

# Arytmetyka wielkich liczb

Wykłady IKA  
Oprac. mgr inż. W. Roszczyński  
18 października 2007

# Typ danych

Określa możliwe wartości, jakie przyjmie zmienna

- char - zmienna przechowuje znaki (litera, cyfry, znaki interpunkcyjne).
- int
- bool
- float - 7 cyfr.
- double - 15 miejsc

# Typ danych

## Dodatkowe kwalifikatory

- Long
- Short
- Signed
- Unsigned

# Typy zmiennych całkowitych

Typy całkowite					
bez znaku - NBC			ze znakiem - U2		
unsigned short int	2B - 16b	Zakres $0 \dots 2^{16} - 1$	short int	2B - 16b	Zakres $-2^{15} \dots 2^{15} - 1$
unsigned int unsigned long int	4B - 32b	Zakres $0 \dots 2^{32} - 1$	int long int	4B - 32b	Zakres $-2^{31} \dots 2^{31} - 1$
unsigned long long int	8B - 64b	Zakres $0 \dots 2^{64} - 1$	long long int	8B - 64b	Zakres $-2^{63} \dots 2^{63} - 1$

# Po co deklaracja typu?

- Shortint - 2B  
<-32768, 32767>  
short x[1000000];

# Po co deklaracja typu?

- Shortint - 2B

<-32768, 32767>

short x[1000000];

- long long - 8B

<-4294967296, 4294967294>

long long x[250000];

- Short <-32768,32767>

```
short a,b;
```

```
a=16400
```

```
b=2*a
```

```
→ b== ??
```

- Short <-32768,32767>

```
short a,b;
```

```
a=16400
```

```
b=2*a
```

```
→ b== -32736
```



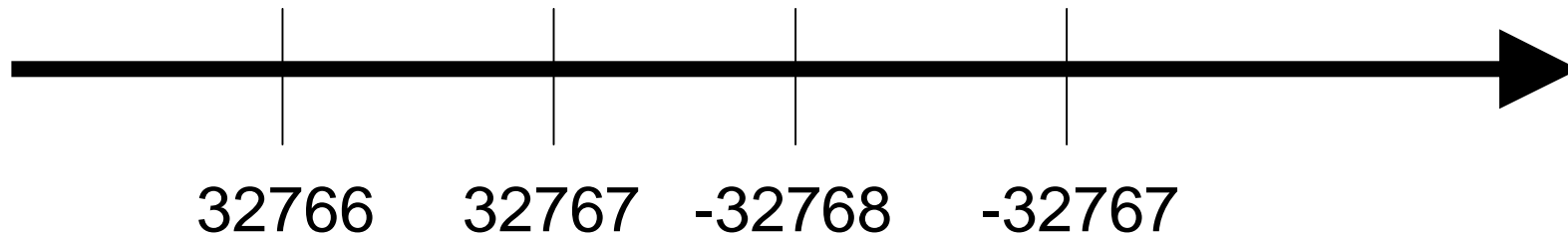
- Short  $\langle -32768, 32767 \rangle$

short a,b;

a=16400

b=2\*a = 32800

→ b== -32736



# I co dalej?

- `long long a;`  
  `a= 4294967294`  
→ `a++?`

38012831038102380123801

+

3726874623846234327648263483684

# Sumujący Jaś

Może to zabrzmieć dziwnie, ale Jaś uwielbia pisemne rachunki. Jaś jest niezmordowany, a koledzy i koleżanki prześcigają się w wymyślaniu dla niego kolejnych przykładów. Nowym wyzwaniem będzie sumowanie liczb zapisanych w postaci ułamków dziesiętnych. W czasie, gdy Jaś będzie pracować, napisz program, który sprawdzi poprawność jego obliczeń.

Napisz program SUM, który:

- wczyta ze standardowego wejścia liczbę, (1 do 100 cyfr, przecinek i od 1 do 100 cyfr po przecinku)
- obliczy ich sumę,
- wypisze wynik na standardowe wyjście.

