

Podstawy programowania w języku C i C++.

Temat: *Język C: typy danych, operatory i wyrażenia.*

Instrukcje wejścia, wyjścia. Przykłady programów.

Cele nauczania: Podstawowe zasady programowania.

Praktyczna znajomość elementów programowania w języku C/C++

Docelowo : Opracowanie algorytmu wybranego zadania geodezyjnego i przedstawienie go w postaci schematu blokowego. Opracowanie programu w wybranym języku programowania Wykonanie dokumentacji programu.

Programy do wprowadzenia i uruchomienia.

Listingi programów należy wkleić do Worda – łącznie z wynikami obliczeń. Program.exe > program.txt

```
/* Program wczytuj1.c - funkcja puts(), scanf() , printf() */  
#include <stdio.h>  
#include <conio.h>  
int main() {  
    int a, b, c; char nazw[15];  
    clrscr();  
    puts("Wprowadzanie danych w jezyku C");  
    puts("Podaj wartosci 3 zmiennych calkowitych oddzielone spacja: ");  
    scanf("%d %d %d", &a,&b,&c);  
    printf("Wprowadzone liczby: a= %d  b= %d  c= %d",a,b,c);  
    printf("\nPodaj swoje nazwisko: ");  
    scanf("%s",nazw);  
    printf("\nNazywasz sie %s ",nazw);  
    getch();  
    return 0;  
}
```

Wynik skierować do pliku: wczytuj1.exe > wczytuj.txt

- Trzy kolejne programy z funkcją putchar() – wydruk znaku

- **/* 1) Program litznak.c - literał znakowy */**

```
#include <studio.h> main() { putchar('H'); }
```

- **/* 2) Program stalachar.c */**

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

```
#include <conio.h>
```

```
#define INIT 'A' //stała
```

```
main()
```

```
{
```

```
const char INITIAL = 'H'; // stała
```

```
putchar(INIT); putchar(INITIAL); putchar('\n'); getch();
```

```
}
```

- **/* 3) Program zmchar.c */**

```
#include <studio.h>
```

```
main()
```

```
{
```

```
char litera; // deklaracja zmiennej
```

```
litera='G'; // definicja zmiennej
```

```
putchar(litera); putchar('\n');
```

```
}
```

```
/* Program z funkcją puts() – wydruk napisu - puts1.c */
```

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

```
#include <conio.h>
```

```
#define KOMUNIKAT "Jak sie masz?"
```

```
int main()    {
```

```
char pozdrowienie[] = "Dzien dobry";
```

```
puts("Jak leci?");
```

```
puts(KOMUNIKAT);
```

```
puts(pozdrowienie);
```

```
getch();
```

```
return 0;
```

```
}
```

```
/* Przykład: getchar1.c – funkcja getchar() i printf() */
```

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

```
#include <conio.h>
```

```
int main(void)
```

```
{
```

```
int inicjal;
```

```
puts("Podaj swój inicjal ");
```

```
inicjal=getchar();
```

```
putchar(inicjal); putchar('\n');
```

```
printf("Twój inicjal to %c. Jego kod ASCII: %i ",inicjal, inicjal);
```

```
getch();
```

```
}
```

/* Program liczby1.c - wydruk w różnych formatach: 10-m, 8-m, 16-m */

```
/* liczby1. c */  
#include <stdio.h>  
#include <conio.h>  
main()  
{  
    int k = 21101;    /* liczba dziesiętna */  
    printf(" %s %d %s", "Zapis liczby ", k, " w różnych formatach \n ");  
    printf("\nk(_10) = %i  k(_8) = %o  k(_16) = %X ", k, k, k );  
    getch();  
}
```

```
/* Program printf2.c – funkcja printf() */
```

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

```
void main(void)
```

```
{
```

```
float    f = 0.521;
```

```
int      i = -123;
```

```
unsigned int u = 24;
```

```
char     c = 'A';
```

```
printf("Zmienna f = %f, a zmienna i jest rowna %d.\n", f, i);
```

```
printf("Zmienna c = %c, a zmienna u jest rowna %u.\n", c, u);
```

```
printf("Zmienna u w zapisie szesnastkowym jest rowna %x, \n", u);
```

```
printf("natomiast w zapisie osemkowym jest rowna %o.", u);
```

```
}
```

```
/* program printf3.c – funkcja printf() */  
#include <stdio.h>  
void main(void)  
{  
    float f = 0.521;  
    printf("Zmienna f = %6.3f \n", f); // dosunięcie do prawej strony – razem 6 znaków  
    printf("Zmienna f = %-6.3f \n", f); // dosunięcie do lewej strony – razem 6 znaków  
    printf("Zmienna f = %06.3f \n", f); // 0 z przodu  
    printf("Zmienna f = %+6.3f \n", f); // znak + z przodu  
    getch();  
}
```

