

FUNKCJE MATEMATYCZNE, np. atan2(y,x);

```
/* az_tan2a.c - obliczenie azymutu - funkcja atan2(y,x); instrukcja do...while */
#include <stdio.h>
#include <math.h>
#include <conio.h>
#define PI 3.14159265
int main ()
{
    double x1, y1, x2, y2, x, y, result, az, d;
    int kont=1;
    puts("Obliczenie azymutu na podstawie funkcji atan2(dy,dx)");
    puts("tan(Az)=DY/DX");

    do
    {
        printf("\nWprowadz X1 Y1 ");          scanf("%lf %lf", &x1, &y1);
        printf("\nWprowadz X2 Y2 ");          scanf("%lf %lf", &x2, &y2);
        x=x2-x1; y=y2-y1;
        printf("Dx=%f Dy=%f\n",x,y);
        result = atan2 (y,x) * 200.0 / PI;
        az=result;
        if (result <0) az=result+400.0;
        d=sqrt(x*x+y*y);
        printf("\nPI=%lf M_PI=%lf",PI, M_PI);
        printf ("\natan2 dla (DX=%lf, DY=%lf) = %lf[grad]\n", x, y, result);
        printf ("\nDX=%f DY=%f Az = %f[grad] Dl=%f\n", x, y, az, d );
        printf("\n 1- obliczenia, 0 - koniec ");
        scanf("%d",&kont);
    } while (kont != 0);

    return 0;
}
```

TABLICE

// tabl1.cpp - Srednia z liczb - tablica, petla for

```
#include <iostream>
using namespace std;

int main()
{
    int rozmiar;
    cout <<"podaj ilosc liczb: ";
    cin >>rozmiar;
    cin.ignore();

    float tablica[rozmiar];
    float suma = 0;
    for (unsigned int i = 0; i < rozmiar; ++i) //pobieranie liczb...
    { //for
        cout <<"Podaj liczbe nr " <<(i+1)<<" : ";
        cin >>tablica[i];
        cin.ignore();
        suma += tablica[i]; //sumowanie elementow...
    } //for

    cout <<"nsrednia wynosi: " << (suma / rozmiar)<<endl;
    cin.get();

    return 0;
}

/* ===== */
// tabldyn.cpp - Tablica dynamiczna
#include <iostream>
```

```

#include <cstdlib>
using namespace std;

int* Tablica; //Wskaźnik na dynamicznie przydzieloną tablicę
int ile; //Rozmiar tablicy

int main()
{
    cout << "Tablica dynamiczna \n";
    cout << "Jak duza ma byc tablica? => "; cin >> ile;
    Tablica = new int[ile];

    for (int n=0; n<ile; n++) {
        cout << "Podaj element numer " << n << " => ";
        cin >> Tablica[n];
    }

    cin.ignore();
    cout << endl << "Zawartosc tablicy:\n";

    for (int n=0; n<ile; n++)
        cout << n << " => " << Tablica[n] << endl;

    delete [] Tablica;

    cin.get();
    // system("pause");
    return 0;
}

/* ===== */

```

WSKAŹNIKI

// zamianaW.cpp - zmiana wartości parametrów aktualnych przez wskaźnik

```

#include <cstdlib>
#include <iostream>
#include <conio.h>

using namespace std;

void ZmienWsk (int *a, int *b)
{
    int pom;
    cout << "2) Funkcja - przed zamiana: a=x=" << *a << " b=y=" << *b << endl;
    pom =*b;
    *b=*a;
    *a=pom;
    cout << "3) Funkcja po zamianie a=" << *a << " b=" << *b << endl;
}

int main()
{
    int x=10, y=30;
    cout << "Zmiana wartosci zmiennych przez wskaznik" << endl;
    cout << "1) Funkcja glowna przed zamiana: x=" << x << " y=" << y << endl;
    ZmienWsk (&x,&y);
    cout << "4) Funkcja glowna: po zamianie: x=" << x << " y=" << y << endl;
    getch();
}

/* ===== */
// zamianaR.cpp - zamiana wartości parametrów aktualnych przez referencje
#include <cstdlib>
#include <iostream>
#include <conio.h>