12345678901234567890123456789012345  
6789012345678901234567890123456789  
0123456789012345678901234567890,12  
3456789012345678901234567890123456  
7890123456789012345678901234567890  
123456789012345678901234567890

+

12345678901234567890123456789012345  
6789012345678901234567890123456789  
0123456789012345678901234567890,12  
3456789012345678901234567890123456  
7890123456789012345678901234567890  
123456789012345678901234567890

# Typ napisowy - string

- Służy do przechowywania napisów
- Może być traktowany jako tablica znaków
- Ma specyficzne funkcje do obsługi
- Nie jest to typ lecz klasa biblioteczna  
`std::string`
- Każdy łańcuch zakończony znakiem `\0`

# Typ napisowy - string

## Deklaracja

- **string** a, napis, zmienna;

Uwaga – niektóre kompilatory wymagają dołączenia biblioteki string,

```
#include<string>
```

# Wczytywanie napisów, wyświetlanie

Wprowadzanie danych

```
string a,b=" ojczyzna moja";  
cin>>a;  
cout<<a<<" "<<b;
```

```
Litwo  
Litwo ojczyzna moja
```

# Działania na stringach

- `int a=2,b=3,c;`

`c=a+b;`

`→ c=??`



# Działania na stringach

- `int a=2,b=3,c;`

`c=a+b;`

→ `c=5`

- `string a="2", b="3", c;`

`c=a+b;`

→ `c=?`

# Działania na stringach

- `int a=2,b=3,c;`

`c=a+b;`

→ `c=5`

- `string a="2", b="3", c;`

`c=a+b;`

→ `c=23`

# Działania na stringach

- Łączenie

```
string a="jest ", b="fajnie ";
```

```
a=a+b;
```

```
→a=="jest fajnie "
```

```
string a="jest ", b="fajnie ";
```

```
a=b+a;
```

```
→a=="fajnie jest "
```

- Inna deklaracja

```
string a("Napis");
```

```
string a="Napis";
```

```
string a(15, '*');
```

# Funkcje

- Aktualny rozmiar napisu

`string napis;`

`napis.length()`

`napis.size()`

dla `napis="Ide do domu"`

`napis.size()` → 11

# Uwaga – spacje

```
string a;
```

```
cin>>a;
```

```
cout<<a;
```

# Uwaga – spacje

```
string a;
```

```
cin>>a;
```

```
cout<<a;
```

Informatyczna Klasa Akademicka

# Uwaga – spacje

```
string a;
```

```
cin>>a;
```

```
cout<<a;
```

```
Informatyczna Klasa Akademicka  
Informatyczna
```



# Uwaga – spacje

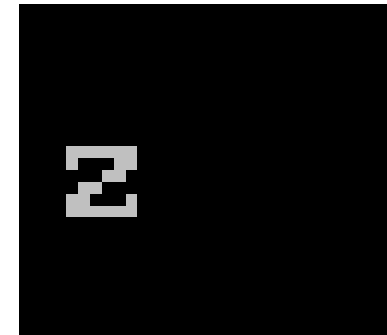
```
string a;  
getline(cin,a);  
cout<<a;
```

```
Informatyczna Klasa Akademicka  
Informatyczna Klasa Akademicka
```

# String – tablica znaków

- Odwoływanie się do konkretnego znaku  
Znaki numerowane od 0 – jak w tablicy

```
string napis="Cisza";  
cout<<napis[3];
```



```
string napis="Kat";  
napis[1]="o";  
napis → Kot
```

# String – tablica znaków

- Przeglądanie znak po znaku

```
string napis="Cisza";  
for (int i=0;i<napis.size();i++)  
{cout<<napis[i]<<endl;}
```

# String – tablica znaków

- Przeglądanie znak po znaku

```
string napis="Cisza";  
for (int i=0;i<napis.size();i++)  
{cout<<napis[i]<<endl;}
```



