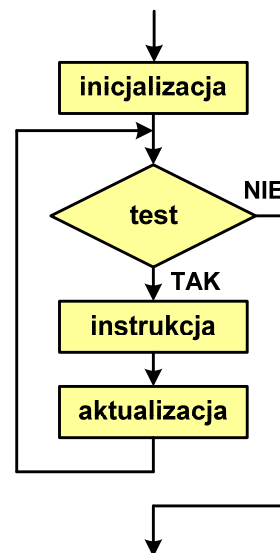
```
int i;
for (i=0; i<10; i=i+1)
    instrukcja;
```

- Instrukcja:

- prosta - jedna instrukcja zakończona średnikiem
- złożona - jedna lub kilka instrukcji objętych nawiasami klamrowymi

- Funkcje pełnione przez wyrażenia

```
for (inicjalizacja; test; aktualizacja)
    instrukcja;
```



Język C - pętla for (przykłady)

```
for (i=0; i<10; i++)
    printf("%d ", i);
```

0 1 2 3 4 5 6 7 8 9

```
for (i=0; i<10; i++)
    printf("%d ", i+1);
```

1 2 3 4 5 6 7 8 9 10

```
for (i=1; i<=10; i++)
    printf("%d ", i);
```

1 2 3 4 5 6 7 8 9 10

Język C - pętla for (przykłady)

```
for (i=1; i<10; i=i+2)  
printf("%d ",i);
```

1 3 5 7 9

```
for (i=10; i>0; i--)  
printf("%d ",i);
```

10 9 8 7 6 5 4 3 2 1

```
for (i=-9; i<=9; i=i+3)  
printf("%d ",i);
```

-9 -6 -3 0 3 6 9

Język C - pętla for (break, continue)

- W pętli **for** można stosować instrukcje skoku: **break** i **continue**

```
int i;  
for (i=1; i<10; i++)  
{  
    if (i%2==0)  
        continue;  
    if (i%7==0)  
        break;  
    printf("%d\n",i);  
}
```

- continue** przerywa bieżącą iterację i przechodzi do obliczania **wyr3**

- break** przerywa wykonywanie pętli

1 3 5

Język C - pętla for (najczęstsze błędy)

- Postawienie średnika na końcu pętli **for**

```
int i;  
for (i=0; i<10; i++);  
printf("%d ",i);
```

10

- Przecinki zamiast średników pomiędzy wyrażeniami

```
int i;  
for (i=0, i<10, i++)  
printf("%d ",i);
```

Błąd kompilacji!

Język C - pętla for (najczęstsze błędy)

- Błędny warunek - brak wykonania instrukcji

```
int i;  
for (i=0; i>10; i++)  
printf("%d ",i);
```

- Błędny warunek - pętla nieskończona

```
int i;  
for (i=1; i>0; i++)  
printf("%d ",i);
```

1 2 3 4 5 6 7 8 9 ...

Język C - pętla nieskończona

```
for (wyr1; wyr2; wyr3)  
instrukcja
```

- Wszystkie wyrażenia (**wyr1**, **wyr2**, **wyr3**) w pętli for są opcjonalne

```
for ( ; ; )  
instrukcja
```

- pętla nieskończona

- W przypadku braku **wyr2** przyjmuje się, że jest ono **prawdziwe**

Język C - pętla while

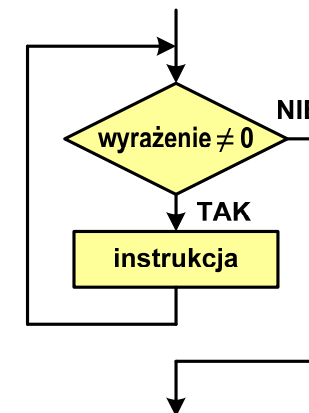
```
while (wyrażenie)  
instrukcja;
```

- „dopóki wyrażenie w nawiasach jest prawdziwe wykonuj instrukcję”

- Wyrażenie w nawiasach:

- prawdziwe** - gdy jego wartość jest różna od zera
- falszywe** - gdy jego wartość jest równa zero

- Jako wyrażenie najczęściej stosowane jest **wyrażenie logiczne**



Język C - pętla while

```
while (wyrażenie)  
instrukcja
```

- Instrukcja:
 - prosta** - jedna instrukcja zakończona średnikiem
 - złożona** - jedna lub kilka instrukcji objętych nawiasami klamrowymi

```
int x = 10;  
while (x>0)  
  x = x - 1;
```

```
int x = 10;  
while (x>0)  
{  
  printf("%d\n", x);  
  x = x - 1;  
}
```

