Podstawy programowania

Stringi

Napisy: 0-terminated

- Napisy traktowane jako tablice zakończone znakiem o kodzie 0:

```
 1  #include <cstdlib>
 2  #include <iostream>
 3  #include <cstring>
 4
 5  using namespace std;
 6
 7  int main(int argc, char** argv) {
 8
 9  char tekst[90] = "Ten tekst musi byc krotszy niz 89 znakow";
10  char nap[10] = "123456789";
11  char napis[] = "Ala ma kota";
12
13  cout << tekst << endl << napis <<endl << nap << endl;
14
```

Napisy: 0-terminated

- Porównywanie: strcmp(str1, str2): zwraca -, + lub 0
- Kopiowanie: strcpy(str1, str2) – znaczy to samo co intuicyjne: str1 = str2;
- Konkatenacja: strcat(str1, str2) – znaczy to samo co intuicyjne: str1 = str1 + str2;
- tolower, toupper, islower, isupper – trywialne
- Konwersje:
  - atol, strtol - zamienia łańcuch na liczbę całkowitą typu long
  - atoi - zamienia łańcuch na liczbę całkowitą typu int
  - atoll, strtoll - zamienia łańcuch na liczbę całkowitą typu long long (64 bity); dodatkowo istnieje przestarzała funkcja atoq będąca rozszerzeniem GNU,
  - atof, strtod - przekształca łańcuch na liczbę typu double

*Ćwiczenia

1. Zadeklaruj napis c-string o długości 10 i przypisz mu 11 znaków (najpierw w deklaracji, a potem za pomocą konkatenacji)
2. Przeczytaj z klawiatury kilka napisów i dodawaj je do jednej zmiennej c-string. Jeśli będzie za dużo znaków, poinformuj o tym użytkownika.
3. Zmień małe na wielkie litery i odwrotnie w napisie przeczytanym z klawiatury.

Napisy: the easy way

- Jesteśmy uratowani! W C++ jest string!

```
 9 string s;
10 cin >> s; //sproboj tu podac cos ze spacja :)
11 cin.ignore();
12 fflush(stdin);
13 cout << s << endl;
14 getline(cin,s);
15 cout << s << endl;
16 string drugi="Ala ma kota";
17 if(s==drugi) cout << "Yupi!"; else cout << "Niestety";
18 cout << endl;
19 s+=" i psa";
20 cout << s << endl;
```

```
35
36 string zdanie="Byc, albo nie byc - oto jest pytanie";
37 cout << zdanie <<endl;
38 for(int i=0;i<zdanie.size();i+=2) { //pokemonizator
39         zdanie.at(i) = toupper(zdanie.at(i));
40         }
41 cout << zdanie;
```

```
24 string csv;
25 do {
26      cin >> csv;
27      cout << csv << endl;
28 }
29 while(csv!=";");
30
31 csv="";
32 getline(cin,csv,'$');
33 cout << csv << endl;
```

Napisy: the easy way

- Pozostałe metody:
  - compare(str1, str2) – analogiczne do tego z C
  - c_str(str) – zwraca „stary” łańcuch 0terminated
  - find, find_first_not_of, find_first_of, …
  - str1.replace(start,length,sourceForReplacement)
  - substr
  - swap

*Ćwiczenia

1. Czym różni się operator [] od at? Sprawdź to praktycznie ☺
2. Zaimplementuj jeden z szyfrów harcerskich: GA-DE-RY-PO-LUKI, PO-LI-TY-KA-RE-NU, KA-CE-MI-NU-TO-WY.
3. Napisz funkcję sprawdzającą czy wyraz jest palindromem. Tu konkurs na najkrótszy program ☺
4. Napisz program, który liczy liczbę wyrazów podanych przez użytkownika.
5. Napisz program odwracający podany przez użytkownika napis z klawiatury.
6. Napisz bardzo prosty programik typu „książka adresowa”. Użytkownik podaje imię, nazwisko i nr gg kontaktów. Są one zapisywane do tablicy. Program ma umożliwiać wyszukiwanie po numerze gg i sortowanie po nazwisku.

