

# Wstęp do programowania

Wykład 3. Pętle.

# Plan wykładu

## 1. Pętle

- **do...while**

- **while**

- **for**

## 2. Przykłady zastosowań.

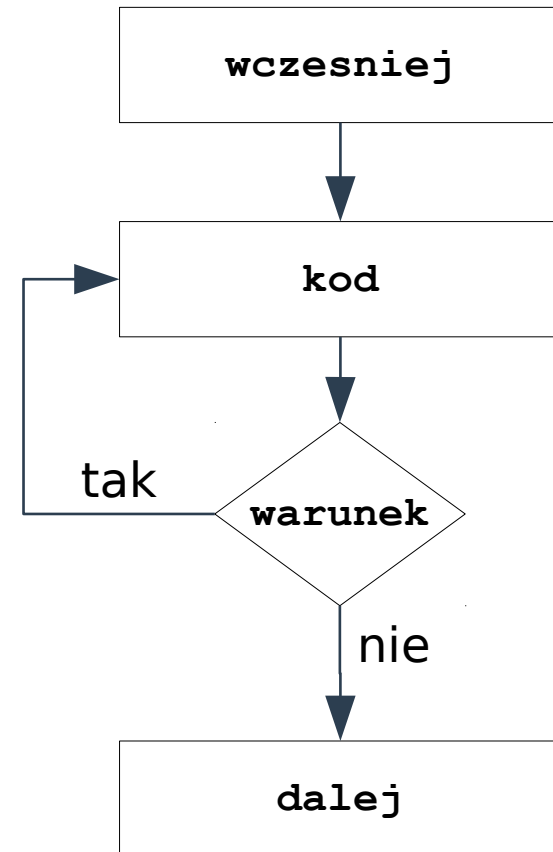
## 3. Rekurencja.

# do ... while

```
wczesniej  
do{  
    kod  
}while (warunek) ;  
dalej
```

Przykład:

```
int i;  
i=0;  
do{  
    printf("%d", i);  
    i++;  
}while (i<10);  
printf("koniec\n");
```

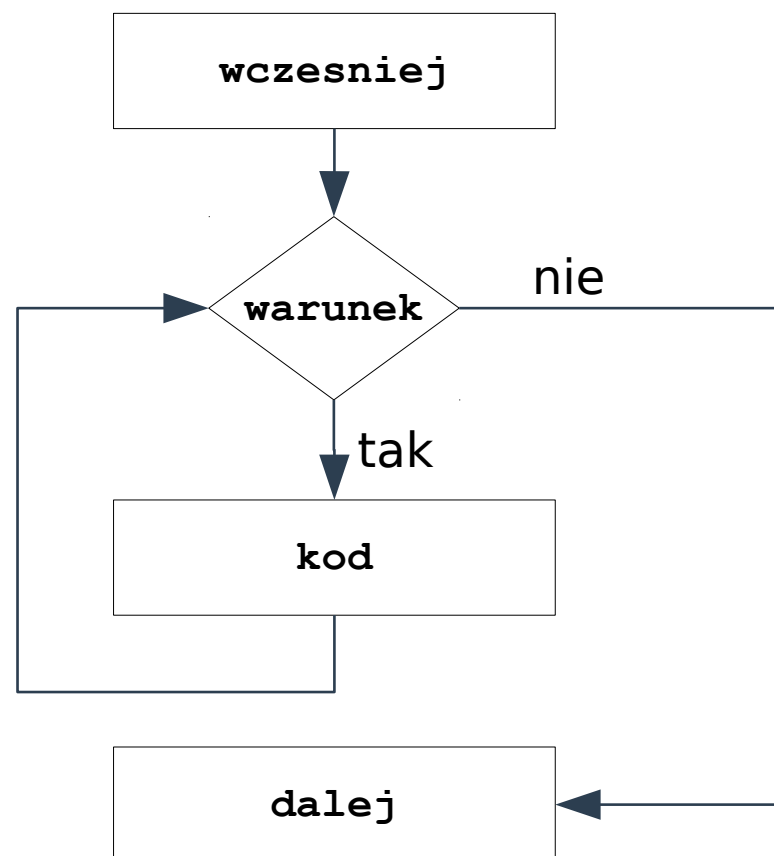


# while

```
wczesniej  
while (warunek) {  
    kod  
}  
dalej
```

Przykład:

```
int i;  
i=0;  
while (i<10) {  
    printf("%d\n", i);  
    i++;  
}  
printf("koniec\n");
```

