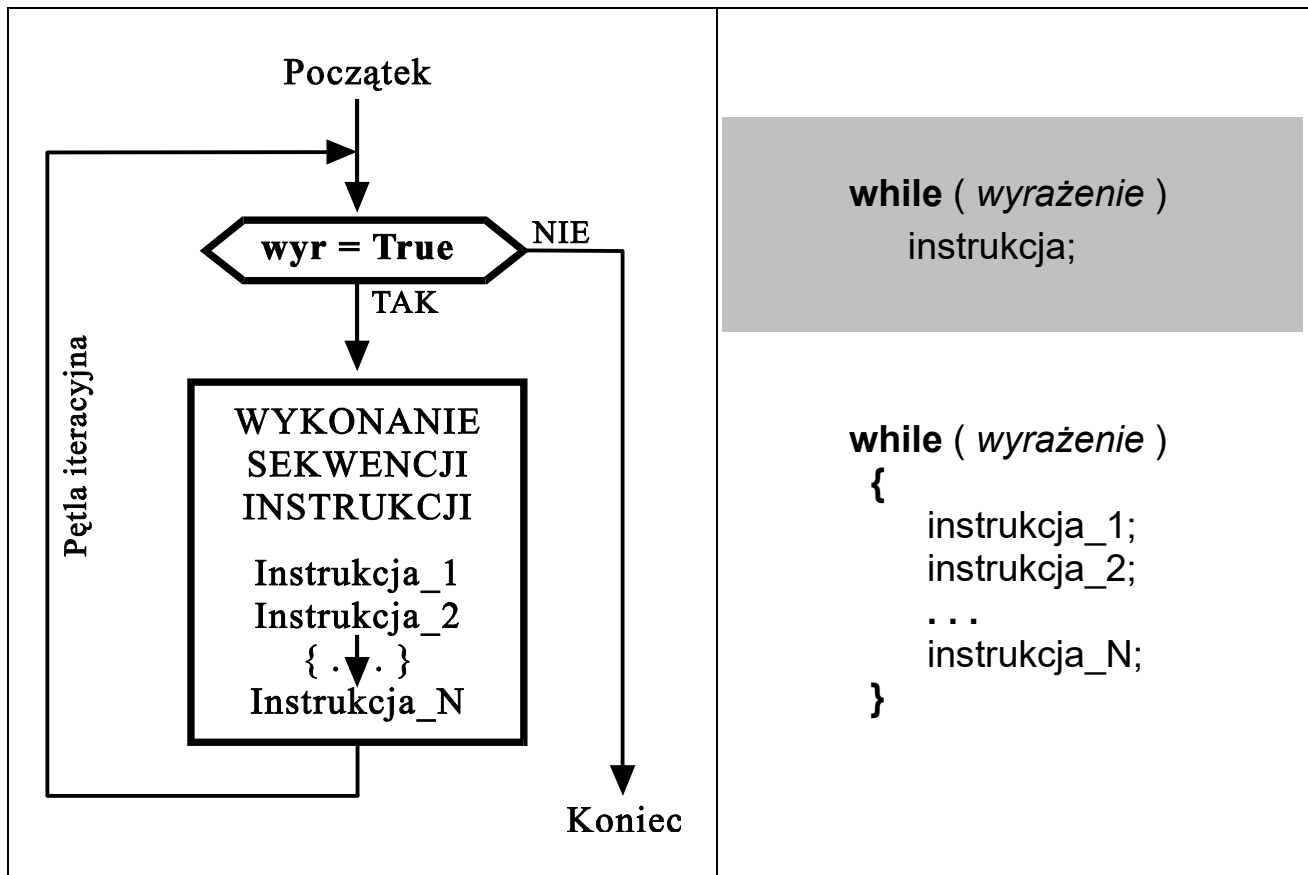


INSTRUKCJE REPETYCYJNE – PĘTLE

- Pętla **while()**

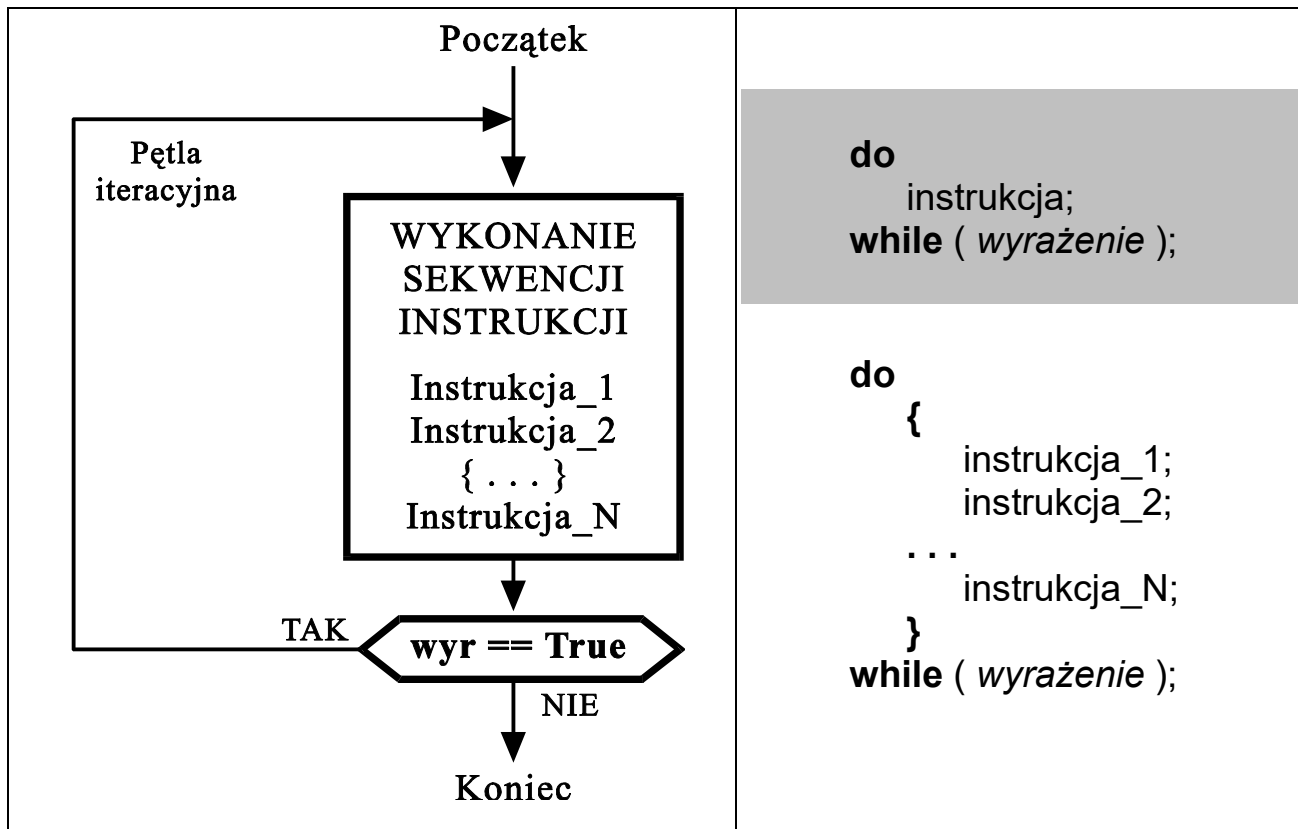


Pętla wykonywana jest tak długo jak wartość **wyrażenie** jest różna od zera

Przykłady:

<pre>int i ; // pętla wyświetlająca liczby 1,2,3 ... i = 1; while(i <=10) { printf ("%2d\n" , i); i = i + 1; }</pre>	<pre>int i = 1; // 1, 2, 3, ... w innym zapisie while(i <11) printf ("%2d\n" , i++);</pre>
<pre>int i ; // pętla wyświetlająca liczby 10,9,8 i = 10; while(i != 0) { printf ("%2d\n" , i); i = i - 1; }</pre>	<pre>int i = 10; //10, 9, 8, ... w innym zapisie while(i) printf ("%2d\n" , i--);</pre>

- Pętla **do while()**



Pętla wykonywana jest tak długo jak wartość **wyrażenie** jest różna od zera

Przykłady:

<pre> int i ; // pętla wyświetlająca liczby 1,2,3 ... i = 1; do { printf ("%2d\n" , i); i = i + 1; } while(i<=10); </pre>	<pre> int i = 1; // 1, 2, 3, ... w innym zapisie do printf ("%2d\n" , i); while(++i <11); </pre>
<pre> int i ; // pętla wyświetlająca liczby 10,9,8 i = 10; do { printf ("%2d\n" , i); i = i - 1; } while(i != 0); </pre>	<pre> int i = 10; // 10, 9, 8, ... w innym zapisie do printf ("%2d\n" , i); while(--i); </pre>

przykład 1: Odczytywanie klawiszy do momentu naciśnięcia 'k' – za pomocą pętli „while”

```
#include <stdio.h> // Język C
#include <conio.h>
// biblioteka <conio.h> zawierająca funkcję getch()
int main( void )
{
    char znak = 'a';
    while( znak != 'k' ) {
        printf( "\n naciśnij jakiś klawisz: " );
        znak = getch( );
    }
}
```

