

Lekcja 7

Teksty

znaki – łańcuchy znaków – napisy string – przekształcenia string – konwersja liczba-tekst

Zmienne znakowe – char (apostrofy)

Przykłady w ramce opisują: deklarowanie i inicjowanie zmiennej znakowej, operacje dodawania liczby do zmiennej znakowej, różnicę pomiędzy znakiem '0' liczbą 0 oraz konwersję znaku na liczbę i liczby na znak, kody ASCII i tzw. rzutowanie.

```
char znak;  
znak='A';  
cout << znak; //A  
//konwersja - znak-liczba  
cout << znak << int(znak) << endl; //A 65  
cout << znak << (int) znak << endl; //A 65  
cout << znak << char(65) << endl; //A A  
//znaki jak liczby!!!  
cout << znak+0 << znak+1 << endl; //65 66  
cout << '1'+1 << 1+1 << endl; //98 2
```

UWAGA. W języku C++ instrukcja `cout << '1'+1` wyświetli na ekranie liczbę 98 (bo kod ASCII dla znaku '1' to 49). Wszystkie normalne języki programowania „widząc” znak traktują go jak znak i można go co najwyżej skleić z innym znakiem, tylko w C++ jest „dziwnie”. Sam sposób zapisu konwersji znak-liczba też można w C++ „udziwnić”. Typowy sposób to np. `char(liczba)`. W C++ można dodatkowo (`char`)65, albo też (`char`)(65).

Zadanie – Tablica ASCII

Wyświetl na ekranie tablicę kodów ASCII. Uporządkuj ją w 8 kolumnach: znak-kod.

```
for (int i=33; i<256; i++){  
    cout.width(4);  
    cout << (char) (i);  
    cout.width(4);  
    cout << i;  
    if (i % 8 == 0) cout << endl;  
}
```

```
string nap="Wacław Libront";  
cout << nap << endl;  
//długość napisu - LENGTH  
int dlug=nap.length();  
//doklejanie za pomocą operatora +  
nap="mgr inż "+nap;  
cout << nap << endl;  
//doklejanie na końcu - APPEND  
nap=nap.append(" Bobowa");  
cout << nap << endl;  
//wstawianie do napisu - INSERT  
nap=nap.insert(15,"Stanisław ");  
cout << nap << endl;  
//usuwanie fragmentu napisu - ERASE  
nap=nap.erase(15,10);  
cout << nap << endl;  
//wybieranie znaku z napisu - AT  
char znak=nap.at(6);  
cout << znak << endl;  
//wybieranie znaku z napisu - tablica  
znak=nap[6];  
cout << znak << endl;  
//wyszukiwanie znaków i napisów - FIND  
int spacja=nap.find(" ");  
cout << spacja << endl;  
//wyszukiwanie kolejnej spacji w tekście  
cout << nap.find(" ",spacja+1) << endl;  
//wycinanie fragmentów - SUBSTR  
string frag=nap.substr(spacja+1,3);  
cout << frag << endl;
```

Zmienne napisowe STRING

W standardowym C nie było mowy o typowych zmiennych tekstowych (string), jak w innych językach programowania. Twórca – dążąc do minimalizacji wymyślił łańcuchy znakowe (tablice), co w praktyce spowodowało mnóstwo problemów. Dodatkowa biblioteka `<string>` porządkuje i upraszcza pracę z tekstami. Co więcej, niektóre wersje kompilatorów (np. Dev) potrafią operować na napisach bez tej biblioteki, ale już MSVC++ musi mieć ją zadeklarowaną!

Uwaga. Pierwsza litera w napisie ma w tablicy znaków numer zero!

LENGTH – długość napisu

APPEND – doklejanie na końcu

INSERT – wstawianie

ERASE – usuwanie

AT lub [] – znak

FIND – wyszukiwanie

SUBSTR - wycinanie

```
Wacław Libront  
mgr inż Wacław Libront  
mgr inż Wacław Libront Bobowa  
mgr inż Wacław Stanisław Libront Bobowa  
mgr inż Wacław Libront Bobowa  
z  
z  
3  
7  
inż
```

Zadania z napisami

Wpisz z klawiatury (lub zadeklaruj w zmiennej) napis, w którym będzie twoje Nazwisko i Imię.