Wynik

Zmienna f = **0.521**

Zmienna f = **0.521**

Zmienna f = **00.521**

Zmienna f = **+0.521**


```
/* Program printf4.c - printf() */
#include <stdio.h>
#define PI      3.1415926
#define PROMIEN 3.3
#define WYSOKOSC 44.4
main()
{
    double promien, wysokosc, objetosc;
    promien = PROMIEN;
    wysokosc = WYSOKOSC;
    objetosc = PI * promien * promien * wysokosc;
    printf("\nObjetosc walca = %f", objetosc);
    printf("\nObjetosc walca = %E", objetosc);
    printf("\nObjetosc walca = %g", objetosc);
    printf("\nObjetosc walca = %15.10f", objetosc);
    printf("\nObjetosc walca = %25.20f", objetosc);
}
```

Wyniki:

Objetosc walca = **1519.010288**

Objetosc walca = **1.519010E+03**

Objetosc walca = **1519.01**

Objetosc walca = **1519.0102875816**

Objetosc walca = **1519.01028758159986900000**

/* Przykład funkcji z gets() – wprowadzenie imienia */

/* Program gets1.c */

#include <stdio.h>

int main(void)

{

char imie[21];

// deklaracja tablicy znaków – tekst na 20 zn., ostatni 21 to znak końca \0

*printf("Podaj swoje imie "); **gets(imie);** // wprowadzenie imienia*

printf("Twoje imie to %s " , imie); // wyświetlenie imienia

return 0;

} // Uwaga! Po uruchomieniu Alt F5 – ekran wyników.

Wersja druga programu wczytywania danych z **gets**

```
/* Program wczytuj2.c */
```

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

```
#include <conio.h>
```

```
int main() {
```

```
    int a, b, c; char nazw[15];
```

```
    clrscr();
```

```
    puts("Wprowadzanie danych w jezyku C");
```

```
    puts("Podaj swoje nazwisko ");
```

```
    gets(nazw);
```

```
    puts("Podaj wartosci 3 zmiennych calkowitych oddzielone spacja: ");
```

```
    scanf("%d %d %d", &a,&b,&c);
```

```
    printf("Wprowadzone liczby: a= %d  b= %d  c= %d",a,b,c);
```

```
    printf("\nNazywasz sie %s ",nazw);
```

```
    getch();
```

```
    return 0;
```

```
}
```

Program kodklaw.c - podaje kod wciśniętego klawisza:

```
/* Program kodklaw.c */
#include <stdio.h>
#include <conio.h>
#include <string.h>
main (void) {
char znak; int kod;
clrscr();
printf("Wciśnij znak na klawiaturze: \n");
scanf("%c",&znak);
printf("Kod wciśniętego znaku to: %d \n",znak);
printf("Podaj kod znaku: \n");
scanf("%d",&kod);
printf("Znak o podanym kodzie to: %c \n",kod);
getch();
return 0 ;
}
```

/* Program kalk1.c – prosty kalkulator, dane liczby całkowite*/

#include <stdio.h>

#include <conio.h>

int main() */* funkcja główna */*

{

int a,b; */* deklaracja zmiennych całkowitych a i b */*

int suma,roznica,iloczyn; float iloraz; */* deklaracje zmiennych */*

clrscr(); */* kasowanie ekranu */*

printf("Prosty kalkulator\n"); */* nagłówek */*

printf("\nPodaj liczbe a: "); **scanf("%d",&a);** */* wczytanie liczby a */*

printf("Podaj liczbe b: "); **scanf("%d",&b);** */* wczytanie liczby b */*

suma=a+b; roznica=a-b; iloczyn=a*b; */* obliczenia */*

iloraz=(float)a/(float)b; */* operator rzutowania w dzieleniu */*

printf("\nWyniki dzialan:\n"); printf("\nSuma: %d ",suma); printf("\nRoznica: %d ",roznica);

printf("\nIloczyn: %d ",iloczyn); printf("\nIloraz: %f ",iloraz); getch();

return 0 ;

}

Program **kolo.c** na pole i obwód koła

```
/* program kolo.c oblicza obwod
i pole koła o promieniu podanym */
#include <stdio.h>
#include <conio.h>
#include <math.h>
#define PI 3.1415926
#define NL printf("\n");
int main()
{
    /* deklaracja zmiennych */
    const double pi=3.1415926;
    float r , o, p;
    /* promień koła, obwod, pole */
    clrscr;
    puts("Program oblicza obwod i pole kola");
    printf("PI=%f pi=%f M_PI=%f",PI, pi, M_PI);
```

```
printf("\nPodaj promien kola: ");
scanf("%f",&r);      /* wczytaj r z klawiatury */
NL;                 /* nowa linia */
o=2.0*PI*r;         /* obwod */
p=PI*r*r;           /* pole */
printf("Obwod kola o promieniu %.3f = %8.3f",r,o );
NL;
printf("Pole tego kola = %8.3f", p); /* wypisz pole koła */
NL;
getch();           /* czeka na naciśnięcie klawisza */
return 0;
}
```

Obliczenie pola koła. Wersja z wprowadzeniem promienia i formatowaniem wydruku

```
/* program polkola3.c */  
/* dyrektywy załączające tzw. nazwane pliki */  
#include <stdio.h> #include <conio.h>  
main() /* funkcja główna */  
{  
    const float PI=3.141593; /* stała */  
    float r, p;  
    clrscr(); /* kasowanie ekranu - określone w conio.h */  
    puts("Obliczenie pola kola o promieniu r "); /* Napis na ekranie */  
    printf("Podaj r => "); scanf("%f",&r); /* wczytanie promienia r */  
    p=PI*r*r; /* obliczenie pola - wyrażenie */  
    printf("Pole kola o promieniu %.0f = %7.3f\n",r, p); /* wydruk */  
    getch(); /* czeka na naciśnięcie klawisza - w stdio.h */  
}
```

