

Programowanie

Dariusz Wardecki, wyk IV

Powtórzenie

Co wypisze poniższy program?

...

```
short a = 0xfffa;  
cout << "Po rzutowaniu: " << hex << (int)a << endl;
```

...

Powtórzenie

Co wypisze poniższa pętla?

...

```
for(int i=0; i < 100; i+=10)
{
    if(!(i%3))
        cout << i << endl;
}
```

...

Operatory logiczne

! - zaprzeczenie (nie)
&& - iloczyn logiczny (i)
|| - suma logiczna (lub)

```
if ( (x>=3) || (x<0) )  
    return x;
```

```
while ( (i>3) && (i<9) ) {  
    cout << i << endl;  
    i++;  
}
```

Operator warunkowy

warunek ? wynik1 : wynik2

Jeżeli 1 to wynik1
jeżeli 0 to wynik2

```
int a,b,c;  
a=2; b=7;  
c = (a>b) ? a : b;  
cout << c << endl;
```

Funkcje

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

int main(){
    double r;
    double P;
    r = 2;
    P = M_PI*r*r;
    cout << P << endl;

    r = 2.5;
    P = M_PI*r*r;
    cout << P << endl;
    return 0;
}
```

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

double pole(double x){
    return M_PI*x*x;
}

int main(){
    double r;
    double P;
    r = 2;
    P = pole(r);
    cout << P << endl;

    r = 2.5;
    P = pole(r);
    cout << P << endl;
    return 0;
}
```

Funkcje

Deklaracja funkcji

```
typ funkcja(typ arg1, typ arg2, ...)
```

Jaki typ zwraca funkcja?

```
int funkcja(typ arg1, typ arg2, ...)
```

```
double funkcja(typ arg1, typ arg2, ...)
```

itp ...

```
void funkcja(typ arg1, typ arg2, ...)
```

czyli nic nie zwraca ...

Funkcje

Argumenty funkcji

```
typ funkcja(typ arg1, typ arg2, ...)
```

Typy proste

char

short

int

long

float

double

itp ...

ale także ...

funkcje

tablice

wskaźniki

referencje

typy własne

o tym na następnych
wykładach ...

Wywołanie funkcji

```
int suma(int a, int b)
```

```
z = suma( 5 , 7 );
```



lub

```
int x = 5;  
int y = 7;  
z = suma( x , y );
```

Wywołanie funkcji

...

```
void wiadomosc(void)
{
    cout << "Czesc, to ja Twój  
komputer!" << endl;
}
```

```
int main()
{
    wiadomosc();
    return 0;
}
```

Wywołanie funkcji

Przekazywanie parametrów przez nazwę

...

```
void f(float x, float y)
{
    x += 2;
    y += 2;
}
int main()
{
    float x = 3, y = 4;
    f(x, y);
    cout << x << "\t" << y << endl;
    return 0;
}
```

Wywołanie funkcji

Wartości domyślne parametrów

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

int divide (int a, int b=2)
{
    int r;
    r=a/b;
    return (r);
}

int main ()
{
    cout << divide (12);
    cout << endl; // wypisze wartosc 6
    cout << divide (20,4); // wypisze wartosc 5
    return 0;
}
```

Przeładowanie funkcji

Różna liczba argumentów

```
typ funkcja(typ arg1)
```

```
typ funkcja(typ arg1, typ arg2)
```

Różne typy argumentów

```
typ funkcja(int arg)
```

```
typ funkcja(double arg)
```

Przetładowanie funkcji

Przykład

```
int suma() {  
    return 0;  
}  
int suma(int a, int b) {  
    return (a + b);  
}  
int main() {  
    cout << suma() << endl;  
    cout << suma(3, 7) << end;  
    return 0;  
}
```

Funkcje rekurencyjne

Funkcja wywołująca samą siebie

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

int silnia(int x){
    if(n > 1)
        return (x * silnia(x-1));
    else
        return 1;
}
int main(){
    int n = 4;
    cout << n << "! = " << silnia(n) << endl;
    return 0;
}
```

Prototypy funkcji

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

void odd (int a); //To sa prototypy
void even (int a);

