

Tablice znakowe

Tablice znakowe służą do przechowywania stringów (łańcuchów znakowych).

Język C++ daje możliwość reprezentowania stringów na dwa sposoby. Tablice znakowe (tzw. C stringi) są jednym z nich. Jest to sposób reprezentacji stringów odziedziczony po języku C.

Każda z komórek tablicy przechowuje pojedynczy znak.

Po ostatnim znaku powinien być umieszczony znak pusty null ('\\0 '). Znak ten jest wykorzystywany do określenia końca łańcucha (wykorzystywany przez funkcje działające na stringach

```
//tablice znakowe
char myString[] = "Hello"; // inicjalizacja przez literał łańcuchowy
char myString1[] = { "Hello" };
char myString2[] = { 'H', 'e', 'l', 'l', 'o', '\\0' }; //inicjalizacja zbiorcza
char myString3[6] = { 'H', 'e', 'l', 'l', 'o' }; //inicjalizacja zbiorcza

//wyświetlenie
printf_s(myString);
```

Działania na tablicy znakowej:

Funkcja strlen — wyznaczanie długości łańcucha

```
char myString[]="Język C";

printf_s("Liczba zankow w lancuchu %s wynosi: %d ", myString, strlen(myString));

char myString[]="Język C";
int len = 0;
while (myString[len] != '\\0')
{
    len++;
}
printf_s("Liczba zankow w lancuchu wynosi: %d ", len);

char myString[] = "Język C";
int len = 0;
for ( int i = 0; i < myString[i]; i++)
{
    len++;
}
printf_s("Liczba zankow w lancuchu wynosi: %d ", len);
```

Funkcje strupr i strlwr — przykładowe realizacje

```
char a[] = "ALA";
char b[] = "ala";

printf_s("%s", strlwr(a));
printf_s("\\n%s",strupr(b));
```

Funkcja strcpy — koncepcja kopiowania napisów

```

char str1[20]= "jezyk C";
char str2[20];

//copying the value of str2 to the string str1
strcpy(str2, str1);
printf("String str1: %s\n", str1);
printf("String str2: %s\n", str2);

```

Funkcja strcat

Funkcja strcat dołącza zawartość tablicy znakowej s1 do tablicy s2. Kopiowaniu podlegają wszystkie znaki łańcucha s1 (aż do \0), zakłada się, że tablica s2 ma rozmiar wystarczający na pomieszczenie kopiowanych znaków.

```

char s1[20];
char s2[20];

strcpy(s1, "Programowanie ");
strcpy(s2, "w jezyku C");
strcat(s1, s2);
//puts(s1);
printf_s("\nwynik:%s ",s1);

```

Wykorzystywane metody w C:

```

strlen - Finds out the length of a string
strlwr - It converts a string to lowercase
strupr - It converts a string to uppercase
strcat - It appends one string at the end of another
strncat - It appends first n characters of a string at the end of another.
strcpy - Use it for Copying a string into another
strncpy - It copies first n characters of one string into another
strcmp - It compares two strings
strncmp - It compares first n characters of two strings
strcmpi - It compares two strings without regard to case ("i" denotes that this function ignores case)
stricmp - It compares two strings without regard to case (identical to strcmpi)
strnicmp - It compares first n characters of two strings, Its not case sensitive
strdup - Used for Duplicating a string
strchr - Finds out first occurrence of a given character in a string
strrchr - Finds out last occurrence of a given character in a string
strstr - Finds first occurrence of a given string in another string
strset - It sets all characters of string to a given character
strnset - It sets first n characters of a string to a given character
strrev - It Reverses a string

```

Metoda get ()

```

char s1[20];

printf_s("Podaj tekst: ");
gets(s1);

printf_s("\nPodany tekst: %s",s1);

```

Zadania do samodzielnego rozwiązania

Zadanie 1. Napisz program, w którym zaprezentujesz użycie obiektu klasy `std::string` w celu inicjalizacji, przechowywania danych wejściowych użytkownika oraz kopiowania, łączenia i określenia długości ciągu tekstowego.

Zadanie 2. W języku C każdemu znakowi alfanumerycznemu odpowiada kod liczbowy nazywany kodem ASCII. Napisz program, który wypisze „fragment” tablicy ASCII, czyli wartości kodów znaków od ‘a’ do ‘z’, od ‘A’ do ‘Z’ oraz od ‘0’ do ‘9’

Zadanie 3. Napisz program sprawdzający ile jest spółgłosek i samogłosek we wprowadzonym tekście.

Zadanie 4. Napisz program, który wypisuje strumień wejściowy w odwrotnej kolejności, zamieniając małe litery na duże. Przykład:

Test

TEST

Zadanie 5. Napisz program, który wczytuje jedną dodatnią liczbę całkowitą i wypisuje słowami jej cyfry. Przykład:

0 -> zero

928 -> dziewięć dwa osiem.

Zadanie 6.

Mamy dwa ciągi znaków: `s1="abcdefghijklmnopqrstuvwxyz"`, `s2="zyxwvutsrqponmlkjihgfedcba"`. Napisz program, który każdej literze ciągu wejściowego znalezionej na pozycji `n` w ciągu `s1` przypisuje odpowiadającą jej literę z ciągu `s2` na pozycji `n`, natomiast nie znalezione znaki wypisuje bez zmian. Przykład:

Podaj ciąg znaków:

ala ma kota

zoz nz plgz

Podaj ciąg znaków:

zoz nz plgz

ala ma kota

Wskazówki:

Przy rozwiązywaniu zadania można skorzystać z funkcji `strchr()` zdefiniowanej w pliku nagłówkowym `string.h`. Funkcja ta wyszukuje zadany znak w zadanym napisie. Przykład użycia:

```
#include <stdio.h>
#include <string.h>
int main()
{
    char str[] = "To jest prosty string";
    char* pch;
    printf("Szukam pierwszego znaku 's' w napisie \"%s\"...\n", str);
```

```

    pch = strchr(str, 's');
    printf("Znaleziono na pozycji %d\n", pch - str + 1);
    return 0;
}

```

Można też potraktować zmienne s1 i s2 jako tablice i wyszukiwać w nich znaki z wczytanego napisu.

Pseudokod kodowania:

```

Zmienna pomocnicza napisZakodowany[]=""
Zmienna pomocnicza flagaSukcesuKodowania
Wczytaj Napis
for i=0 to dlugoscNapisu          ///dlugoscNapisu wyznaczamy funkcją
strlen()
    flagaSukcesuKodowania=0
    for j=0 to dlugoscS1          //to jest dlugosc tablicy s1
        jeżeli Napis[i] jest równy s1[j] to
            napisZakodowany[i]=s2[j]
            flagaSukcesuKodowania=1
            break
    next j
    jeżeli flagaSukcesuKodowania jest równa 0 to //nie udało się zakodować
znaku
        napisZakodowany[i]=Napis[i]
next i
Wypisz napisZakodowany

```