- **Wejście – wyjście - przykładowy program w C++:**

// Program wewycpp.cpp

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
void main() {
```

```
int a, b, c;
```

```
cout << "Podaj wartosc zmiennej a: ";
```

```
cin >> a;
```

```
cout << "\nPodaj wartosc zmiennej b: ";
```

```
cin >> b;
```

```
cout << "a = " << a << " b = " << b << endl;
```

```
getch();
```

```
}
```


/* Program daneos1.c - dane osobowe */

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

```
int main(void)
```

```
/* funkcja glowna, brak argumentow, zwraca typ calkowity int */
```

```
{
```

```
    char imie[20]; /* deklaracja tablicy znakow – łańcuch – tekst na 19 znaków */
```

```
    int i; /* deklaracja zmiennej całkowitej i */
```

```
    printf("\nPodaj swoje imie "); /* wydruk napisu na ekran, \n – nowa linia */
```

```
    gets(imie); /* wprowadzenie imienia */
```

```
    puts("Ile masz lat? "); /* wydruk napisu – funkcja puts() */
```

```
    scanf("%d",&i); /* wprowadzenie lat */
```

```
    printf("\n%s ma %d lat.",imie, i); /* wyświetlenie imienia i lat */
```

```
    return 0; /* funkcja glowna zwraca 0 – pomyślny koniec */
```

```
}
```

Prosty kalkulator – liczby rzeczywiste

```
/* Program kalk2.c */
#include <stdio.h>
#include <conio.h>
int main() /* funkcja glowna */
{
float a,b; /* deklaracja zmiennych
           rzeczywistych a i b */
float suma,roznica,iloczyn;
float iloraz; /* deklaracja zmiennej
              rzeczywistej */
clrscr(); /* kasowanie ekranu */
printf("Prosty kalkulator\n");
printf("\nPodaj liczbe a: ");
scanf("%f",&a); /* wczytanie liczby a */
printf("Podaj liczbe b: ");
scanf("%f",&b); /* wczytanie liczby b */
printf("\nLiczba wprowadzona a = %f ",a);
printf("\nLiczba wprowadzona b = %f ",b);

/* c.d - obliczenia */
suma=a+b;
roznica=a-b;
iloczyn=a*b;
iloraz=a/b;

/* Wydruk wynikow */
printf("\nWyniki dzialan:\n");
printf("\nSuma: %f ",suma);
printf("\nRoznica: %f ",roznica);
printf("\nIloczyn: %f ",iloczyn);
printf("\nIloraz: %f ",iloraz);

getch(); /* czeka na naciśnięcie
          klawisza */
return 0 ;
}
```

Banknoty do wydania – program change.c

```
/* Program change.c - zastosowanie operatora % - reszta z dzielenia całkowitego */  
#include <stdio.h>  
main() {  
int amount, twenties, tens, fives, ones, r20, r10;  
printf("Wprowadz kwote do wydania: ");           /* Wprowadzenie kwoty do wydania */  
scanf("%d", &amount);  
twenties= amount/20;           /* banknoty 20-dolarowe */  
r20=amount % 20;           /* r20 reszta pozostała po 20-ch */  
tens= r20/10;           /* ilość banknotów 10-dolarowych */  
r10=r20 % 10;           /* r10 reprezentuje resztę po 10-ch */  
fives= r10 / 5;           /* banknoty 5-dolarowe */  
ones = r10 % 5;           /* reszta - dolarówki */  
putchar('\n'); printf("By przekazac %d wydaj banknoty: \n", amount);  
printf("%d dwudziestki\n", twenties); printf("%d dziesiątki\n", tens);  
printf("%d piatki \n", fives);  
printf("%d pojedyncze (s)\n", ones);  
}
```

// Program wielop1.cpp – przypisanie wieloznakowe

```
#include <iostream.h>
#include <conio.h>
int main() {
int a = 1, b;
cout << "Operatory przypisania wieloznakowe\n";
cout << "a= " << a << endl;
a += 5; /* to samo, co a = a + 5; czyli 6 ; */
cout << "a+=5 czyli a=a+5 = " << a << endl;
a /= a - 3; /* to samo, co a = a / (a - 3); czyli 6/(6-3)=3
*/
cout << "a/=a-3 czyli a=a/(a-3) = " << a << endl;
a +=1; // 4
cout << "a+=1 czyli a=a+1 = " << a << endl;
a %= 2; /* to samo, co a = a % 2; 4%2 */
cout << "a%2 czyli a=a%2 = " << a << endl << endl;;
a=10;
cout << "Nowe a = " << a << endl;
b= (a++, a+=10); // 21
cout << "b= (a++, a+=10) = " << b; // 21
cout << " bo a=a+10 = 20 i b=a++ = a+1 =21" << endl;
getch();
}
```

Wyniki

Operatory przypisania wieloznakowe

a= 1

a+=5 czyli a=a+5 = 6

a/=a-3 czyli a=a/(a-3) = 2

a+=1 czyli a=a+1 = 3

a%2 czyli a=a%2 = 1

Nowe a = 10

b= (a++, a+=10) = 21 bo a=a+10 = 20

i b=a++ = a+1 =21

Podejmowanie decyzji – operacje logiczne

Przykład:

```
int main()
{
    int a, b, c, d; a=1; b=0;
    c = a&&b; // → 0;
    d = a||b; // → 1
    cout << c << endl << d;
return 0;
}
```

Szablony programu dla ucznia – dane wpisane w #define

```
/* Program Szablon1.c */
/* #include - włączenia tekstowe bibliotek */
#include <stdio.h> /* prawie zawsze np. do printf() */
#include <conio.h> /* np. do getch() */
#include <math.h>

/* #define - stałe makroinstrukcje */
#define PROGRAM "program.c"
#define NAZWISKO "Nowak Jan"
#define ROK 2011
#define SZKOLA "Szttygarka"
#define KLASA "2BG"
#define NL printf("\n");
#define TAB putchar('\t');
#define PI 3.141592653
#define KRESKA puts("-----")
#define KRESKA2 puts("=====");
#define KONIEC puts("\n\nNacisnij cos ")
```

```
/* zmienne globalne - stałe, zmienne */
const double pi=3.14159265;
const char szkola[]="Szttygarka";
/* Przykładowa funkcja - deklaracja */
float suma1(float l1, float l2);

/* ===== Funkcja główna * ===== */
int main()
{
    clrscr();
    printf("Program: %s \n",PROGRAM);
    KRESKA2;
    puts(NAZWISKO);
    puts(KLASA);
    puts(SZKOLA);    KRESKA;
    /* -- dalszy kod programu głównego */
    /* ----- */
    KONIEC;
    getch();    return 0;
}

/* ===== Funkcje - definicje ===== */
float suma1(float l1, float l2) /* Definicja funkcji */
{
    return(l1+l2);
}
```

```

/* Program Szablon2.c */
/* #include - włączenia tekstowe bibliotek */
#include <stdio.h>      /* prawie zawsze np. do
printf() */
#include <conio.h>      /* np. do getch() */
#include <math.h>
/* #define - stałe makroinstrukcje */
#define PROGRAM "program.c"
#define NAZWISKO "Nowak Jan"
#define ROK 2011
#define SZKOLA "Szttygarka"
#define KLASA "2GBA"
#define NL printf("\n");
#define TAB putchar('\t');
#define PI 3.141592653
#define KRESKA puts("-----")
#define KRESKA2 puts(„=====“);
#define KONIEC puts("\n\nNacisnij cos ")
#define KATRAD(g) ((g)/(ROG))
/* zmienne globalne - stałe, zmienne */
const double pi=3.14159265;
const char szkola[]="Szttygarka";
float suma1(float l1, float l2); /*Przykładowa
funkcja - deklaracja */

```

```

/* ===== Funkcja główna * ===== */
int main()
{
float a, b;
clrscr();
printf("Program: %s \n",PROGRAM);
KRESKA2;
puts(NAZWISKO);
puts(KLASA);
puts(SZKOLA);
KRESKA;
puts("Obliczenie sumy 2 liczb");
printf("\nPodaj liczbe a => ");
scanf("%f", &a);
printf("\nPodaj liczbe b => ");
scanf("%f", &b);
printf("Suma %f + %f = %f",a, b,
suma1(a,b));
KONIEC;
getch(); return 0;
}
/* Definicja funkcji */
float suma1(float l1, float l2) {
return(l1+l2); }

```

Przykład struktury programu z zastosowaniem wielu instrukcji podstawowych

```
/* Program strukt.pr.c – Przykł. struktury programu */
/* Naglowek - komentarz wieloliniowy w C i C++ */
//jednolinyowy w C++
/* Program Nazwa.c Autor: Nazwisko Imie. Data:
3.11.2011
Kompilator Turbo C 1.01
Uwagi: Struktura ogolna programu
*/
/* preprocesor: #include - wlaczenia tekstowe bibliotek
i #define - stale makroinstrukcje */
#include <stdio.h> /* prawie zawsze c C, np. do printf() */
#include <conio.h> /* np. do getch() */
#include <math.h>
#define NL printf("\n");
#define PI 3.14159
#define R 10.0
#define KRESKA puts("-----")
)

/* zmienne globalne - stale, zmienne */
const float KURS_USD= 3.40, KURS_EURO= 4.50;

float suma_float(float l1, float l2); /* zapowiedź funkcji */

float pole_kola(double r) /* definicja funkcji */
{
double pole, pole1;
pole=PI*r*r;
puts("\nFunkcja pole_kola()");
printf("\nPole z PI =%f",pole);
pole1=M_PI*r*r;
printf("\nPole z M_PI =%f ",pole1);
NL; NL;
return pole1;
}
```

