

Programowanie

Dariusz Wardecki, wyk III

Komentarz

Jak szybkie jest Sito Eratostenesa?

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	45	46	47	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

$$\sim N^2$$

$$\sim N\sqrt{N}$$

$p_1, p_2, p_3, \dots, p_i$ - liczby pierwsze

$\sum_{i=1}^N \frac{N}{p_i}$ - liczba wykreśleń

$$\sum_{i=1}^N \frac{N}{i} = N \sum_{i=1}^N \frac{1}{i}, \quad \sum_{i=1}^N \frac{1}{i} \sim \ln N$$

$$\sim N \log N$$

$$\sim N \log(\log N)$$

Powtórzenie

Co wypisze program?

```
...  
cout << 5/2 << endl;  
...
```

```
...  
cout << 5./2 << endl;  
...
```

```
...  
cout << 5%2 << endl;  
...
```

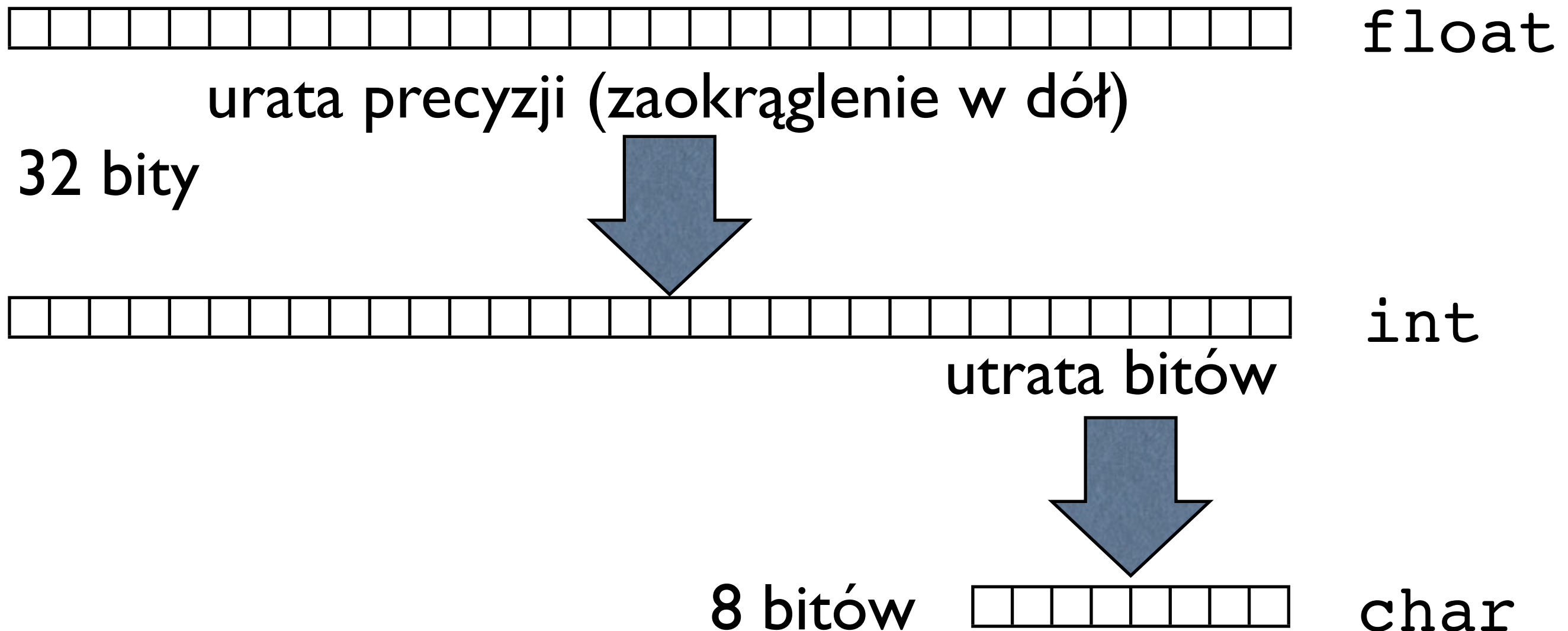
Powtórzenie

Znajdź błąd w kodzie

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

int main( )
    cout << "Hello world!" << endl;
return 0;
```

Rzutowanie typów prostych



Rzutowanie typów prostych

Rzutowanie niejawne - przez przypisanie (operator=)

...

```
int a;  
double b = 2.25;  
a = b;
```

...

...

```
char a = 'A';  
double b;  
b = a;
```

...

Rzutowanie typów prostych

Rzutowanie jawne -
(typ) zmienna

...