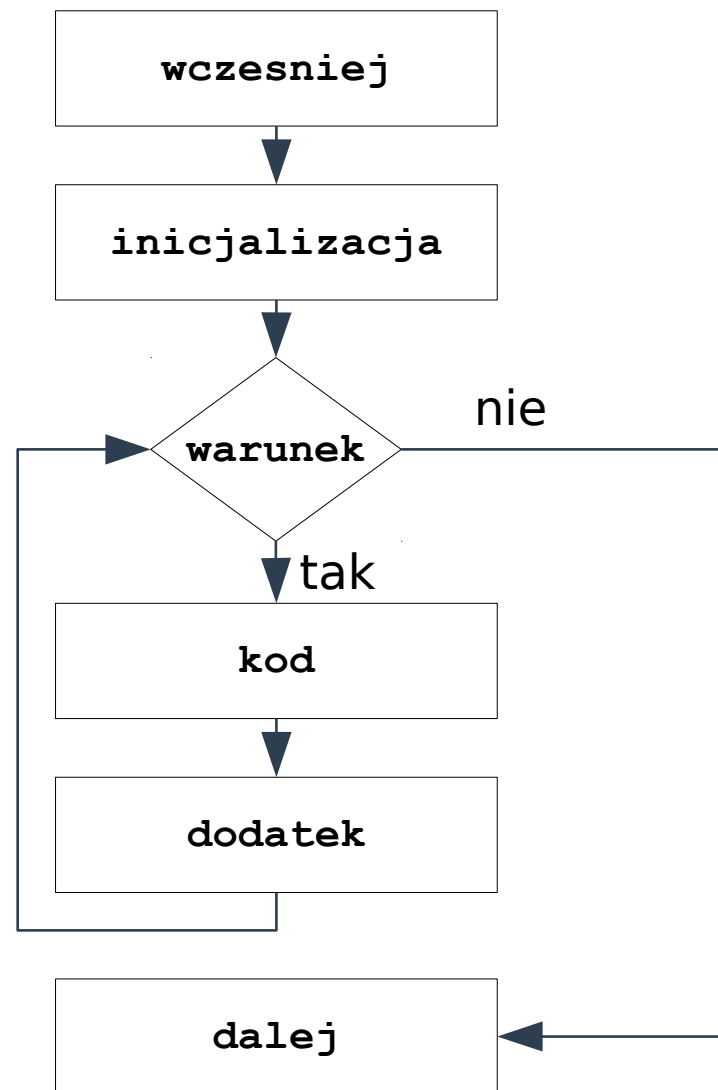


# for

```
wczesniej  
for(inicjalizacja; warunek; dodatek){  
    kod  
}  
dalej
```

Przykład:

```
int i;  
for(i=0; i<10; i++){  
    printf("%d", tablica[i]);  
}  
printf("koniec\n");
```