C  
i  
s  
z  
a

# String – tablica znaków

- Uwaga

```
string napis="Cisza";  
cout<<napis[7];
```



E

# String – tablica znaków - inaczej

```
string napis="Cisza";  
cout<<napis.at(3);
```

```
string napis="Cisza";  
cout<<napis.at(7);
```

```
Out of range  
This application has requested the Runtime to terminate it in an unusual way.  
Please contact the application's support team for more information.
```

# Inne funkcje - usuwanie

- napis.erase(od\_numeru\_znaku,ile\_znaków)  
**string napis="Cisza";**  
**napis.erase(0,2);**  
**cout<<napis;**



**sza**

# Inne funkcje - wstawianie

- napis.insert(gdzie\_wstawić,co\_wstawić)

```
string napis="-  tworca Dziadow";  
napis.insert(0,"Adam Mickiewicz");  
cout<<napis;
```

Adam Mickiewicz- tworca Dziadow



# Inne funkcje

```
napis="Ale porażka"
```

```
napis.copy(napis2, 7, 4)
```

→ napis2=porażka

```
int pozycja;
```

```
pozycja=napis.find("pora")
```

→ pozycja=4

# Zamiana pojedynczego znaku na liczbę

```
string napis="6";  
napis=napis+napis;  
→ napis="66"
```

```
string napis="4234";  
napis=napis+napis;  
→ napis=" 42344234 "
```

# Zamiana pojedynczego znaku na liczbę (rzutowanie)

```
string napis="432342";  
int liczba;
```

Zamiana pojedynczego znaku

```
liczba=int(napis[3])
```

→ liczba=51



51

# Zamiana pojedynczego znaku na liczbę (rzutowanie)

```
string napis="432342";
```

```
int liczba;
```

Zamiana pojedynczego znaku

```
liczba=int(napis[3])-48;
```

lub

```
liczba=int(napis[3]-'0');
```

→ liczba=3

**Dodanie dwóch wielkich liczb**

Jak dodajemy pisemnie

**12893879**

**13152**

---

# Jak dodajemy pisemnie

$$\begin{array}{r} 12893879 \\ 13152 \\ \hline \end{array}$$

$$2+9=11$$

# Jak dodajemy pisemnie

A handwritten addition problem is shown. The first number, 12893879, is written in black ink above a green horizontal line. The second number, 13152, is written in black ink below the green line. A black horizontal line is drawn under the second number. The sum, 12893879 + 13152 = 12907031, is not fully visible. A red vertical line is drawn between the units and tens columns. A red '1' is written above the green line, to the left of the red vertical line, indicating a carry from the units column. Another red '1' is written below the black line, to the right of the red vertical line, indicating a carry from the tens column. The text „w pamięci” is written in black ink to the right of the red vertical line, above the green line.

$$\begin{array}{r} 12893879 \\ + 13152 \\ \hline \end{array}$$

„w pamięci”



# Jak dodajemy pisemnie

1

„w pamięci”

12893879

13152

1

$1+7+5=13$

# Jak dodajemy pisemnie

A handwritten addition problem is shown. The numbers 12893879 and 13152 are aligned by their rightmost digits. A horizontal green line is drawn above the numbers. Two vertical red lines are drawn: one between the 8 and 7, and another between the 7 and 9. The sum 31 is written below the horizontal line, aligned with the two red lines. To the right of the numbers, the text „w pamięci” is written, and the calculation 1+7+5=13 is shown.

$$\begin{array}{r} 12893879 \\ 13152 \\ \hline 31 \end{array}$$

„w pamięci”

$$1+7+5=13$$

# Jak dodajemy pisemnie

	11		„w pamięci”
<hr/>			
12893879			
	13152		$1+8+1=10$
<hr/>			
	031		

Jak dodajemy pisemnie

$$\begin{array}{r} 12893879 \\ \phantom{00}13152 \\ \hline 12907031 \end{array}$$

# Obsługa kolumny

```
int wpamieci,p,q;
```

```
.....
```

```
wynik=wpamieci+p+q;
```

```
wpamieci=?
```

```
wynik=?
```