Obsługa plików

- I znowu – można jak w C, za pomocą „starych” struktur i metod:

```c
#include <stdio.h>
#include <stdlib.h>

int main ()
{
  FILE *fp; /* używamy metody wysokopoziomowej - musimy mieć zatem identyfikator pliku, uwaga na gwiazdkę! */
  char tekst[] = "Hello world";
  if ((fp=fopen("test.txt", "w"))==NULL) {
    printf ("Nie mogę otworzyć pliku test.txt do zapisu!\n");
    exit(1);
    }
  fprintf (fp, "%s", tekst); /* zapisz nasz łańcuch w pliku */
  fclose (fp); /* zamknij plik */
  return 0;
}
```

* … ale my się uczymy C++ i mamy strumienie:

Odczyt (istream)

```
1  #include <iostream>
2  #include <fstream>
3
4  int main()
5  {
6     char znak;
7     std::ifstream plik("testowy.txt");
8     do {
9        znak = plik.get();
10       std::cout << znak;
11    } while (znak != '.');
12
13 system("PAUSE");
14 return 0;
15 }
```

```
6  std::string linia;
7  std::ifstream plik("testowy.txt");
8
9     do {
10       getline(plik,linia);
11       std::cout << linia <<std::endl;
12    } while (!plik.eof());
```

Zapis (ostream)

```
4  int main()
5  {
6     char znak;
7     std::ofstream plik("testowy.txt");
8     do {
9        znak = std::cin.get();
10       plik << znak;
11    } while (znak != '.');
12    return 0;
13 }
```

*Pliki i strumienie

```
1  #include <iostream>
2  #include <fstream>
3
4  int main()
5  {
6  std::string linia;
7  std::ofstream plik;
8  plik.open("testowy.txt",std::ofstream::app);
9  for(int i=0;i<10;++i) {
10 plik << rand()%10 <<std::endl;
11 }
12 std::ifstream plikO;
13 plikO.open("testowy.txt");
14    do {
15       getline(plikO,linia);
16       std::cout << linia <<std::endl;
17    } while (!plikO.eof());
18
19 system("PAUSE");
20 return 0;
21 }
```

http://www.cplusplus.com/reference/iostream/ifstream/

http://www.cplusplus.com/reference/iostream/ofstream/

Odczyt

* **int get();** - zwraca znak ze strumienia (lub EOF jeśli skończył się plik)
* **istream& get(char& c);**
  - wpisuje odczytany ze strumienia znak (koniec pliku należy sprawdzić przez eof()).
* **istream& get(char* buf, int len, char eot = '\n');** - pobiera ze strumienia do `buf' maksymalnie `len' znaków i aż do napotkania znaku `eot' (tablica znaków jest terminowana zerem)
* **istream& getline(char* buf, int len, char eot = '\n');** - jak powyższa get, ale znak `eot' nie jest zapisywany w `buf'
* **istream& read(char* ptr, long len);**
* **istream& read(void* ptr, long len);**
  - odczytują ze strumienia maksymalnie `len' znaków do `ptr'; jest to jak widać funkcja do odczytu danych binarnych, zatem tablica NIE jest terminowana zerem!
* **int peek();** - podgląda następny znak w strumieniu (strumień nie ulega zmianie)
* **istream& unget();**- wstawia znak z powrotem do strumienia ☺

*Zadanie domowe lub ćwiczenia

1. Napisz program liczący pole trapezów odczytywanych z pliku tekstowego. W każdej linijce pliku tekstowego znajduje się opis jednej figury (a b h). Oblicz pole i zapisz wyniki do nowego pliku tekstowego
2. Zaimplementuj Szyfr Cezara: kodowanie i dekodowanie (http://pl.wikipedia.org/wiki/Szyfr_Cezara) wraz z obsługą plików (szyfrogram, deszyfrogram). Wymyśl sobie własną technikę zapisu.
3. Odczytaj tekst wpisywany przez użytkownika a następnie wypisz statystykę analizy częstości występowania poszczególnych liter (od ‘a’ do ‘z’) zamieniając wielkie litery na małe.
4. Przerób powyższe, aby tekst odczytywany był z pliku tekstowego, a wyniki analizy – zapisywane do oddzielnego pliku.
5. Wypełnij tablicę 10 x 10 tabliczką mnożenia… ale w kodzie trójkowym.
6. Poczytaj o flagach modyfikujących format wyjścia strumienia.