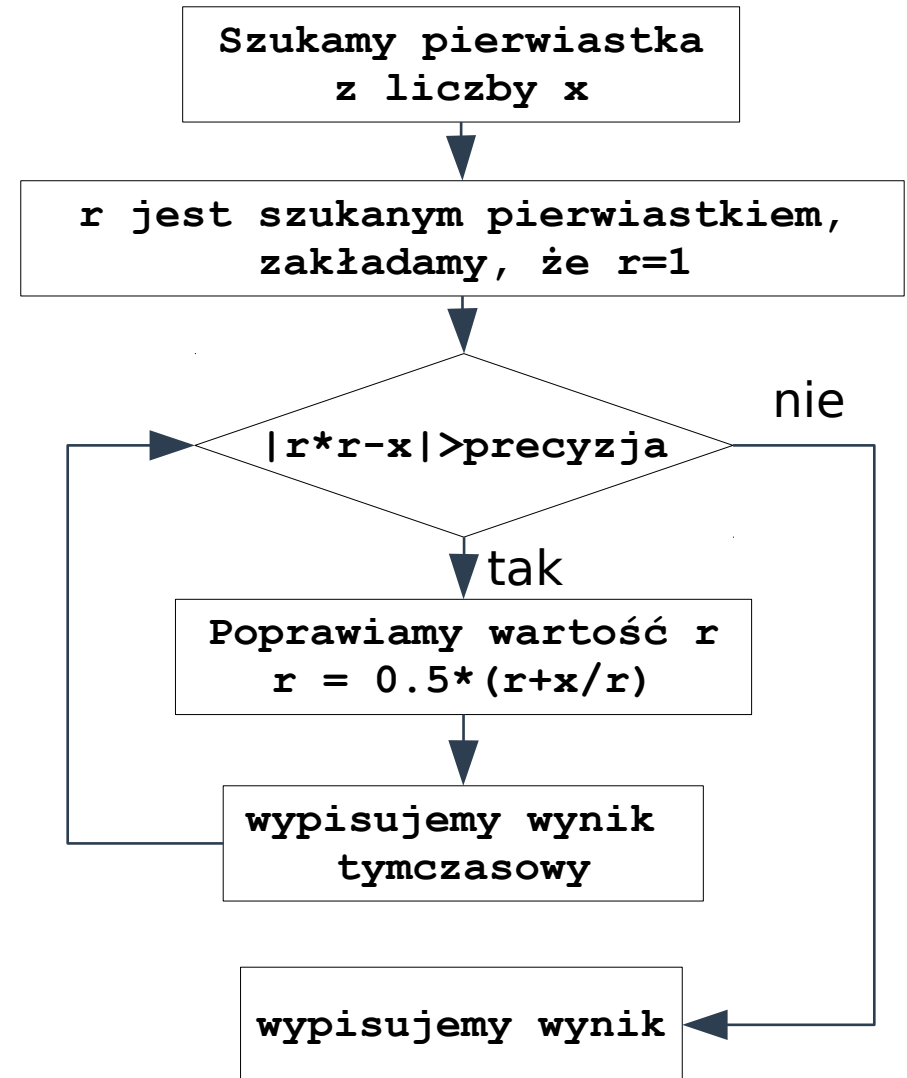
# Przykłady (pierwiastek)

```
#include<stdio.h>
#include<math.h>

#define PRECYZJA 0.00001

double pierwiastek(double x){
    double r = 1;
    while(fabs(r*r-x)>PRECYZJA){
        r = 0.5*(r+x/r);
        printf("przyblizenie %f\n", r);
    }
    return r;
}

void main(void){
    double liczba;
    printf("podaj liczbe\n");
    scanf("%lf",&liczba);
    printf("pierwiastek z %lf = %lf\n",
        liczba, pierwiastek(liczba));
}
```



# Przykłady (argumenty wywołania programu)

```
#include<stdio.h>

void main(int argc, char **argv){
    int i;
    for(i=0; i<argc; i++){
        printf("argument %d:%s\n", i, argv[i]);
    }
}
```