Np. wynik = 13

# Obsługa kolumny

```
int wpamieci,p,q;
```

```
.....
```

```
wynik=wpamieci+p+q;
```

```
wpamieci=wynik/10;
```

```
wynik=wynik%10; //lub wynik%=10
```

# Dane w łańcuchach – zamiana na liczby

```
string l1,l2,pom;
```

```
int wpamieci,p,q;
```

```
..... np.
```

```
p=? l1= "312312321312312,123123123"
```

```
q=? l2= "234423423342443,867876867"
```

```
wynik=wpamieci+p+q;
```

```
wpamieci=wynik/10;
```

```
wynik=wynik%10;
```

# Dane w łańcuchach – zamiana na liczby

```
string l1,l2,pom;  
int wpamieci,p,q;  
  
.....  
p=int(l1[i]-'0');  
q=int(l2[i]-'0');  
wynik=wpamieci+p+q;  
wpamieci=wynik/10;  
wynik=wynik%10;
```



# Dane w liczbie – dopisanie do łańcucha wynikowego

```
string l1,l2,pom;  
int wpamieci,p,q;  
.....  
p=int(l1[i]-'0');  
q=int(l2[i]-'0');  
wynik=wpamieci+p+q;  
wpamieci=wynik/10;  
wynik=wynik%10;  
pom=char(wynik+'0')+pom;
```

# Dodanie dwóch łańcuchów w/w w pętli

```
string l1,l2,pom;  
int wpamieci=0,p,q,wynik;  
pom="";  
for (int i=200;i>=0;i--)  
{  
    if (i!=100){  
        p=int(l1[i]-'0');  
        q=int(l2[i]-'0');  
        wynik=wpamieci+p+q;  
        wpamieci=wynik/10;  
        wynik=wynik%10;  
        pom=char('0'+wynik)+pom;  
    }  
    else pom='.'+pom;}  
}
```

# Dodanie dwóch łańcuchów w/w w pętli

```
string l1,l2,pom;  
int wpamieci=0,p,q,wynik;  
pom="";  
for (int i=200;i>=0;i--)  
{  
    if (i!=100){  
        p=int(l1[i]-'0');  
        q=int(l2[i]-'0');  
        wynik=wpamieci+p+q;  
        wpamieci=wynik/10;  
        wynik=wynik%10;  
        pom=char('0'+wynik)+pom;  
    }  
    else pom='.'+pom;}  
}
```

# Dodanie dwóch łańcuchów w/w w pętli

```
string l1,l2,pom;  
int wpamieci=0,p,q,wynik;  
pom="";  
for (int i=200;i>=0;i--)  
{  
    if (i!=100){  
        p=int(l1[i]-'0');  
        q=int(l2[i]-'0');  
        wynik=wpamieci+p+q;  
        wpamieci=wynik/10;  
        wynik=wynik%10;  
        pom=char('0'+wynik)+pom;  
    }  
    else pom='.'+pom;}  
}
```

# Dodanie dwóch łańcuchów w/w w pętli

```
string l1,l2,pom;  
int wpamieci=0,p,q,wynik;  
pom=" ";  
for (int i=200;i>=0;i--)  
{  
    if (i!=100){  
        p=int(l1[i]-'0');  
        q=int(l2[i]-'0');  
        wynik=wpamieci+p+q;  
        wpamieci=wynik/10;  
        wynik=wynik%10;  
        pom=char('0'+wynik)+pom;  
    }  
    else pom='.'+pom;}  
}
```

# Dodanie dwóch łańcuchów w/w w pętli

```
string l1,l2,pom;  
int wpamieci=0,p,q,wynik;  
pom="";  
for (int i=200;i>=0;i--)  
{  
    if (i!=100){  
        p=int(l1[i]-'0');  
        q=int(l2[i]-'0');  
        wynik=wpamieci+p+q;  
        wpamieci=wynik/10;  
        wynik=wynik%10;  
        pom=char('0'+wynik)+pom;  
    }  
    else pom='.'+pom;}  
}
```

