Minikurs języka C
Lista 4
24 III 2021
grupa PWit

W tym zadaniu zaimplementujesz kilka standardowych operacji na napisach (łańcuchach znaków) w języku C. Dla przypomnienia:

- Znak ma typ char, a stałe znakowe pisze się w apostrofach, np. instrukcja char znak = ’c’; deklaruje zmienną znak przypisując jej wartość początkową ’c’.
- Napis to ciąg znaków zakończonych znakiem końca napisu o kodzie ASCII 0, czyli znakiem ’\0’.
- Stała napisowa to ciąg znaków ograniczonych znakami ", np. stałą napisową jest "Nietoperz".
- Stała napisowa jest napisem, zatem występuje w niej niewidoczny znak końca napisu. Stąd np. sizeof("Nietoperz") jest równe 10, a nie 9.
- Zmienna napisowa to a) tablica znaków lub b) wskaźnik do bloku pamięci zawierającego ciąg znaków. W obydwu wypadkach końcowym znakiem w tablicy/bloku musi być znak końca napisu. Przykłady: a) char napis1[10] = "Nietoperz", b) char *napis2 = "Nietoperz".
- Zmienne napisowe można modyfikować, o ile nie ma ku temu przeciwwskazań. Dla przykładu, instrukcja napis1[0] = ’n’ sprawi, że napis zapamiętany w tablicy napis1 zmieni się na "nietoperz". Instrukcja napis2[0] = ’n’ może (ale nie musi – to zależy od środowiska wykonania) wygenerować błąd czasu wykonania. A to dlatego, że wskaźnik napis2 może wskazywać na niemodyfikowalną pamięć.
- Zmienną napisową będącą wskaźnikiem można traktować jak każdy inny wskaźnik, np. przypisywać jej nowoprzydzieloną pamięć za pomocą malloc lub za pomocą przypisania adresu już istniejącego bloku w pamięci. Oto przykłady: char *napis3 = (char*) malloc(10) oraz char *napis4 = &napis1. Te dwa wskaźniki wskazują na modyfikowalną pamięć, zatem poprawne są instrukcje napis3[5] = ’n’ czy napis4[0] = ’n’.
- Napis można wypisać na ekran przy pomocy funkcji prinf (np. printf("%s", napis1)) oraz wczytać z klawiatury przy pomocy funkcji scanf (np. scanf("%s", napis1)). W przypadku wczytywania z klawiatury musisz uważać na liczbę wczytanych znaków!
- Tego, czy dwie zmienne napisowe zawierają ten sam napis nie da się stwierdzić przy pomocy operatora == (wyrażenie napis1 == napis2 nie zadziała). Napisy trzeba porównywać znak po znaku. Podobnie, napisów nie da się przypisywać za pomocą operatora przypisania =. Trzeba je kopiować znak po znaku. Nie ma również możliwości przypisania czy wyodrębnienia fragmentu napisu (analogu Pythonowego operatora [:]).

## Zadanie 1

Mając powyższe na uwadze, zaimplementuj dane funkcje zgodnie ze specyfikacją podaną na stronie http://www.cplusplus.com/reference/cstring/. W rozwiązaniu nie można używać żadnych funkcji z nagłówka `string.h`. W operacjach na łańcuchach znaków używaj wyłącznie **składni wskaźnikowej** (arytmetyka na wskaźnikach, porównywanie ich oraz

operacje dereferencji). Używanie **składni tablicowej** (operator indeksowania [.]) jest **zabronione**.

```
char * strcpy ( char * destination, const char * source );
char * strcat ( char * destination, const char * source );
```

Możesz założyć, że wskaźniki `destination` podawane jako argumenty w powyższych funkcjach zawsze wskazują na bloki pamięci o wystarczającej długości. Następnie napisz program, który testuje powyższe funkcje, (np. pobierając dane z wejścia). Przekonaj siebie samego i sprawdzającego te zadanie, że funkcje są poprawnie zaimplementowane dobierając ``złośliwe'' przypadki testowe -- wypisz te przypadki w komentarzu.

## Zadanie 2

Zaimplementuj poniższe funkcje zgodnie ze specyfikacją z zadania 1.

```
int strcmp ( const char * str1, const char * str2 );
int strncmp ( const char * str1, const char * str2, size_t num );
```

dla chętnych (bez punktów):

```
const char * strchr ( const char * str, int character );
const char * strstr ( const char * str1, const char * str2 );
```

## Zadanie 3

Zadanie sprawdzane automatycznie. Pojawi się w serwisie SKOS.