```
int main()
{
int a, b=3, suma, reszta;
float kw_zl, kw_usd;
clrscr();
puts("Struktura programu w C - przyklad\n");
KRESKA; puts("Naglowek");
puts("Preprocesor - #include, #define");
puts("Zmienne globalne i prototypy funkcji lub definicje funkcji");
puts("Funkcja main()"); puts("Funkcje - definicje"); NL;
KRESKA; puts("Przyklady wynikow"); NL;
puts("Funkcja glowna"); NL;
a=10; kw_zl=10;
kw_usd=kw_zl/KURS_USD;
suma= suma_float(a,b); reszta=a%b;
printf("Suma %d + %d = %d \n",a,b,suma);
printf("Suma 2.50 + 3.40 = %f", (2.50+3.40));
NL;
printf("Reszta z dzielenia %d przez %d = %d ",a,b,reszta);
NL; NL;
printf("Kwota %.2f PLN = %.2f USD ",kw_zl, kw_usd); NL;
printf("Pole kola o promieniu %f = %f ",R, pole_kola(R)); NL;
getch();
return 0;
}

float suma_float(float l1, float l2) /* definicja funkcji */
{
return(l1+l2);
}
```


Szablon programu do obliczeń geodezyjnych

```
/* Schemat programu do obliczeń geodezyjnych */
/* Program Szablgeo.c */
/* preprocesor: #include - włączenia tekstowe bibliotek */
#include <stdio.h> /* prawie zawsze c C, np. do printf() */
#include <conio.h> /* np. do getch() */
#include <math.h>
/* #define - stale makroinstrukcje */
#define PROGRAM "program.c"
#define NAZWISKO "Nowak Jan"
#define ROK 2011
#define SZKOLA "Szttygarka"
#define KLASA "2BG"
#define NL printf("\n");
#define TAB putchar('\t');
#define PI 3.141592653
#define ROG 63.66197724
#define ROS=57.2957795130;
#define KRESKA puts("-----")
#define KRESKA2puts("=====");
#define KONIEC puts("\n\nNacisnij cos ")
#define KATRAD(g) ((g)/(ROG))
#define KATGRAD(r) ((r)/(ROS))
```

```
/* zmienne globalne - stale, zmienne */
const double pi=3.14159265;
const double rograd=63.66197724,
rostop=57.2957795130823; /* Ro[grad]=200/PI,
Ro[stopn]=180/PO
const char szkola[]="Szttygarka";
```

```
/* Przykładowa funkcja - deklaracja */
float suma1(float l1, float l2);
```

```
/* ===== Funkcja glowna * ===== */
```

```
int main()
{
    clrscr();
    printf("Program: %s \n",PROGRAM);
    KRESKA2;
    puts(NAZWISKO);
    puts(KLASA);
    puts(SZKOLA);
    KRESKA;
    printf("63.66198[grad]=%f[rad]",KATRAD(rograd));
    /* -- dalszy kod programu, głównego. -- */
    KONIEC;
    getch();
    return 0;
}
```

```
/* ===== Funkcje - definicje ===== */
```

```
float suma1(float l1, float l2) /* Definicja funkcji */
{
    return(l1+l2);
}
```

/* Program daneuczni.c – dane ucznia */

```
/* Program Daneuczni.c */
```

```
/* preprocesor: #include - włączenia tekstowe bibliotek
```

```
i #define - stałe makroinstrukcje */
```

```
#include <stdio.h> /* prawie zawsze c C, np. do printf() */
```

```
#include <conio.h> /* np. do getch() */
```

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