```

```

using namespace std;

void ZmienRef (int &a, int &b)
{
    int pom;
    cout << "\n2) Funkcja - przed zamiana: a=x=" << a << " b=y=" << b << endl;
    pom =b;
    b=a;
    a=pom;
    cout << "\n3) Funkcja po zamianie a=" << a << " b=" << b << endl;
}

int main()
{
    int x=10, y=30;
    cout << "Zmiana wartosci zmiennych przez referencje" << endl;
    cout << "Podaj wartosc calkowita x "; cin >> x;
    cout << "Podaj wartosc calkowita y "; cin >> y;
    cout << "\n1) Funkcja glowna przed zamiana: x=" << x << " y=" << y << endl;
    ZmienRef (x,y);
    cout << "\n4) Funkcja glowna: po zamianie: x=" << x << " y=" << y << endl;
    getch();
}

```

```

/* ===== */
TABLICE

```

```

// pp_forT.cpp – Obliczenie pola prostokąta. for, system, cout.precision

```

```

#include <iostream>
#include <cstdlib>
#define MAX 50
#define SW 15
#define WYDR cout.fill(' '); cout.width(10);
using namespace std;

int main() // Program glowny. Najwazniejsza czesc programu
{
    double a, b, p; // zmienne rzeczywiste podwojnej precyzji 8 bajtow, 15 cyfr znacacych
    int i=0, m=0, n=0; // zmienna petli, ilosc faktyczna obl, ilosc zadeklarowana obliczen
    int d; // dokladnosc wydruku po kropce dziesietnej
    double ta[MAX], tb[MAX], tp[MAX];

    // pocz:

    system("cls"); // kasowanie ekranu
    cout << "Program pp_forT.cpp\n";
    cout << "Petla for, tablica" << endl;
    cout << "\nObliczenie pola prostokata o danych bokach a i b \n(zmienne double - 15 cyfr znacacych)" << endl;

    cout<<"\nWprowadz ilosc obliczen < " << MAX << " : " ; // ilosc obliczen
    do {
        cin>> n; // Wprowadzenie a
        cin.ignore(); // Odrzuca Enter
    } while (n > MAX);

    // if (n==0) return 1;

    // Ustalenie dokladnosc wydruku
    cout.width(20); // minimalna ilosc znakow wypisywanej liczby
    cout.setf(ios::fixed); // zapewnia ze precision() odnosi sie do miejsc po kropce
    cout << "\nDokladnosc wydruku ? " ; cin >> d;
    cout.precision(d); // precyzja (patrz wyzej o ios::fixed)

    for (i=1; i<=n; i++)
    { //for - poczatek

```

```

cout << "\nObliczenie: " << i << endl;
cout<<"Wprowadz a (ujemne - pominięte) lub 0 koniec programu: "; // bok a
cin>> a; // Wprowadzenie a
cin.ignore(); // Odrzuca Enter
m=i;
if (a < 0) { i--; continue; }
if (a == 0) { m = --i; break; }
cout<<"Wprowadz b : "; // Pyta o b
cin>> b; cin.ignore(); // Wprowadzenie b; Odrzuca Enter
p=a*b;
cout<<"Pole prostokata o bokach a = " << a << " i b = " << b ;
cout << " wynosi ";
cout << p << endl; // Wyświetla wyniki
ta[i]=a; tb[i]=b; tp[i]=p;
} //for - koniec

if (m>0) cout << "\nWyniki obliczen - wydruk elemntow tablicy w petli for" << endl;
for (i=1; i<=m; i++)
{
cout << "Obliczanie: " << i;
cout << " a = "; WYDR; cout << ta[i];
cout << " b = "; WYDR; cout << tb[i];
cout << " Pole = "; WYDR; cout << tp[i] << endl;
}

// goto pocz;

cout << endl << "Koniec obliczen " << endl;
system("PAUSE");
return 0;
}

/* ===== */

```

TABLICE, ZAPIS DO PLIKU

```

/* pp_forST.c - Pola dzialek prostokątnych. Pętla for, struktura, system, plik */
#include <stdio.h>
#include <conio.h>
#define MAX 50
#define NL putchar('\n')
#define WYN "pprostT.txt"

struct dzialka
{
int nr; char nazwa[10]; float a, b, p;
}t[MAX];

int main()
{
float a, b, pole; int i=0, j=0, n=0;
char flaga = 'T';
FILE *fp; /* zmienna plikowa */
fp = fopen(WYN, "w");
system("cls");
puts("Obliczenie pol dzialek prostokatnych"); NL;

do
{
printf("Podaj dane dzialki %d \n",i+1);
printf("Podaj nr dzialki (liczba calkowita) : ");
scanf("%d",&t[i].nr); fflush(stdin);
printf("Podaj nazwe dzialki      : ");
gets(t[i].nazwa); fflush(stdin);
printf("Podaj dlugosc a: ");
scanf("%f",&t[i].a); fflush(stdin);
}