Kompilacja:

```
gcc -o argumenty argumenty.c
```

Uruchomienie

```
./argumenty jeden dwa trzy cztery
```

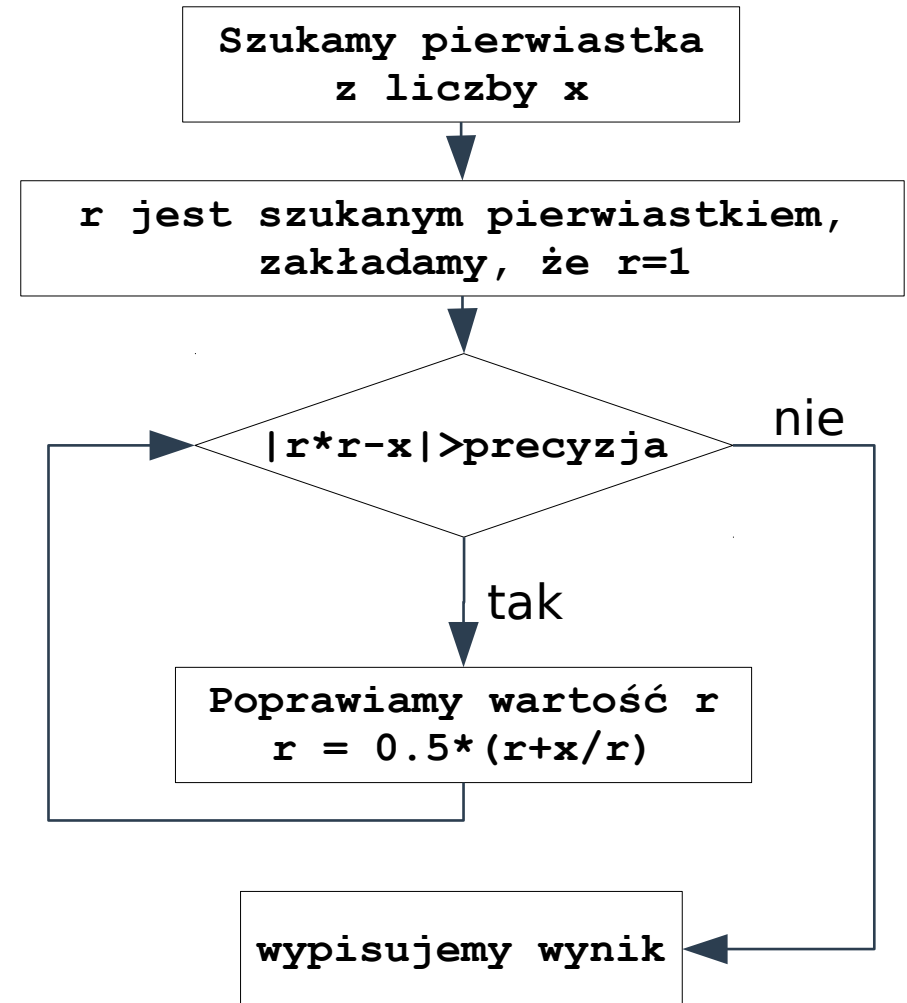
# Przykłady (pierwiastek)

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

#define PRECYZJA 0.00001

double pierwiastek(double x){
    double r = 1;
    while(fabs(r*r-x)>PRECYZJA){
        r = 0.5*(r+x/r);
    }
    return r;
}