```
#define PROGRAM "program.c"
```

```
#define NAZWISKO "Nowak Jan"
```

```
#define ROK 2011
```

```
#define KLASA "2BG"
```

```
#define NL printf("\n");
```

```
#define TAB putchar('\t');
```

```
#define PI 3.14159
```

```
#define KRESKA puts("-----")
```

```
#define KRESKA2 puts("=====");
```

```
#define KONIEC puts("\nNacisnij cos ")
```

```
/* zmienne globalne - stałe, zmienne */
```

```
const int klasa = 2;
```

```
int main()
```

```
{
```

```
    char c1;
```

```
    char przedmiot[20], szkola[]="Szytygarka";
```

```
    int rok_urodz;
```

```
    float ocena;
```

```
    clrscr();
```

```
    printf("Program: %s \n",PROGRAM);
```

```
    KRESKA2;
```

```
    puts(NAZWISKO);
```

```
    puts(KLASA);
```

```
    puts(szkola);
```

```
    KRESKA;
```

```
    printf("\nPodaj swój ulubiony przedmiot (jeden wyraz): ");
```

```
    gets(przedmiot); /* wprowadzenie imienia */
```

```
    printf("Wprowadz swój inicjał: ");    c1=getchar();
```

```
    printf("Podaj rok urodzenia: ");
```

```
    scanf("%d",&rok_urodz); /* wprowadzenie roku */
```

```
    printf("Twoja średnia ocena ");
```

```
    scanf("%f",&ocena); /* wprowadzenie roku */
```

```
    KRESKA;
```

```
    printf("Twój inicjał: ");
```

```
    TAB;
```

```
    putchar(c1);
```

```
    NL;
```

```
    printf("\n %s ma %d lat.",NAZWISKO, (ROK-rok_urodz));
```

```
/* wyświetlenie imienia i lat */
```

```
    NL;
```

```
    printf("Średnia ocena z przedmiotu %s = %.1f ",przedmiot, ocena);
```

```
    NL;
```

```
    KRESKA;
```

```
    KONIEC;
```

```
    getch();
```

```
    return 0;
```

```
}
```

```
float suma_float(float l1, float l2)
```

```
{
```

```
    return(l1+l2);
```

```
}
```

Pole koła – wprowadzenie promienia i formatowanie wydruku

```
/* program polkola3.c */
/* dyrektywy załączające tzw. nazwane pliki */
#include <stdio.h>
#include <conio.h>
/* funkcja glowna */
main()
{
    const float PI=3.141593; /* stała */
    float r, p;
    clrscr(); /* kasowanie ekranu - określone w conio.h */
    puts("Obliczenie pola koła o promieniu r "); /* Napis na ekranie */
    printf("Podaj r => ");
    scanf("%f",&r); /* wczytanie r */
    p=PI*r*r; /* obliczenie pola - wyrażenie */
    printf("Pole koła o promieniu %.0f = %7.3f\n",r, p); /* wydruk */
    getch(); /* czeka na naciśnięcie klawisza - w stdio.h */
}
```

Kompilacja programu: BCC32 polkola3.c lub Alt F9 w TC

Pole koła w języku C++. Funkcja

```
// Polkola4.cpp
```

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
float pole_kola(float r)    // funkcja
```

```
{
```

```
    float p;
```

```
    p = 3.1415 * r * r;
```

```
    return p;
```

```
}
```

```
void main(void)
```

```
{
```

```
    float a, p;
```

```
    cout << "Podaj promien: ";    cin >> a;
```

```
    p = pole_kola(a);    cout << "Pole kola wynosi: " << p << endl;    getch();
```

```
}
```

Pole prostokąta - dane w programie, funkcja przed main()

```
/* -----polepros2.c - z funkcja -----*/  
#include <stdio.h>  
#include <conio.h>  
float PoleProstokata(float bok1, float bok2) /* definicja funkcji (/  
{  
    /* w tym miejscu bok1 jest równy 5, natomiast b jest równe 10 */  
    float wynik;  
    wynik = bok1 * bok2;  
    return wynik;  
}  
void main(void)  
{  
    float a, b, p;  
    a = 5; b = 10; /* zmienne */  
    p= PoleProstokata(a, b); /* wywołanie funkcji */  
    printf("Pole prostokata o bokach %f i %f = %f ", a, b, p);  
    getch();  
}
```

Pole prostokąta - dane wprowadzone, funkcja na końcu

```
/* ppros3.c - z funkcja */
#include <stdio.h>
#include <conio.h>
float PoleProstokata(float bok1, float bok2); /* zapowiedz funkcji */
/* funkcja glowna */
int main(void)
{
    float a, b, p;
    puts("Obliczenie pola prostokata o bokach a i b ");
    printf("Podaj a i b oddzielone spacja: ");
    scanf("%f %f",&a, &b);
    /* wywołanie funkcji na obliczenie pola */
    p= PoleProstokata(a, b);
    printf("Pole prostokat o bokach %f i %f = %f ", a, b, p);
    getch();
    return 0;
}
float PoleProstokata(float bok1, float bok2) /* definicja funkcji */
{
    /* w tym miejscu bok1 jest równy a, natomiast bok2 jest równy b */
    float wynik;
    wynik = bok1 * bok2;
    return wynik;
}
```

Pole prostokąta - wprowadzanie danych, funkcja

```
* ppros4.c - z funkcja i wprowadzaniem danych */
```

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

```
#include <conio.h>
```

```
float PoleProstokata(float bok1, float bok2); /* zapowiedz funkcji */
```

```
/* funkcja glowna */
```

```
int main()
```

```
{  
    float a, b, p;  
    puts("Obliczenie pola prostokata o bokach a i b ");  
    printf("Wprowadz a  ");  
    scanf("%f", &a); /* wprowadzenia a */  
    printf("Wprowadz b  ");  
    scanf("%f", &b); /* wprowadzenia b */  
    p= PoleProstokata(a, b); /* wywołanie funkcji z parametrami a i b */  
    printf("Pole prostokat o bokach %f i %f = %10.2f \n", a, b, p);  
    printf("\nNacisnij cos ");  
    getch(); /* czeka na znak */  
    return 0;  
}
```

```
/* Funkcja - definicja */
```

```
float PoleProstokata(float bok1, float bok2)  
{  
    /* w tym miejscu bok1 jest równy a,  
    natomiast bok2 jest równy b */  
    float wynik;  
    wynik = bok1 * bok2;  
    return wynik;  
}
```

