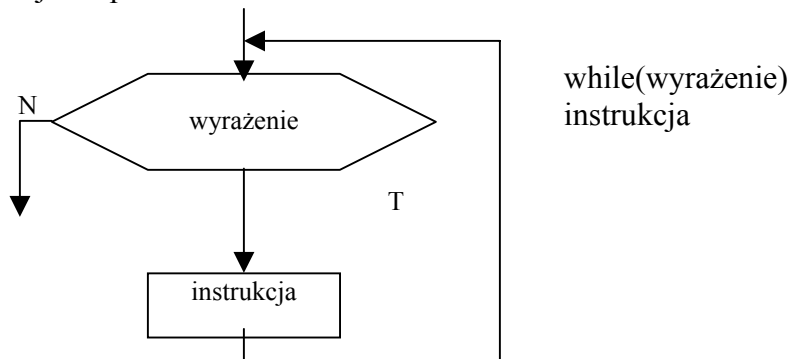


Temat zajęć:                   **Operatory i instrukcje w języku C - 2**

Autor: mgr inż. Sławomir Samolej

**Zagadnienie 1. (instrukcja cyklu: while)**

Do wykonywania cyklicznych obliczeń w języku C stosuje się instrukcje cyklu (pętli). Podstawową instrukcją cyklu w języku C jest instrukcja „while”. Schemat blokowy i składnia instrukcji ma postać:



Jeśli wartość wyrażenia jest uznawane za prawdziwe, to wykonana zostanie instrukcja i program ponownie przejdzie do sprawdzenia wyrażenia. Instrukcja będzie powtarzana dotąd, gdy wyrażenie przestanie być prawdziwe z punktu widzenia języka C.

Uwagi:

- Wyrażenie jest prawdziwe z punktu widzenia języka C, jeśli wartość zwracana przez wyrażenie jest różna od 0. Jeśli wyrażenie zwraca wartość równą 0, to jest ono nieprawdziwe.
- Należy pamiętać, że po instrukcji „while(wyrażenie)” może znajdować się tylko jedna instrukcja! Przy czym instrukcja może być traktowana jako pojedyncza instrukcja prosta, np.:

```
c=3+b;
```

lub jako jedna instrukcja złożona, czyli ciąg instrukcji zamkniętych w nawiasie

klamrowym {...}, np.:

```
{  
    c=3+b;  
    if (c<10)    b=b+1;  
    else        b=b+2;  
}
```

## Przykłady instrukcji „while”

1.

```
#include <stdio.h>
void main(void)
{
    char a=10;
    while(a>0)
        a=a-1;
    printf("a=%d\t",a);
}
```

2.

```
#include <stdio.h>
void main(void)
{
    char a=10;
    while(a>0)
    {
        a=a-1;
        printf("a=%d\t",a);
    }
}
```

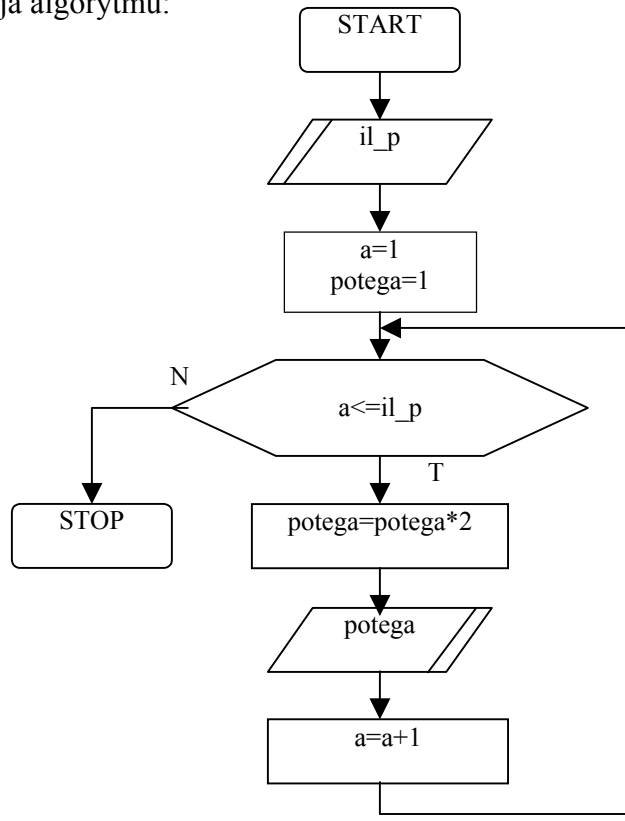
3.

```
#include <stdio.h>
void main(void)
{
    char a=10;
    while(a>0)
    {
        a=a-1;
        if(a%2==1)
            printf("a=%d\t",a);
        else
            printf("a=%d\t",2*a);
    }
}
```