przykład 2: Odczytywanie klawiszy do momentu naciśnięcia ESC – za pomocą pętli „do while”

```
#include <stdio.h> // Język C
#include <conio.h>
int main( void )
{
    char znak;
    do {
        printf( " \n naciśnij jakiś klawisz: " );
        znak = getche( );
    } while(znak != 27 ); // getch w wersji z przyrostkiem 'e' (echo)
                        // 27 = kod klawisza 'Esc' i.e. Escape
}
```

przykład 3: Wyświetlenie kreski składającej się z 10 znaków „minus”

```
#include <stdio.h> // Język C
int main( )
{
    int licznik=0; while( licznik<10 ) { printf( "-" ); ++licznik; }
}
```

przykład 4: Losowanie liczb z przedziału 1÷6, do momentu trafienia na „szóstkę”

```
#include <stdio.h> // Język C
#include <stdlib.h> // dołączenie biblioteki zawierającej funkcje „rand” i „srand”
#include <time.h> // dołączenie biblioteki zawierającej funkcję „time”
int main( )
{
    int liczba_losowa;
    srand( time(0) );
    do {
        liczba_losowa = rand()%6 + 1; // symulacja rzutu kostką
        printf("\n Wylosowano: %d ", liczba_losowa );
    } while( liczba_losowa != 6 );
    printf( "\n\n Koniec programu. Nacisnij ENTER" ); getchar();
}
```

- Pętla for(;;)

```
for( wyrażenie_inicjujace ; wyrażenie_testujace ; wyrażenie_modyfikujace )
    wykonywana_instrukcja ;
```

jest równoważna konstrukcji:

```
wyrażenie_inicjujace ;
while( wyrażenie_testujace )
{
    wykonywana_instrukcja ;
    wyrażenie_modyfikujace ;
}
```

<pre>int i ; i = 10; while(i != 0) { printf ("%2d\n" , i); i = i - 1; }</pre>	<pre>int i ; for(i = 10; i != 0 ; i = i - 1) printf("%2d\n" , i); lub int i ; for(i = 10; i ; printf("%2d\n" , --i));</pre>
---	--

przykład 5:

Wyświetlenie paska w postaci 80 znaków '#'

```
#include <stdio.h> // Język C
int main( )
{
    for( int i=0 ; i<80 ; ++i ) printf( "# " ); // czy lepiej ++i czy i++ ?
}
```

przykład 6:

Program wypisujący tabelę wybranych kodów ASCII

```
#include <stdio.h> // Język C
int main( )
{
    for( int znak=32; znak<256; ++znak )
        printf( "%4d = %c" , znak , znak );
}
```

przykład 7:

Prymitywny kalkulator sumujący liczby wpisywane na klawiaturze

```
#include <stdio.h> // Język C
int main( )
{
    double suma=0, liczba;
    while( scanf( "%lf" , &liczba ) )
        printf( "\t%.2f\n" , suma+=liczba );
}
```

przykład 8:

Policzenie sumy N liczb wpisywanych z konsoli

```
#include <stdio.h> // Język C
int main( )
{
    int i, N;
    float liczba, suma;
    printf( "Podaj ile liczb chcesz zsumowac N = " );
    scanf( "%d", &N );
    suma=0;
    for( i=1; i<=N; ++i )
    {
        printf( "Podaj %d liczbę: ", i );
        scanf( "%f", &liczba );
        suma = suma+liczba;
    }
    printf( "\n\n Suma %d podanych liczb wynosi: %.3f \n", N , suma );
    fflush( stdin ); // usunięcie znaków z bufora klawiatury
    getchar( );
    return 0 ;
}
```