Wyświetlenie czasu C++ - program w Dev C++

// Program **czas.cpp** w języku C++ - wyświetlenie czasu - komentarz w C++

```
#include <cstdlib>
#include <iostream>
#include <time.h>
using namespace std;

int main(int argc, char *argv[])
{
    int a;    time_t czas;
    do {
        cout << "1 - Wyświetl aktualny czas" << endl;
        cout << "2 - Zakończ program" << endl;
        cout << "Twój wybór?";    cin >> a;
        if (a == 1)
        {
            time(&czas);    cout << ctime(&czas);
        }
    }while (a != 2);
    cout << "Do zobaczenia" << endl;

    system("PAUSE");
    return EXIT_SUCCESS;
}
```



```
/* Program znaki1.c */
#include <stdio.h>
#include <conio.h>
#include <string.h>
main (void)
{
char znak1,znak2,znak3; //deklaracja znaków
char slowo1[10], slowo2[15];
char slowo3[]="Adam"; // definicja
clrscr(); // czyszczenie ekranu
znak1='a'; // inicjalizacja zmiennej znak1
znak2=102; // znak w postaci kodu dziesiętnego ASCII - litera f
printf("Podaj znak3: "); scanf("%c",&znak3); // podajemy jakiś znak
printf("\nPodaj slowo1: "); scanf("%s",slowo1); // wpisujemy słowo 1
printf("\nPodaj slowo2: "); scanf("%s",slowo2); // wpisujemy słowo 2
printf("\nZmienne zawieraja znaki: ");
printf("znak1 (a), znak2 (102), znak3 (wprowadzony): %c %c %c ",znak1,znak2,znak3);
printf("\noraz slowa: ");
printf("slowo1, slowo2, slowo3: %s %s %s ",slowo1, slowo2, slowo3);
getch();
return 0 ;
}
```

Program, który podaje kod wciśniętego przez nas klawisza i znak kodu

- */* Program, kodkl1.c - podaje kod wciśniętego przez nas klawisza i znak kodu */:*
- `#include <stdio.h>`
- `#include <conio.h>`
- `#include <string.h>`
- `main (void)`
- `{`
- `char znak;`
- `int kod;`
- `clrscr();`

- `printf("Wciśnij znak na klawiaturze: \n");`
- `scanf("%c",&znak);`
- `printf("Kod wciśniętego znaku to: %d \n",znak);`

- `printf("Podaj kod znaku: \n");`
- `scanf("%d",&kod);`
- `printf("Znak o podanym kodzie to: %c \n",kod);`

- `getch();`
- `return 0 ;`
- `}`

Odczytanie długości łańcucha znaków – strlen(lancuch)

```
/* dluglanc.c */
#include <stdio.h>
#include <conio.h>
#include <string.h>
main (void)
{
int dlugosc;
char *lancuch;
clrscr();
lancuch="Adam Nowak";
dlugosc=strlen(lancuch);
printf("Lancuch '%s' ma: %d znaków \n",lancuch, dlugosc);
printf("Pierwsza litera łańcucha to: %c \n",lancuch[0]);
printf("Ostatnia litera łańcucha to: %c \n",lancuch[dlugosc-1]);
getch();
return 0 ;
}
```

Inne funkcje operujące na łańcuchach: **strlwr,strupr, strcat, strrv, streset – z biblioteki string.h**

strcat() - łączy dwa łańcuchy, **strcmp()** - porównuje dwa łańcuchy rozróżniając małe i duże litery, **strlwr()** i **strupr()** - zamienia w danym łańcuchu duże litery na małe i odwrotnie, **strrev()** - odwraca kolejność znaków w łańcuchu, **strset()** - wypełnia łańcuch danym znakiem.

```
/* znaki5.c - operacje na tekstach */
#include <stdio.h>
#include <conio.h>
#include <string.h>
main (void)
{
char *lancuch1, *lancuch2, *lancuch3;
char znakwyp='x'; // znak wypełniający
clrscr();
puts("Operacje tekstowe\n");
lancuch1="Janusz";
lancuch2="Kowalski";
printf("Lancuch1 to: %s \n",lancuch1);
printf("Lancuch2 to: %s \n",lancuch2);
printf("\nZmieniamy duze litery na male: \n");
strlwr(lancuch1);
printf("Lancuch1 wyglada teraz tak: %s
\n",lancuch1);
```

```
printf("\nZmieniamy male litery na duze: \n");
strupr(lancuch2);
printf("Lancuch2 wyglada teraz tak: %s
\n",lancuch2);
printf("\nLaczymy dwa lancuchy: \n" );
lancuch3=strcat(lancuch1,lancuch2);
printf("Lancuch3 wyglada teraz tak: %s
\n",lancuch3);
printf("\nOdwracamy kolejnosc znakow w
lancuchu: \n" );
strrev(lancuch3);
printf("Lancuch3 wyglada teraz tak: %s
\n",lancuch3);
printf("\nWypelniamy lancuch znakiem 'x':\n" );
strset(lancuch3,znakwyp);
printf("Lancuch3 wyglada teraz tak: %s
\n",lancuch3);
getch();
return 0 ;
}
```