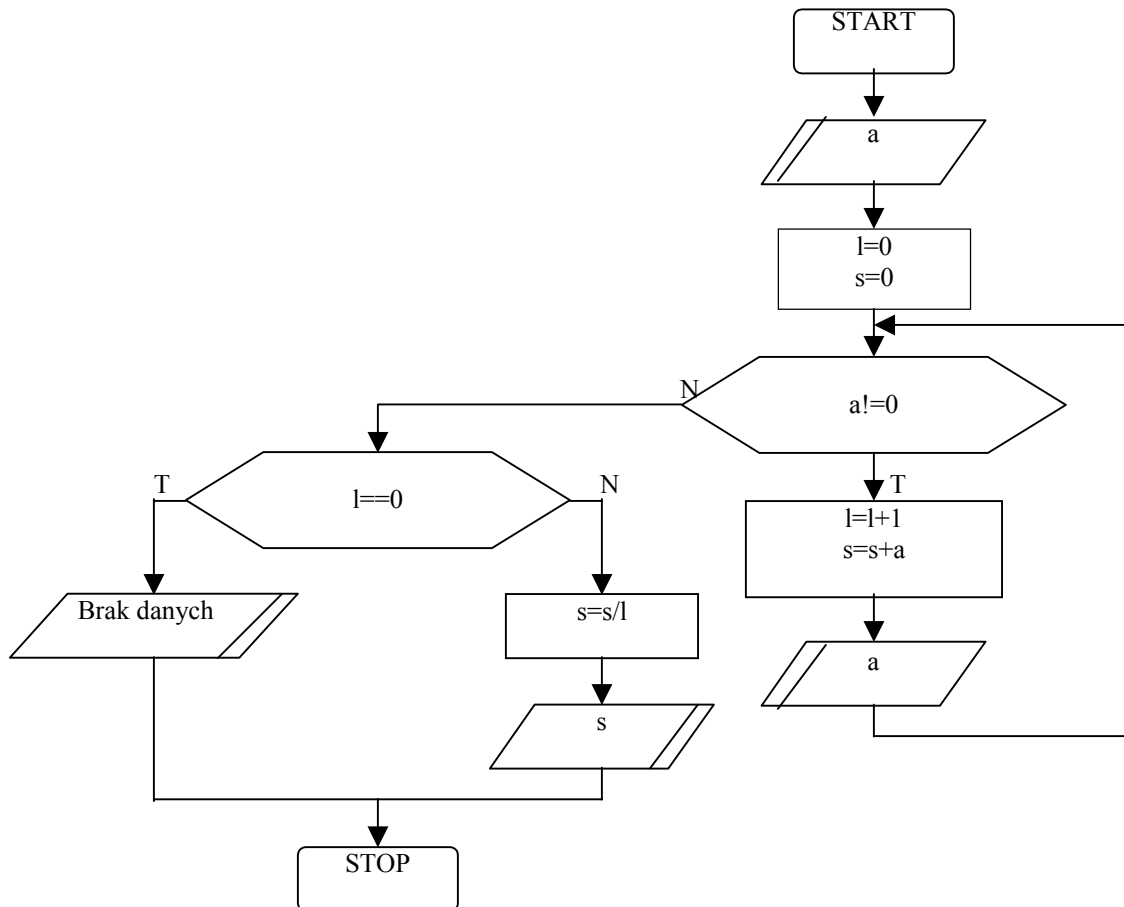
### Zadania:

- Uruchomić przykładowe programy i przeanalizować wyniki ich działania.
- Napisać program, w którym wypisywać będzie zadaną ilość kolejnych potęg liczby 2. Użytkownik będzie podawał ile kolejnych potęg ma być podane, a program ma je po kolei wypisać.