- 1) wyodrębnij nazwisko i imię, do dwóch zmiennych
- 2) stwórz anagramy nazwiska i imienia
- 3) zamieniamy miejscami nazwisko z imieniem
- 4) zamień miejscami pierwsze litery
- zlepek Nmię Iazwisko
- 5) wyświetl inicjały NI

Konwersja tekst-liczba

Dowolny napis przedstawiający liczbę przekonwertujemy na prawdziwą liczbę za pomocą funkcji **atoi** (na całkowitą) lub **atof** (rzeczywistą)

```
cout << atoi("213")+1 << endl;
cout << atof("213")+1.1 << endl;
```

Konwersja liczba tekst

Konwersja w drugą stronę jest w języku C problematyczna i twórcy nie wykazali się pomysłowością. Istnieje mnóstwo sposobów wymyślonych przez niezależnych twórców, ale żaden z nich nie jest doskonały i nie ma jednej wygodnej funkcji

```
int liczba=123;
char L[20];
itoa(liczba,L,10);
cout << strlen(L) << endl;
```

```
#include <sstream>
...
stringstream zmienna;
zmienna << 123.31;
string tekst=zmienna.str();
cout<<tekst.length()<<endl;
```

Zamiana liczby całkowitej na tekst za pomocą funkcji **itoa** jest jak najbardziej możliwa, ale ... twórca wymyślił, że będzie też możliwość zamiany na dowolny system pozycyjny – w naszym przykładzie podstawą systemu jest 10.

Nie ma niestety funkcji **ftoa** i jeśli chcemy zamienić liczbę rzeczywistą na tekst możemy posłużyć się biblioteką **<sstream>**. Do zmiennej strumieniowej wlewamy liczbę, a następnie zamieniamy ją na stringa.

Zadanie – Zamiana DWÓJKOWE-DZIESIĘTNE

Wpisz liczbę dwójkową z klawiatury (w postaci tekstu) i zamień na jej dziesiętny odpowiednik. Napisz odpowiednią funkcję zamieniającą dwójkowy zapis na liczbę dziesiętną.

Jak zamienić liczbę dwójkową na dziesiętkową? Z matematycznego punktu widzenia nie jest to trudne – wystarczy zsumować tzw. wagi cyfr w liczbie dwójkowej. Co to oznacza? Każda cyfra w liczbie jest numerowana – najmniej znacząca (pierwsza z prawej) jest na **pozycji** zerowej, kolejna na pozycji pierwszej, itd. Wystarczy liczbę 2 (**podstawę** systemu dwójkowego) podnieść do potęgi, która jest cyfrą na tej pozycji. Te wszystkie potęgi (tzw. **wagi bitów**) należy zsumować – i to jest będzie liczba w systemie dziesiętkowym. Funkcja **DW_DZ()** wykonuje te wszystkie operacje, w jak najbardziej „edukacyjny” sposób. Podczas wyodrębniania poszczególnych bitów z napisu będziemy odwoływać się do tablicy napisu, pamiętając o tym, że bity musimy pobierać od tyłu!

```
string NI="Libront Wacław";
//długość i spacja
int DLUG=NI.length();
int SPAC=NI.find(" ");
//wyodrębnienie wyrazów
string IMIE=NI.substr(0,SPAC);
string NAZW=NI.substr(SPAC+1,DLUG-SPAC);
cout << IMIE << " " << NAZW << endl;
//pierwsze litery - inicjały
char INII=IMIE[0];
char ININ=NAZW[0];
cout << INII << " " << ININ << endl;
// zamieniamy literki
cout<<IMIE[0]<<NAZW.substr(1,NAZW.length());
cout<<" ";
cout<<NAZW[0]<<IMIE.substr(1,IMIE.length());
cout << endl;
//zamieniamy na małe <> duże
for (int i=0;i<DLUG;i++)
    if (NI[i]>=65 && NI[i]<=90)
        cout << char(NI[i]+32);
    else
        if (NI[i]>=97 && NI[i]<=122)
            cout << char(NI[i]-32);
        else cout << NI[i];
cout << endl;
//anagramy
for (int i=NAZW.length()-1;i>=0;i--)
    cout << NAZW[i];
for (int i=IMIE.length()-1;i>=0;i--)
    cout << IMIE[i];
cout << endl;
```

```
int DW_DZ(string dw){
    const int POD=2;
    char bit;
    int waga;
    int poz=0;
    int dz=0;
    int dlug=dw.length();
    do {
        bit=dw[dlug-poz-1];
        waga=pow(POD,poz);
        if (bit!='0')
            dz=dz+waga;
        poz++;
    } while (poz<dlug);
    return dz;
}
```

Zadanie – Zamiana DZIESIĘTNE- DWÓJKOWE

Wpisz z klawiatury liczbę dziesiętną. Na ekranie wypisz jej dwójkowy odpowiednik. Napisz funkcję zamieniającą liczby dziesiętne na dwójkowe, której wynikiem jest tekst zero-jedynkowy.