Przykłady programów z if

```
/* Program If1. c */;
#include <iostream.h>
#include <conio.h>
int main()
{
    int a;
    cout << "Podaj liczbe calkowita >0 "; cin >> a;
    if (a<0) return 1; //wyjscie awaryjne z systemu
    cout << "OK"; getch(); return 0; // wyjscie normalne
}
```

```
/* Program If2. c */;
#include <iostream.h>
#include <conio.h>
int main()
{
    int a;
    cout << "Podaj liczbe calkowita > 0 ==> "; cin >> a;
    if (a<0) cout << "a < 0";
    else cout << "a >=0 ";
    getch();
    return 0;
}
```

```

/* program rowkwadr.cpp */
#include <iostream.h>
#include <conio.h>
#include <math.h>
int main()
{
float a, b, c, delta;
clrscr();
cout << "Program oblicza pierwiastki rownania kwadratowego." << endl;
cout << "Podaj wspolczynniki: " << endl;
cout << "a= " ; cin >>a;
cout << "b= "; cin >> b;
cout << "c= "; cin >> c;
delta=b*b-4*a*c;
if (a !=0)
{
if (delta<0) cout << "Delta ujemna - brak rozwiazan.\n" ;
else
if (delta==0) cout << "x= " << (-b+sqrt(delta))/(2*a);
else
if (delta>0)
{
cout << "x1= " << (-b+sqrt(delta))/(2*a) << endl;
cout << "x2= " << (-b-sqrt(delta))/(2*a) << endl;
}
}
else
{
if ((a==0) && (b==0)) cout << "Rownanie ma nieskonczenie wiele rozwiazan.";
else
if ((a==0) && (b!=0))
{
cout << "Rwnanie o podanych wspolczynnkach jest liniowe.\n";
cout << "jego pierwiastek jest rowny: " << -c/b << endl;
}
}
}
getch();
return 0;
}

```

Instrukcja wyboru switch .. case – przykład programu

```
// Program Miesiace.cpp - podaje nazwe miesiaca na podstawie jego numeru
#include <iostream.h>
#include <conio.h>
int main()
{
int miesiac; // numer miesiąca w roku
cout << "Podaj numer miesiaca w roku: ";
cin>> miesiac;
switch (miesiac)
{ // switch
case 1: cout << "Styczen"; break;
case 2: cout << "Luty"; break;
case 3: cout << "Marzec"; break;
case 4: cout << "Kwiecien"; break;
case 5: cout << "Maj"; break;
case 6: cout << "Czerwiec"; break;
case 7: cout << "Lipiec"; break;
case 8: cout << "Sierpien"; break;
case 9: cout << "Wrzesien"; break;
case 10: cout << "Pazdziernik"; break;
case 11: cout << "Listopad"; break;
case 12: cout << "Grudzien"; break;
default: cout << "Numer nie poprawny";
} // switch
getch();
return 0;
}
```

Instrukcje iteracyjne Organizacja obliczeń cyklicznych

Instrukcje iteracyjne służą do wielokrotnego wykonywania pewnych sekwencji instrukcji i zazwyczaj są one nazywane po prostu pętlami.

Instrukcja while - dopóki Warunek wykonuj Instrukcję;

Składnia: `while (wyrażenie) instrukcja;`

Instrukcja do while – powtarzaj Instrukcję dopóki Warunek

Składnia: `do instrukcja while (wyrażenie);`

Instrukcja for: np. dla i od 1 do N - dla $i=W1$ do $W2$ wykonuj Instrukcję

`For (i=W1; i<W2; i++) Instrukcja;`

Składnia:

`for (wyrażenie1; wyrażenie2; wyrażenie3) instrukcja;`

`for (wyrażenie1; wyrażenie2; wyrażenie3) { lista instrukcji }`

Instrukcję for stosuje się w przypadkach, gdy z górną można określić liczbę wykonań pętli.

Przykład programu z while – suma 6 podanych liczb

// Program Suma6.cpp; Oblicza sumę 6 podanych liczb

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
int main()
```

```
{
```

```
int i; float suma, x;
```

```
i=1; // Nadanie wartości początkowej i - licznik
```

```
suma=0; // Nadanie wartości początkowej suma
```

```
cout << "Program oblicza sume podanych 6 liczb " << endl;
```

```
while (i<=6)
```

```
{ // while
```

```
    cout << i << " Podaj liczbe x=";    cin >> x;    suma=suma+x;
```

```
    cout << "\Aktualna suma=" << suma << endl;
```

```
    i++;
```

```
} // while
```

```
cout << "\nSuma koncowa =" << suma << "\nNacisnij Enter" << endl;
```

```
getch();
```

```
return 0;
```

```
}
```

Program z do ... while

// Program ilosc.cpp wyznacza długość ciągu liczb zakończonych 0

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
int main()
```

```
{
```

```
int dlug, liczba, suma=0;
```

```
clrscr();
```

```
cout << "Wprowadzaj liczby, 0 koniec \n";
```

```
dlug=0;
```

```
do
```

```
{
```

```
cin >> liczba;
```

```
dlug = ++dlug;
```

```
suma=suma+liczba;
```

```
}
```

```
while (liczba!=0);
```

```
cout << "Dlug. ciagu wynosi " << --dlug << endl; // 0 nie wliczamy – odejmujemy 1
```

```
cout << "Suma ciagu wynosi " << suma << endl;
```

```
getch();
```

```
}
```

Programy z for

```
/* Program for1.cpp Wydruk cyfr od 1 do 5 */
```

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
int main() {
```

```
int i;
```

```
for (i=1; i<= 5; i++) cout << i << endl;
```

```
getch();
```

```
return 0;
```

```
}
```

```
/* Program for2.cpp Wydruk cyfr od 5 do 1 */
```

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
int main() {
```

```
int i;
```

```
for (i=5; i>= 1; i--) cout << i << endl;
```

```
getch();
```

```
return 0; }
```