

Tablice tekstowe – zastosowania

Na poprzednich zajęciach poznaliśmy podstawowe wiadomości dotyczące użytkowania tablic w języku C++. Na bieżących zajęciach zajmiemy się prostymi przykładami zastosowań tablic tekstowych. Na początek rozważmy następujący listing (plik *wspak.cpp*)

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

void wspak(char napis[]);

int main()
{
    char tekst[]="\"To jest przykładowy tekst\"";

    cout << "Program dokona wypisania wspak podanego napisu" << endl << endl;
    cout << "Napis przed modyfikacją: " << tekst << endl;
    wspak(tekst);
    cout << "Napis po modyfikacji: " << tekst << endl;
}

void wspak(char napis[])
{
    unsigned long dlugosc = strlen(napis), indeks;
    char bufor;

    for(indeks=0; indeks<dlugosc/2; indeks++)
    {
        bufor = napis[indeks];
        napis[indeks] = napis[dlugosc-indeks-1];
        napis[dlugosc-indeks-1] = bufor;
    }
}
```

Program w trakcie pracy wyświetli następujący tekst:

Program dokona wypisania wspak podanego napisu

Napis przed modyfikacją: "To jest przykładowy tekst"

Napis po modyfikacji: "tsket ywodalkyzrp tsej oT"

Jak widać, program wypisuje napis, stanowiący zawartość tablicy *tekst*, wprost (w pierwotnej kolejności znaków) i *wspak*. Funkcją roboczą programu jest funkcja *wspak()*, odwracająca kolejność znaków w tablicy tekstowej, będącej jej argumentem. W treści funkcji pojawia się wyrażenie:

```
unsigned long dlugosc = strlen(napis);
```

Jest to definicja zmiennej *dlugosc*, która zostaje zainicjalizowana wartością, zwróconą przez funkcję *strlen()*, wywołaną z argumentem *napis*. Jak pamiętamy, tablica tekstowa zawiera wszystkie znaki wprowadzonego do niej napisu, oraz znak o kodzie 0, kończący tekst, czyli *NULL*. Funkcja z biblioteki standardowej o nazwie *strlen* (ang. *string length* – długość łańcucha) zwraca ilość znaków, zawartych w łańcuchu tekstowym („łańcuch tekstowy” albo „string” to tradycyjna nazwa tekstu, zakończony znakiem *NULL*) stanowiącym treść napisu, nie wliczając końącego napis znaku *NULL*. Aby móc skorzystać z funkcji *strlen* należy dołączyć do programu plik biblioteczny *cstring*. Zadanie odwrócenia tekstu w tablicy zrealizowane jest w pętli i polega na wykonaniu *dlugosc/2* zamian miejscami par znaków w napisie, gdzie operator */* oznacza dzielenie całkowitoliczbowe (całkowity iloraz plus ewentualna reszta z dzielenia). Co jednak się stanie, gdy „długość” jest liczbą nieparzystą? Nie stanowi to problemu, ponieważ w sytuacji, gdy mamy do czynienia z napisem o nieparzystej długości, środkowy znak nie zmienia swojej pozycji, a pozostałe

znaki stanowią już zbiór o parzystej liczbie elementów. Jak należy dokonywać zamian znaków miejscami? Zamieniamy znak n -ty w kolejności z pierwszej połowy tablicy (mniejsze indeksy) ze znakiem n -tym, licząc od końca tekstu. Operacja ta musi pozostawić na swoim miejscu znak *NULL*, bo nie zmienia się długość łańcucha tekstowego przy zmianie kolejności znaków w napisie. Procedura zamiany pary znaków miejscami polega na zapamiętaniu jednego ze znaków z pary, wpisaniu na pozycję znaku zapamiętanego drugiego ze znaków, a następnie wpisaniu uprzednio zapamiętanego znaku w miejsce drugiego ze znaków w parze.

W drugim programie dokonamy ciekawej rzeczy: obliczeń na zmiennych znakowych. Przeanalizujemy drugi program (plik *duze.cpp*)

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

void duze(char napis[]);

int main()
{
char tekst[]="\"To jest przykładowy tekst z cyframi 123\"";

    cout << "Program dokona zamiany na duze litery dla podanego napisu" << endl << endl;
    cout << "Napis przed modyfikacja: " << tekst << endl;
    duze(tekst);
    cout << "Napis po modyfikacji: " << tekst << endl;
}

void duze(char napis[])
{
unsigned long indeks=0;
char znak, roznica = 'A'-'a';

    while(znak=napis[indeks])
    {
        if( (znak>='a') && (znak<='z') ) napis[indeks]+=roznica;
        indeks++;
    }
}
```

Program w trakcie pracy wyświetli następujący tekst:

Program dokona zamiany na duze litery dla podanego napisu

Napis przed modyfikacja: "To jest przykładowy tekst z cyframi 123"

Napis po modyfikacji: "TO JEST PRZYKŁADOWY TEKST Z CYFRAMI 123"

Funkcja *duze* dokonuje w napisie zamiany wszystkich liter z małych na duże. Kod znaków (tzw. *ASCII*) najczęściej obecnie używany w komputerach jest tak skonstruowany, że litery alfabetu łacińskiego (niniejsze rozważania nie uwzględniają tzw. „polskich liter”, czyli znaków diakrytycznych np. ą, ś, ń itd.) są w nim umieszczone w kolejności alfabetycznej, przy czym stanowią zwarte bloki: małych oraz dużych liter. Kody poszczególnych znaków np. litery 'a', są niewielkimi liczbami całkowitymi, często ośmiobitowymi (oznacza to potencjalną możliwość zakodowania 256 znaków). Co należy zrobić, aby w tekście zamienić małe litery na duże? Należy przejrzeć tekst znak po znaku; w wypadku natrafienia na małą literę (znak o kodzie z przedziału kodu litery 'a' do kodu litery 'z'), należy dokonać zamianę danej litery na dużą. Dokonanie zamiany litery z małej na dużą polega na dodaniu różnicy kodów wybranej litery dużej i małej (np. 'A' oraz 'a'). Duże litery alfabetu mają kody z przedziału: 'A' – 65 do 'Z' – 90, zaś małe litery: 'a' – 97 do 'z' – 122. Różnica pomiędzy wartością kodu dowolnej dużej litery względem małej wynosi 'A' – 'a' = -32 i o tyle należy zmienić wartość kodu dowolnej małej litery, aby przekształcić ją w dużą literę.