```

```

printf("Podaj dlugosc b: "); scanf("%f",&t[i].b); fflush(stdin);
pole=t[i].a*t[i].b; printf("\nPole = %f\n",pole);
t[i].p=pole;
i++;
if (i<MAX)
{
printf("\nCzy wprowadzasz nastepne dane? T lub N: ");
flaga=getchar(); putchar('\n'); fflush(stdin);
}
} while (i<MAX && (flaga == 'T' || flaga == 't') );

n=i;
puts("\nZestawienie obliczen pol dzialek\n");
fprintf(fp,"\n Zestawienie obliczen pol dzialek\n");
printf("Lp Nr dz. Nazwa a b pole \n");
fprintf(fp, "Lp Nr dz. Nazwa a b pole \n");

for (j=0; j<n; j++)
{
printf("%3d %5d %6s %7.2f %7.2f %8.2f\n", j+1, t[j].nr, t[j].nazwa, t[j].a, t[j].b, t[j].p );
fprintf(fp,"%3d %5d %6s %7.2f %7.2f %8.2f\n", j+1, t[j].nr, t[j].nazwa, t[j].a, t[j].b, t[j].p );
}
fclose(fp); // zamkniecie pliku
printf("\nWyniki w pliku %s", WYN);
getch();
return 0;
}

/* ===== */

```

// pp_forST1.cpp - Pola dzialek prostokątnych. Pętla for, struktura, system, tablica, plik

```

#include <stdio.h>
#include <iostream>
#include <conio.h>
#include <cstdlib>
#define MAX 50
#define NL putchar('\n')
/* plik wynikow */
#define WYN "pprostT.txt"
using namespace std;

struct dzialka
{ int nr; char nazwa[10]; float a, b, p; };

int main() // Program glowny. Najwazniejsza czesc programu
{

dzialka t[MAX];
float a, b, pole; int i, j=0, n=0; char flaga = 'T';
FILE *fp; /* zmienna plikowa */
fp = fopen(WYN, "w");

system("cls");
puts("Obliczenie pol dzialek prostokatnych"); NL;

i=0;
do
{
printf("Podaj dane dzialki %d \n",i+1);
cout << "Podaj nr dzialki (liczba calkowita) : ";
cin >> t[i].nr; cin.ignore();
printf("Podaj nazwe dzialki : ");
gets(t[i].nazwa); fflush(stdin);
cout << "Podaj dlugosc a: "; cin >> t[i].a; cin.ignore();
cout << "Podaj dlugosc b: "; cin >>t[i].b; cin.ignore();
pole=t[i].a*t[i].b; cout << pole << endl;
}
}

```

```

t[i].p=pole; i++;
if (i<MAX)
{
    cout << "\nCzy wprowadzasz nastepne dane? T lub N: ";
    flaga=getchar(); putchar('\n'); cin.ignore();
}
} while (i<MAX && (flaga == 'T' || flaga == 't') );

n=i;
puts("\nZestawienie obliczen pol dzialek\n");
fprintf(fp," Zestawienie obliczen pol dzialek\n");
cout << "Lp Nr dz. Nazwa a b pole \n";
fprintf(fp, "Lp Nr dz. Nazwa a b pole \n");

for (j=0; j<n; j++)
{
    printf("%3d %5d %6s %7.2f %7.2f %8.2f\n", j+1, t[j].nr, t[j].nazwa, t[j].a, t[j].b, t[j].p );
    fprintf(fp,"%3d %5d %6s %7.2f %7.2f %8.2f\n", j+1, t[j].nr, t[j].nazwa, t[j].a, t[j].b, t[j].p );
}

fclose(fp); // zamkniecie pliku
cout << "Wyniki w pliku " << WYN << endl;
cin.get();
return 0;
}

/* ===== */