Przykład: suma liczb dodatnich

```
#include <stdio.h>  
#include <math.h>  
  
int main(void)  
{  
  int x, suma = 0;  
  
  printf("Podaj liczbę: ");  
  scanf("%d", &x);  
  
  while (x>0)  
  {  
    suma = suma + x;  
    printf("Podaj liczbę: ");  
    scanf("%d", &x);  
  }  
  
  printf("Suma liczb: %d\n", suma);  
  
  return 0;  
}
```

```
Podaj liczbę: 4  
Podaj liczbę: 8  
Podaj liczbę: 2  
Podaj liczbę: 3  
Podaj liczbę: 5  
Podaj liczbę: -2  
Suma liczb: 22
```

Język C - pętla while

- Program pokazany na poprzednim slajdzie zawiera typowy schemat przetwarzania danych z wykorzystaniem pętli **while**

```
printf("Podaj liczbę: ");  
scanf("%d", &x);  
  
while(x>0)  
{  
    suma = suma + x;  
    printf("Podaj liczbę: ");  
    scanf("%d", &x);  
}
```

wczytanie danych

operacje na danych

wczytanie danych

- Dane mogą być wczytywane z klawiatury, pliku, itp.

Język C - pętla while (najczęstsze błędy)

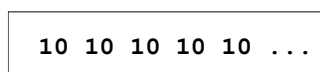
- Postawienie średnika po wyrażeniu w nawiasach powoduje powstanie pętli nieskończonej - program zatrzymuje się na pętli

```
int x = 10;  
while (x>0);  
    printf("%d ", x--);
```



- Brak aktualizacji zmiennej powoduje także powstanie pętli nieskończonej - program wyświetla wielokrotnie tę samą wartość

```
int x = 10;  
while (x>0)  
    printf("%d ", x);
```



Język C - pętla while (break, continue)

- **break** i **continue** są to instrukcje skoku

```
int x=0;  
while (x<10)  
{  
    x++;  
    if (x%2==0)  
        continue;  
    if (x%5==0)  
        break;  
    printf("%d\n", x);  
}
```

- **continue** przerywa bieżącą iterację

- **break** przerywa wykonywanie pętli

Język C - pętla while (pętla nieskończona)

- W pewnych sytuacjach celowo stosuje się pętlę nieskończoną (np. w mikrokontrolerach)

```
while (1)  
{  
    instrukcja;  
    instrukcja;  
    ...  
}
```

- W układach mikroprocesorowych program działa aż do wyłączenia zasilania

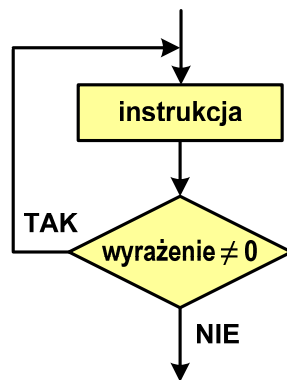
Język C - pętla do ... while

```
do
    instrukcja
while (wyrażenie);
```

- „wykonuj instrukcję dopóki wyrażenie w nawiasach jest prawdziwe”

- Wyrażenie w nawiasach:

- prawdziwe** - gdy jego wartość jest różna od zera
- falszywe** - gdy jego wartość jest równa zero



Język C - pętla do ... while

```
do
    instrukcja
while (wyrażenie);
```

- Instrukcja:

- prosta** - jedna instrukcja zakończona średnikiem
- złożona** - jedna lub kilka instrukcji objętych nawiasami klamrowymi

```
int x = 10;
do
    x = x - 1;
while (x>0);
```

```
int x = 10;
do
{
    printf("%d\n", x);
    x = x - 1;
}
while (x>0);
```

Język C - pętla do ... while (break, continue)

- break** i **continue** są to instrukcje skoku

```
int x=0;

do
{
    x++;
    if (x%5==0)
        break;
    if (x%2==0)
        continue;
    printf("%d\n", x);
}
while (i<10);
```

- break** przerywa wykonywanie pętli
- continue** przerywa bieżącą iterację

Przykład: suma liczb < 100

```
#include <stdio.h>

int main(void)
{
    int x, suma = 0;

    do
    {
        printf("Podaj liczbę: ");
        scanf("%d", &x);
        suma = suma + x;
    }
    while (suma<100);

    printf("Suma liczb: %d\n", suma);

    return 0;
}
```

```
Podaj liczbę: 34
Podaj liczbę: 9
Podaj liczbę: 26
Podaj liczbę: -8
Podaj liczbę: 67
Suma liczb: 128
```