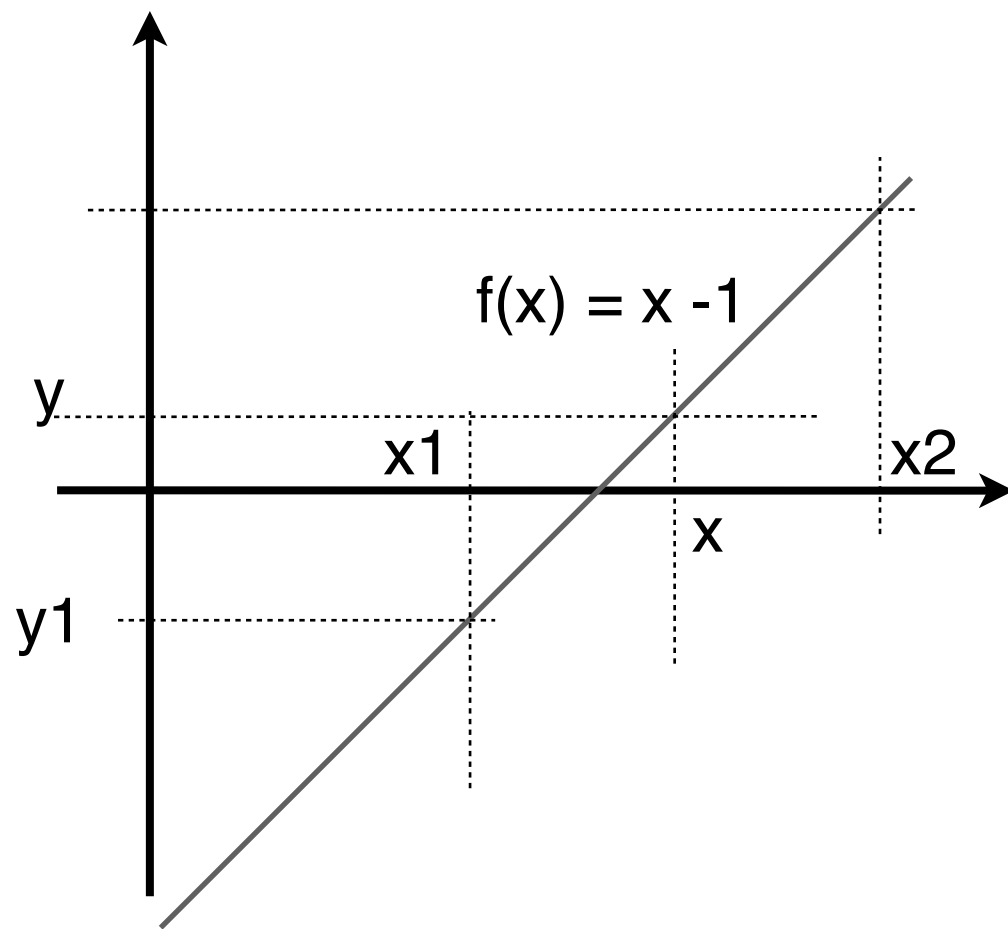
int main (){
    int i;
    do {
        cout << "Type a number (0 to exit): ";
        cin >> i;
        odd (i);
    } while (i!=0);
    return 0;
}

void odd (int a){
    if ((a%2)!=0) cout << "Number is odd.\n";
    else even (a);
}

void even (int a){
    if ((a%2)==0) cout << "Number is even.\n";
    else odd (a);
}
```

Funkcje - przykłady

Poszukiwanie miejsc zerowych (bisekcja)



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

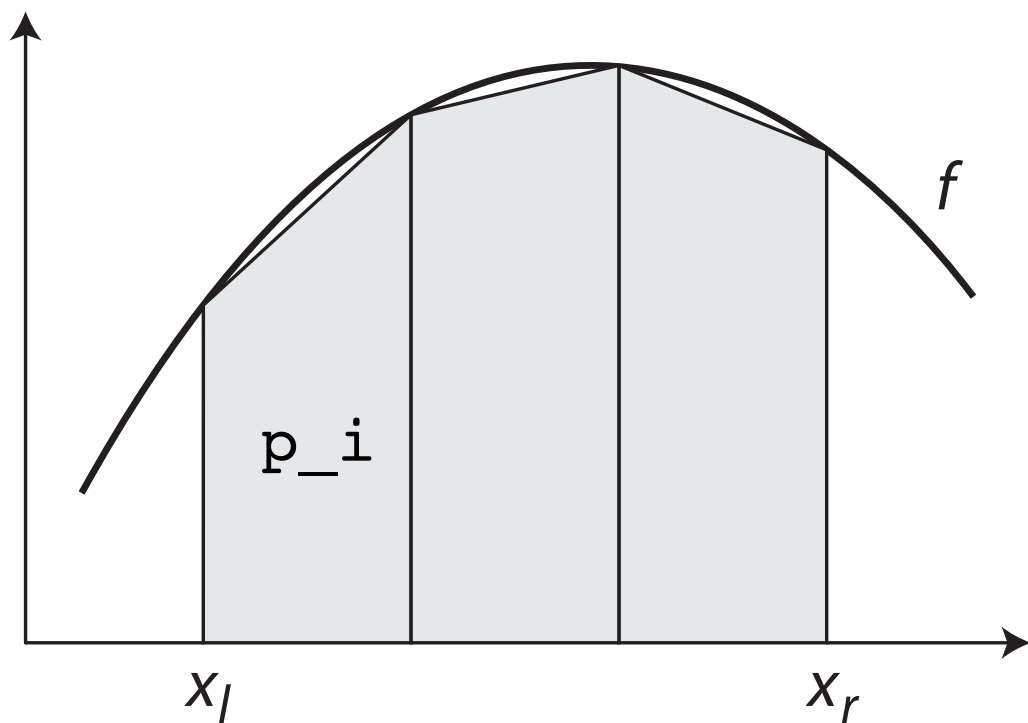
```
double bisection(double x1, double x2, double eps){
    double y1 = x1 - 1; //inna funkcja np. sin(x1)
    while (fabs(x2 - x1) > eps) {
        double x = (x1 + x2)/2;
        double y = x - 1; //analogicznie sin(x)
        if (y*y1 > 0) {
            x1 = x;
            y1 = y;
        }
        else
            x2 = x;
    }
    return (x1 + x2)/2;
}

int main () {
    cout << bisection(0, 2, 1e-10) << endl;
    return 0;
}
```

Funkcje - przykłady

Całkowanie metodą trapezów

$$I = \int_a^b f(x) dx$$



$$p_i = h * (f1 + f2) / 2$$

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

double trapezoid(double a, double b, int n)
{
    double h=(b-a)/n, s=0;
    for (int i=0; i<n; i++)
        s+=h*(sin(a+i*h) + sin(a+(i+1)*h))/2;
    return s;
}

int main ()
{
    cout << trapezoid(0, M_PI, 100) << endl;
    return 0;
}
```

Manual C++

- `http://www.cplusplus.com/doc/tutorial/`