```

// PolaDzXY.cpp - pole dzialek ze współrzędnych - tablica

```

#include <iostream>
#include <stdio.h>
#define TAB "\t"
using namespace std;

int main()
{
    int n, rozmiar;
    char dz[20];
    cout << "Obliczenie pol dzialek ze wspolrzednych" << endl;
    cout << "2P=Suma(X[i]*(Y[i=1]-Y[i-1]) -2P=Suma(Y[i]*(X[i=1]-X[i-1]) " << endl;
    do
    {
        cout << "\nPodaj ilosc punktow 0 - koniec: ";
        cin >>n;
        if (n==0) break;
        rozmiar=n+2;
        cin.ignore();

        long tabN[rozmiar];
        float tabX[rozmiar], tabY[rozmiar];
        float s1=0, s2=0;
        int suma = 0;

        cout << "Podaj nazwe dzialki " ;
        gets(dz);
        cout << endl;
        // fflush(stdin);
        // cin.ignore();

        for (unsigned int i = 1; i <= n; ++i) //pobieranie liczb...
        {
            cout <<"Podaj NR punktu "<< i <<" : ";
            cin >>tabN[i];
            cout << "Podaj X Y : ";
            cin >>tabX[i] >> tabY[i];
            cin.ignore();
        }
    }
}

```

```

tabN[0]=tabN[n];
    tabX[0]=tabX[n]; tabY[0]=tabY[n];
tabN[n+1]=tabN[1]; tabX[n+1]=tabX[1]; tabY[n+1]=tabY[1];

cout << "Wykaz punktow: Nr X Y " << endl;
cout << "Lp Nr X Y " << endl;
for(int i=1; i<=n+1; i++ )
{
    if (i<=n)
        cout << i << TAB << tabN[i] << TAB << tabX[i] << TAB << tabY[i] << endl;

    s1+=tabX[i]*(tabY[i+1]-tabY[i-1]);
    s2+=tabY[i]*(tabX[i+1]-tabX[i-1]);
}
s2=-s2;
cout << endl;
cout << "2P= " << s1 << " -2P=" << -s2 << endl;
cout << "Dzialka " << dz << TAB << "Pole = " << 0.25*(s1+s2) << endl;

} while (n >0);

cin.get();

return 0;
}

/* ===== */

// PDXYplik.cpp - pole dzialek ze wspolrzecznych - zapis wynikow do pliku PolaWyn.txt"
// rozmiar tablicy ustalony po uruchomieniu programu podaniu danych
#include <iostream>
#include <stdio.h>
#include <fstream>

#define TAB "\t"
#define WYN "PolaWyn.txt"
using namespace std;

int main()
{
int n, rozmiar;
char dz[20];
cout << "Obliczenie pol dzialek ze wspolrzecznych" << endl;
cout << "2P=Suma(X[i]*(Y[i=1]-Y[i-1]) -2P=Suma(Y[i]*(X[i=1]-X[i-1]) " << endl;
ofstream fw(WYN);

do
{
cout << "\nPodaj ilosc punktow, 0 - koniec: ";
cin >>n;
if (n==0) break;
rozmiar=n+2;
cin.ignore();
long tabN[rozmiar];
float tabX[rozmiar], tabY[rozmiar];
float s1=0, s2=0;
int suma = 0;

cout << "Podaj nazwe dzialki ";
gets(dz);
cout << endl;
// fflush(stdin); // cin.ignore();

for (unsigned int i = 1; i <= n; ++i) //pobieranie liczb...
{
cout <<"Podaj NR punktu "<< i <<" : ";

```

```

cin >>tabN[i];
cout << "Podaj X Y : ";
cin >>tabX[i] >> tabY[i];
cin.ignore();
}

tabN[0]=tabN[n];
tabX[0]=tabX[n]; tabY[0]=tabY[n];
tabN[n+1]=tabN[1]; tabX[n+1]=tabX[1]; tabY[n+1]=tabY[1];

cout << "Wykaz punktow: Nr X Y " << endl;
cout << "Lp Nr X Y " << endl;
for(int i=1; i<=n+1; i++)
{
if (i<=n)
cout << i << TAB << tabN[i] << TAB << tabX[i] << TAB << tabY[i] << endl;

s1+=tabX[i]*(tabY[i+1]-tabY[i-1]);
s2+=tabY[i]*(tabX[i+1]-tabX[i-1]);
}
s2=-s2;
cout << endl;
cout << "2P= " << s1 << " -2P=" << -s2 << endl;
cout << "Dzialka: " << dz << TAB << "Pole = " << 0.25*(s1+s2) << endl;
fw << "Dzialka: " << dz << TAB << "Pole = " << 0.25*(s1+s2) << endl;

} while (n >0);

fw.close();
cout << "Nacisnij cos ";
cin.get();
return 0;
}

```