void main(int argc, char **argv){
    double liczba;
    if (argc<2)
        printf("brak argumentu\n");
    else{
        liczba = atof(argv[1]);
        printf("pierwiastek z %lf = %lf\n",
            liczba, pierwiastek(liczba));
    }
}
```





# Przykłady (liczba pierwsza)

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

int isPrime(int d){
    int i;
    for(i=2; i<sqrt(d); i++)
        if (d%i==0)
            return i;
    return 0;
}

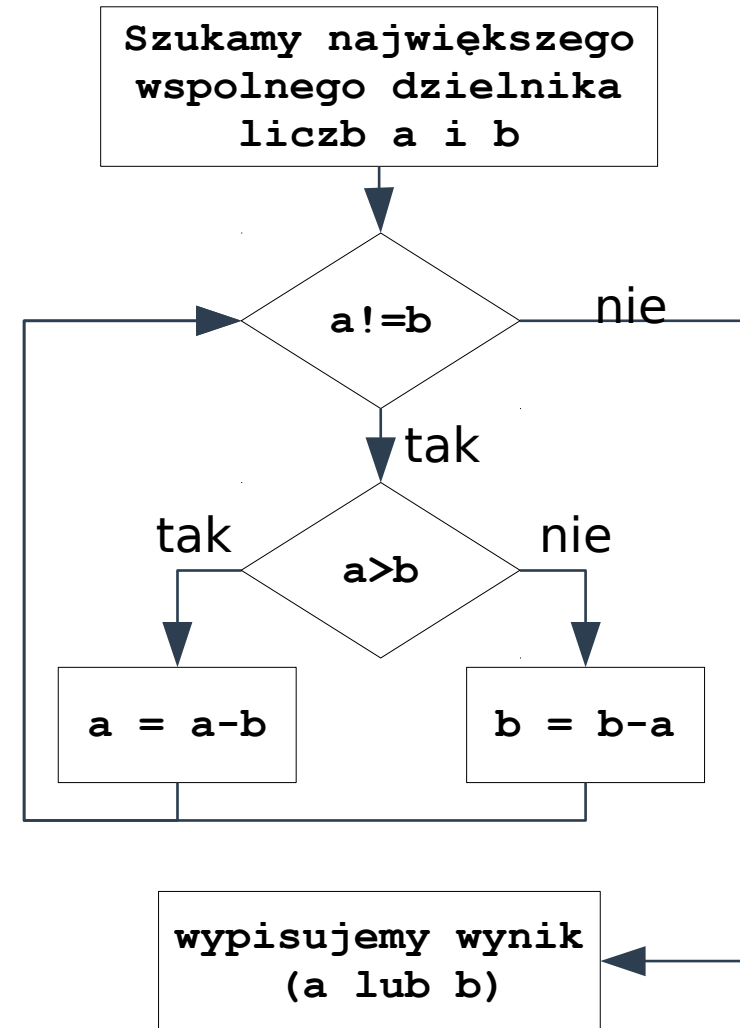
void main(int argc, char **argv){
    if (argc<2){
        printf("brak argumentu\n");
        return;
    }
    int d, liczba = atoi(argv[1]);
    if ((d=isPrime(liczba))==0)
        printf("%d jest liczba pierwsza\n", liczba);
    else
        printf("%d nie jest liczba pierwsza bo %d = %d*%d \n", liczba, liczba,
            d, liczba/d);
}
```

# Przykłady (największy wspólny dzielnik - algorytm Euklidesa)

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

int nwd(int a, int b){
    while(a!=b)
        (a>b)?(a-=b):(b-=a);
    return a;
}

void main(int argc, char **argv){
    int i1, i2;
    if (argc<3){
        printf("brak argumentow\n");
        return;
    }
    i1 = atoi(argv[1]);
    i2 = atoi(argv[2]);
    printf("NWD(%d,%d)=%d\n",
        i1, i2, nwd(i1, i2));
}
```



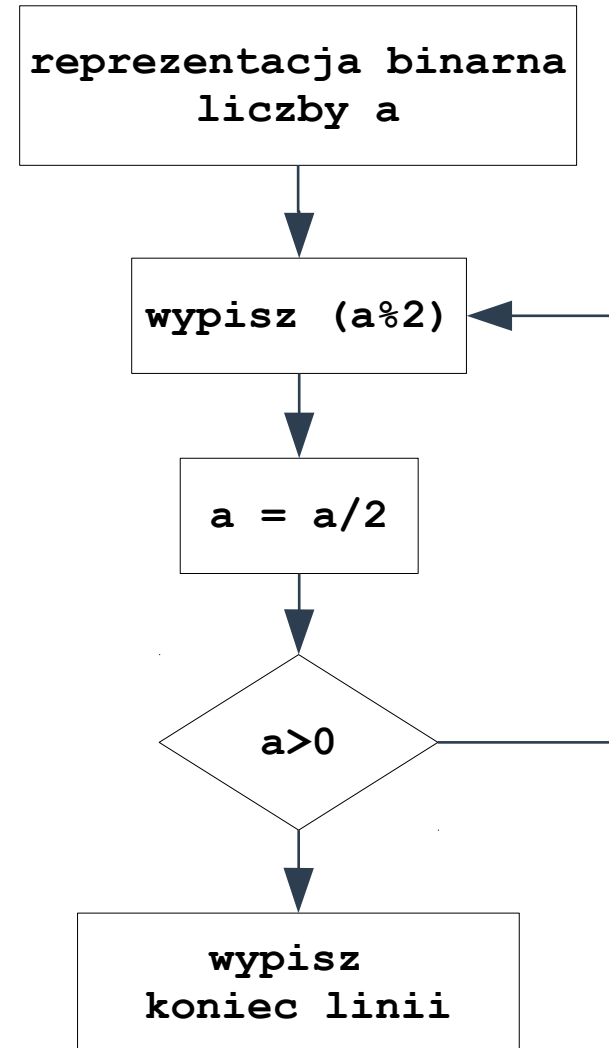
# Przykłady (konwersja dec → bin)

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

void dec2bin(int a){
    do{
        printf("%d", a%2);
        a/=2;
    }while(a>0);
}

void main(int argc, char **argv){
    if (argc<2){
        printf("brak argumentu\n");
        return;
    }
    dec2bin(atoi(argv[1]));
    printf("\n");
}
```