Propozycja algorytmu:



- Napisać program, który pobiera od użytkownika kolejne liczby zmiennopozycyjne i oblicza z nich średnią arytmetyczną. Pobieranie liczb powinno się zakończyć w chwili podania przez użytkownika liczby 0.  
Propozycja algorytmu:



- Podany poniżej program umożliwia powielenie pojedynczej linii znaków wprowadzonych przez użytkownika z konsoli. Program czyta pojedynczy znak z wejścia, sprawdza, czy nie jest on znakiem końca linii, a następnie wypisuje go na konsoli. Odczytywanie znaków trwa do momentu wykrycia znaku końca wiersza ( $\backslash n$ ):  

```

#include <stdio.h>
void main(void)
{
    int c;
    while((c=getchar())!='\n')
        {putchar(c);}
}

```

Program należy uzupełnić o fragment kodu sprawdzający, czy dany znak we wprowadzonym tekście jest literą. Jeśli tak, to zamiast znaku ma zostać wypisana '\*'. Gdy znak nie jest literą, to program ma wypisać znak '-'.

Przykładowo tekst:

Ala ma kota.

Powinien zostać zdekodowany na:

\*\*\*\_\*\*\_\*\*\*\*\_

## Zagadnienie 2. (operatory bitowe)

Do wykonywania operacji na poszczególnych bitach zmiennych w języku C wprowadzono operatory bitowe. Zestawienie operatorów oraz przykładowe wyniki ich działania zamieszczono poniżej:

Oprator	Operacja
&	Bitowe i (and)
	Bitowe lub (or)
^	Bitowe xor
~	Bitowe zaprzeczenie
>>	Przesunięcie bitowe w prawo
<<	Przesunięcie bitowe w lewo

	Wynik (bin)	Wynik (hex)
<b>A</b>	<b>01101011</b>	<b>0x6B</b>
<b>B</b>	<b>00110011</b>	<b>0x33</b>
A & B	00100011	0x23
A   B	01111011	0x7B
A ^ B	01011000	0x58
~ A	10010100	0x94
A << 2	10101100	0xAC
A >> 2	XX011010	(XX==00) 0x1A

### Zadania:

- ❑ Napisać program, który sprawdza poprawność tabeli z przykładowymi wynikami działania operatorów bitowych.
- ❑ Przykładowy program:

```
#include <stdio.h>
void main(void)
{
    unsigned char c;
    c=0x34;
    printf("%x\n",c);
    c=c | 0x03;
    printf("%x\n",c);
}
```

powoduje ustawienie 2 ostatnich bitów zmiennej c na 1. Należy napisać program, który będzie zerował 3 i 1 bit od prawej danej liczby. Np.: liczba 0x37 zostanie przekształcona na 0x32.
- ❑ Przykładowy program:

```

#include <stdio.h>
void main(void)
{
    unsigned char c,licznik=8;
    c=0x34;
    printf("Szesnastkowo: %#8x\n",c);
    printf("Binarnie   : ");
    while(licznik>0)
    {
        if(c&0x80) putchar('1');
        else      putchar('0');
        c=c<<1;           // krócej: c<<=1;
        licznik=licznik-1; // krócej: licznik--;
    }
}

```

dokonuje analizy bit po bicie wskazanej zmiennej 8 bitowej oraz na podstawie analizy wyświetla jej reprezentację binarną.

Napisać program, który analizuje zmienną 8 bitową i wylicza ilość zerowych bitów w zmiennej.

- Napisać program, który „pakuje” dwie liczby unsigned char (8 bitów) do jednej liczby unsigned short (16 bitów). Np.: liczby 0x22 i 0x45 po spakowaniu powinny stworzyć jedną liczbę: 0x2245.