Dzielimy całkowicie liczbę przez dwa. Reszty z dzielenia (zera lub jedynki) sklejamy w liczbę dwójkową – kolejne bity doklejamy zawsze na początek! Wynik z dzielenia całkowitego przez dwa staje się na końcu naszą liczbą podstawową i całą operację powtarzamy, dopóki można wykonywać dzielenie.

Uwaga. Należy rozróżnić zmienną napisową `dw`, która na początku jest tzw. „łańcuchem pustym” od znaków ‘0’ i ‘1’ w apostrofach, które są doklejane do łańcucha `dw` za pomocą zwykłej operacji dodawania.

```
string DZ_DW(int dz){
    string dw="";
    int bit;
    do {
        bit=dz % 2;
        if (bit==1) dw='1'+dw;
        else dw='0'+dw;
        dz=dz/2;
    } while (dz>0);
    return dw;
}
```

Wczytywanie tekstów z klawiatury

Do tej pory wczytywaliśmy z klawiatury (za pomocą polecenia `cin`) do zmiennych pojedyncze liczby lub teksty. Kłopoty pojawiają się ponieważ `cin` nie reaguje na spacje i enter. Wystarczy uruchomić sprawdzic ciąg instrukcji z ramki i wpisać z klawiatury wielowyrazowy napis, np. „Nazwisko Imię” rozdzielone spacją. Po wciśnięciu klawisza ENTER na ekranie wyświetlone zostanie tylko „Nazwisko”, gdyż `cin` wczytuje znaki do napotkania spacji! Jak sobie można z tym poradzić? Jednym z rozwiązań jest zastosowanie instrukcji `getline(cin, zmienna)`.

```
string N;
cin>>N;
cout<<N;
```

```
Nazwisko Imię
Nazwisko
```

UWAGA. Jeśli będziemy mieszać w programie `cin` i `getline` pojawiają się kolejne problemy związane z tzw. buforem klawiatury ☺

Zadanie – Ulica i numer domu

Wczytaj w pętli do tablicy ulicę wraz z numerem domu, aż do naciśnięcia klawisza ENTER – wczytania pustego tekstu.

Zadanie zrealizowano za pomocą instrukcji `getline` i pętli `while`, która kończy swoje działanie, gdy w zmiennej, do której wczytujemy napisy z klawiatury będzie tekst pusty. Na końcu wypisujemy wszystkie wczytane napisy na ekranie.

```
Ulica: Długoszkowskich 1
Ulica: Rynek 13/12
Ulica: Grunwaldzaka 72 a
Ulica:
Długoszkowskich 1
Rynek 13/12
Grunwaldzaka 72 a
```

```
int ile=0;
string u;
string ulice[100];
do{
    cout<<"Ulica: ";
    getline(cin,u);
    if (u!="") ulice[ile]=u;
    ile++;
} while(u!="");
for (int i=0;i<ile;i++)
    cout<<ulice[i]<<endl;
```

Zadania

- 1) Napisz funkcję ANAGRAM, która zamieni podany jako parametr tekst na jego odwrotność
- 2) Napisz funkcję, która utworzy tekst złożony z losowego zlepków liter podanych w parametrze
- 3) Napisz funkcję, która policzy ilość liter w tekście. Jako parametr podajemy tekst i literę
- 4) Napisz funkcję, która policzy wszystkie litery w tekście – bez spacji
- 5) Napisz funkcję PALINDROM, która sprawdzi czy wpisany wyraz jest palindromem (czytany od tyłu jest taki sam).
a) Zmodyfikuj funkcję aby sprawdzała całe zdania - usuń najpierw spacje z tekstu!
- 6) Mini szyfrowanie. Napisz funkcję, która zwróci zaszyfrowany tekst. Zasada szyfrowania: kod ASCII każdego znaku zwiększamy o konkretną liczbę
- 7) Mini szyfrowanie. Napisz funkcję, która zwróci zaszyfrowany tekst. Zasada szyfrowania: przestawiamy znaki w tekście 1-2 3-4 5-6 itd.
- 8) Napisz funkcję, która zliczy samogłoski (spółgłoski) w tekście
- 9) Napisz funkcję, która zliczy wyrazy w tekście. Zliczamy spacje +1
- 10) Dwie duże liczby znajdują się w łańcuchach tekstowych. Wykonaj ich dodawanie.