```
int a = 5;  
int b = 2;  
cout << (double)a/b;
```

...

...

```
char a = 'A';  
cout << (int)a;
```

...

```
int n;  
cin >> n;  
cout << (char)(n+1);
```

...

Rzutowanie typów prostych

Rzutowanie jawne -
(typ) zmienna

...

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

...

Po rzutowaniu: fffa

Instrukcje warunkowe

```
cout << "Podaj liczbe dodatnia: ";  
cin >> n;
```

```
if(n < 0){  
    cout << "Liczba ujemna!" << endl;  
    return 1;  
}
```

...

Instrukcje warunkowe

```
int n;  
cout << "Podaj liczbe: ";  
cin >> n;  
  
if(n-4){  
    cout << "Liczba rozna od 4!" << endl;  
}  
else  
    cout << "Liczba rowna 4!" << endl;  
...
```

Instrukcje warunkowe

...

```
if(warunek1) instrukcja1;  
else if(warunek2) instrukcja2;  
else if(warunek3) instrukcja3;
```

...

Warunek zwraca wartość logiczną 0(false) lub nie 0(true)

Operatory relacji:

<	==
<=	!=
>	
>=	

Instrukcje warunkowe

```
switch(wyrazenie)
{
    case wart1:
        instr1;
        break;
    case wart2:
        instr2;
        break;
    default:
        instr3;
        break;
}

cin >> n;
switch(n)
{
    case 1:
        cout << "Opcja X\n";
        break;
    case 2:
        cout << "Opcja Y\n";
        break;
    default:
        cout << "Brak opcji\n";
        break;
}
```

Pętle

Policzyć sumę czterech pierwszych wyrazów szeregu $\sum_{n=1}^N \frac{1}{n}$

```
int n = 1;
double x_n;
double S;
S = 1;
n = n + 1;
x_n = 1./n;
S = S + x_n;
n = n + 1;
x_n = 1./n;
S = S + x_n;
n = n + 1;
x_n = 1./n;
S = S + x_n;
```

```
int n = 1;
double x_n;
double S;

S = 1;
pętla(...)
{
    n = n + 1;
    x_n = 1./n;
    S = S + x_n;
}
```

Pętle

```
while(warunek) {  
  
}
```

```
int n = 1;  
double x_n;  
double S;
```

```
S = 1;  
while(n <= 4)  
{  
    n = n + 1;  
    x_n = 1./n;  
    S = S + x_n;  
}
```

Pętle

```
do{  
  
}while(warunek)
```

```
int n = 1;  
double x_n;  
double S;  
  
S = 1;  
do  
{  
    n = n + 1;  
    x_n = 1./n;  
    S = S + x_n;  
} while(n < 4)
```

Pętle

```
for( ; ; ){  
  
}
```

```
int n = 1;  
double x_n;  
double S;
```

```
S = 1;  
for(n;n<=4;n=n+1 )  
{  
    x_n = 1./n;  
    S = S + x_n;  
}
```

Pętle

`break` – natychmiastowe
przerwanie pętli

`continue` – przejście do
następnego kroku

```
int n = 1;  
double x_n;  
double S;
```

```
S = 1;  
for(n; n<=1000; n=n+1)  
{  
    x_n = 1./n;  
    S = S + x_n;  
    if(x_n < 0.01)  
        break;  
}
```

Pętle zagnieżdżone

Przykład - tabliczka mnożenia

...

```
for(int i=1;i<=10;i++){  
    for(int j=1;j<=10;j++){  
        cout << i*j << "\t";  
    }  
    cout << endl;  
}
```

...

Pętle zagnieżdżone

1)

```
...  
for( ... ) {  
    for( ... ) {  
        for( ... )  
    }  
}  
...
```

2)

```
...  
while( ... ) {  
    for( ... ) {  
    }  
}  
...
```

3)

```
...  
for( ... ) {  
    do{  
    } while( ... )  
}  
...
```

itd ...

Operatory arytmetyczne

+ % - reszta z dzielenia

-

*

```
cin >> n;
```

/

```
if(!n%2)
```

=

```
cout << "n parzyste\n";
```

```
a = a + 1;
```

```
b = b - 1;
```

```
jest równoważne
```

```
a++;
```

```
b--;
```

x=x+2	x+=2
x=x-2	x-=2
x=x*2	x*=2
x=x/2	x/=2

Operatory arytmetyczne

inkrementacji		dekrementacji	
pre	post	pre	post
++a	a++	--a	a--

```
int a = 5; b = 5; c = 5; d = 5;
cout << ++a << endl           6
    << b++ << endl           5
    << --c << endl           4
    << c-- << endl;          5
```