# Dodanie dwóch łańcuchów w/w w pętli

```
string l1,l2,pom;  
int wpamieci=0,p,q,wynik;  
pom="";  
for (int i=200;i>=0;i--)  
{  
    if (i!=100){  
        p=int(l1[i]-'0');  
        q=int(l2[i]-'0');  
        wynik=wpamieci+p+q;  
        wpamieci=wynik/10;  
        wynik=wynik%10;  
        pom=char('0'+wynik)+pom;  
    }  
    else pom='.'+pom;}  
}
```

# Dodanie dwóch łańcuchów w/w w pętli

```
string l1,l2,pom;  
int wpamieci=0,p,q,wynik;  
pom="";  
for (int i=200;i>=0;i--)  
{  
    if (i!=100){  
        p=int(l1[i]-'0');  
        q=int(l2[i]-'0');  
        wynik=wpamieci+p+q;  
        wpamieci=wynik/10;  
        wynik=wynik%10;  
        pom=char('0'+wynik)+pom;  
    }  
    else pom='.'+pom;}  
}
```



# Dodanie dwóch łańcuchów w/w w pętli

```
string l1,l2,pom;  
int wpamieci=0,p,q,wynik;  
pom="";  
for (int i=200;i>=0;i--)  
{  
    if (i!=100){  
        p=int(l1[i]-'0');  
        q=int(l2[i]-'0');  
        wynik=wpamieci+p+q;  
        wpamieci=wynik/10;  
        wynik=wynik%10;  
        pom=char('0'+wynik)+pom;  
    }  
    else pom='.'+pom;}  
}
```

# Dodanie dwóch łańcuchów w/w w pętli

```
string l1,l2,pom;  
int wpamieci=0,p,q,wynik;  
pom="";  
for (int i=200;i>=0;i--)  
{  
    if (i!=100){  
        p=int(l1[i]-'0');  
        q=int(l2[i]-'0');  
        wynik=wpamieci+p+q;  
        wpamieci=wynik/10;  
        wynik=wynik%10;  
        pom=char('0'+wynik)+pom;  
    }  
    else pom='.'+pom;}  
}
```

# Dodanie dwóch łańcuchów w/w w pętli

```
string l1,l2,pom;  
int wpamieci=0,p,q,wynik;  
pom="";  
for (int i=200;i>=0;i--)  
{  
    if (i!=100){  
        p=int(l1[i]-'0');  
        q=int(l2[i]-'0');  
        wynik=wpamieci+p+q;  
        wpamieci=wynik/10;  
        wynik=wynik%10;  
        pom=char('0'+wynik)+pom;  
    }  
    else pom='.'+pom;}  
}
```

# Dodanie dwóch łańcuchów w/w w pętli

```
string l1,l2,pom;  
int wpamieci=0,p,q,wynik;  
pom="";  
for (int i=200;i>=0;i--)  
{  
    if (i!=100){  
        p=int(l1[i]-'0');  
        q=int(l2[i]-'0');  
        wynik=wpamieci+p+q;  
        wpamieci=wynik/10;  
        wynik=wynik%10;  
        pom=char('0'+wynik)+pom;  
    }  
    else pom='.'+pom;}
```

# Dodanie dwóch łańcuchów w/w w pętli

```
string l1,l2,pom;  
int wpamieci=0,p,q,wynik;  
pom="";  
for (int i=200;i>=0;i--)  
{  
    if (i!=100){  
        p=int(l1[i]-'0');  
        q=int(l2[i]-'0');  
        wynik=wpamieci+p+q;  
        wpamieci=wynik/10;  
        wynik=wynik%10;  
        pom=char('0'+wynik)+pom;  
    }  
    else pom='.'+pom;}  
}
```

# Czego jeszcze brakuje?

- Wczytywania danych
- Normalizacji (tak, aby każda liczba miała 100 cyfr, przecinek, 100 cyfr)
- Obcinania wyników  
tzn. jak wyjdzie 0000043242,423420000  
skrót do 43242,42342

Wykład dostępny:

[www.lo8.poznan.pl](http://www.lo8.poznan.pl)

→ IKA