przykład 9: Wyrysowanie na ekranie prostokątnej ramki o zadanych współrzędnych

```
#include <stdio.h> // Język C
#include <windows.h> // biblioteka dostępna tylko w systemie MS Windows
#include <conio.h> // niestandardowa biblioteka conio (console input/output) DOS/Windows
void gotoxy(int x, int y) // przykładowa implementacja operacji przestawienia kursora w konsoli
{
    HANDLE hConsole = GetStdHandle(STD_OUTPUT_HANDLE);
    struct COORD pos; pos.X = x; pos.Y = y;
    SetConsoleCursorPosition(hConsole, pos);
}
int main()
{
    int x1, y1, x2, y2, x, y;
    printf("Podaj współrzędne 1 naroznika \n\r X1 = "); scanf("%d", &x1);
    printf("\n Y1 = "); scanf("%d", &y1);
    printf("\nPodaj współrzędne 2 naroznika \n\r X2 = "); scanf("%d", &x2);
    printf("\n Y2 = "); scanf("%d", &y2);
    gotoxy(x1, y1)
    for ( x = x1; x <= x2; ++x ) // rysowanie górnej krawędzi
        printf("-");
    gotoxy(x1, y2)
    for ( x = x1; x <= x2; ++x ) // rysowanie dolnej krawędzi
        printf("-");
    for ( y = y1 + 1; y < y2; ++y )
    {
        gotoxy(x1, y); printf("|"); // rysowanie lewej krawędzi
        gotoxy(x2, y); printf("|"); // rysowanie prawej krawędzi
    }
    while ( kbhit() ) getch(); // funkcja kbhit() sprawdza czy jest coś w buforze klawiatury
    getch(); // funkcja getch () odczytuje i usuwa jeden znak z bufora klawiatury
    return 0;
}
```

przykład 10:

Wyświetlenie wszystkich liczb 1÷1000 podzielnych przez 13

```
#include <stdio.h> // Język C
int main( )
{
    for( int liczba=13; liczba<=1000; liczba+=13 )
        printf( "%3d \n " , liczba);
}
```

przykład 11:

Wyświetlenie wszystkich par liczb $x,y \in [1,100]$ spełniających

równanie: $x^2 + y^2 < 500$

```
#include <stdio.h>
int main( )
{
    for( int x=1; x<100; x++ )
        for( int y=1; y<100; y++ )
            if( x*x + y*y < 500 )
                printf( "\n x=%d y=%d" , x, y);
}
```

przykład 12:

Program klasyfikujący naciskane klawisze

```
#include <stdio.h> // Język C
#include <conio.h>
#define ESC 27 //definicja kodu klawisza «Escape»

int main(void)
{
    int klawisz=0;
    clrscr();
    while( klawisz != ESC )
    {
        printf( "\n\nNaciśnij jakiś klawisz (ESC->Koniec): " );
        klawisz = getch();
        if( 'a'<=klawisz && klawisz<='z' )
            printf( "=> To jest mala litera." );
        else if( 'A'<=klawisz && klawisz<='Z' )
            printf( "=> To jest duza litera." );
        else if( '0'<=klawisz && klawisz<='9' )
            printf( "=> To jest cyfra." );
        else if( klawisz == 13 )
            printf( "=> To jest klawisz ENTER." );
        else if( klawisz == ' ' )
            printf( "=> To jest klawisz spacji");
        else
            printf( "=> To jest inny klawisz.");
    }
}
```

```

#include <stdio.h>           // Przykład 13 : program rozpoznający klawisze funkcyjne
#include <conio.h>
#include "def_klawiszy.h"   // dołączenie pliku zawierającego definicje klawiszy
int main( void )
{
    int klawisz;
    do
    {
        printf( "\n\n Nacisnij jakis klawisz: " );
        klawisz = getch( );
        switch( klawisz )
        {
            case ENTER : printf( "To jest ENTER" ); break;
            case ESC   : printf( "To jest ESCAPE" ); break;
            case PREFIX : // jeżeli pierwszy znak jest prefiksem (224 lub 0 ?)
                klawisz = getch( );
                switch( klawisz )
                {
                    case DELETE : printf( "Delete" ); break;
                    case UP_ARROW : printf( "Up arrow" ); break;
                    case DOWN_ARROW : printf( "Down arrow" ); break;
                }
                break;
            case BACKSPACE : printf( "To jest BACKSPACE" ); break;
            default : printf( "Inny - nieznaný pojedynczy klawisz" ); break;
        }
    }
    while( klawisz != ESC );
}

```

// dodatkowy plik dyskowy «def_klawiszy.h» zawierający definicje kodów wybranych klawiszy

```

#ifndef DEF_KLAWISZY
#define DEF_KLAWISZY
#define PREFIX      224
// klawisze "zwykłe" - kodowane za pomocą jednego znaku
#define ESC        27
#define ENTER      13
#define BACKSPACE  8
// klawisze "funkcyjne" - kodowane za pomocą dwóch znaków
#define DELETE     83 // 224, 83
#define UP_ARROW   72 // 224, 72
#define DOWN_ARROW 80 // 224, 80
#define LEFT_ARROW 75 // 224, 75
#define RIGHT_ARROW 77 // 224, 77
#define HOME       71 // 224, 71
#define END        79 // 224, 79
#endif

```