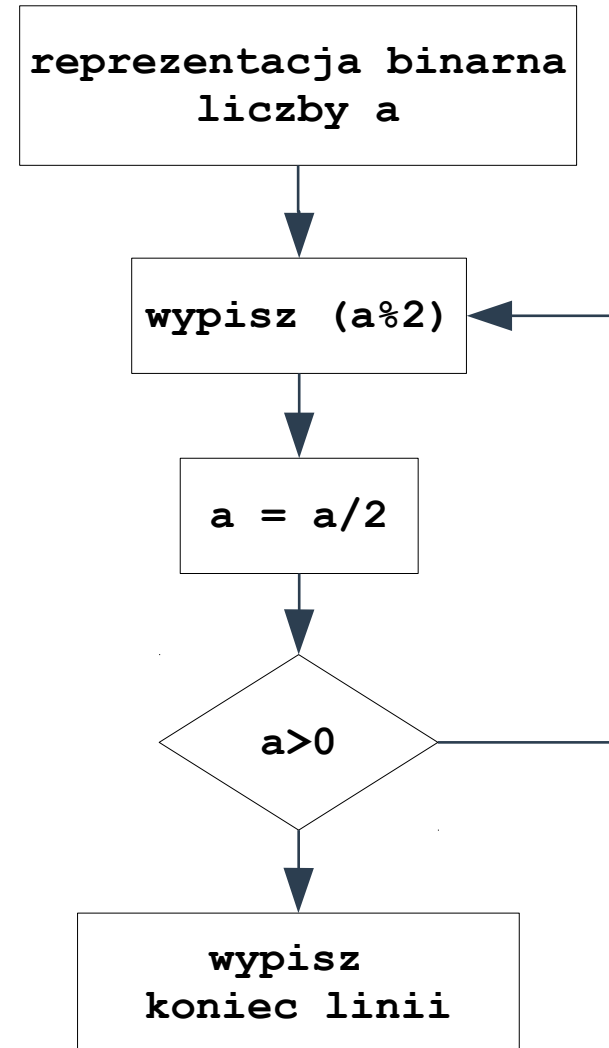
Ale ten program wypisuje liczbę od tyłu!



# Przykłady (konwersja dec → bin)

Rozwiązanie:

Tworzymy tablicę do której wpisujemy wartości (zamiast wypisywać je na ekran), a następnie korzystając z pętli (np. `for`), wypisujemy zawartość tablicy od tyłu (zadanie domowe).



# Przykłady (konwersja dec → bin)

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

void dec2bin(int a){
    if (a>0)
        dec2bin(a/2);
    printf("%d", a%2);
}

void main(int argc, char **argv){
    if (argc<2){
        printf("brak argumentu\n");
        return;
    }
    dec2bin(atoi(argv[1]));
    printf("\n");
}
```

Ten program zawsze wypisuje 0 na początku liczby – proszę spróbować to poprawić.

# Plan wykładu

Dziękuję za uwagę