

Lista zadań nr 2

łańcuchy znaków

Zadanie 1

Napisz funkcję posiadającą dwa parametry - pierwszy typu `char*`, drugi typu `char`. Funkcja poszukuje w łańcuchu miejsca pierwszego wystąpienia znaku. W przypadku znalezienia znaku, funkcja ma zwrócić wskaźnik do niego, w przeciwnym wypadku wskaźnik pusty (w podobny sposób działa funkcja `strchr()`). Wykorzystaj funkcję w przykładowym programie.

Zadanie 2

Napisz funkcję posiadającą dwa parametry będące wskaźnikami typu `char*` (podobnie jak w funkcji `strcpy()`). Funkcja powinna zamienić w tekście znajdującym się w tablicy podanej jako drugi argument wszystkie znaki odstępu na znaki podkreślenia i umieścić go w tablicy podanej jako pierwszy argument (zawartość tablicy podanej jako drugi argument ma pozostać niezmienną). Kody znaku odstępu i podkreślenia oznaczane są przez `' '` i `'_'`. Wykorzystaj funkcję w przykładowym programie.

Zadanie 3

Napisz funkcję `void merger(char *s3, const char *s1, const char *s2);` umieszczającą w tablicy wskazywanej przez `s3` najpierw napis wskazywany przez `s1` (z pominięciem kończącego go `'\0'`), a następnie napis wskazywany przez `s2`. W programie pobrać z wejścia dwa łańcuchy `str1` oraz `str2`, które będą argumentami funkcji `merger()`. Utwórz dynamiczną tablicę `str3` o odpowiednim rozmiarze, tak aby można było scalić do niej oba łańcuchy wywołując funkcję `merger()` dla tych argumentów.

Zadanie 4

Napisać funkcję zamieniającą każdą małą literę łańcucha na wielką literę. Znaki, które nie są literami należy pozostawić niezmiennymi.

Zadanie 5

Napisz funkcję zamieniającą każdą literę łańcucha na literę znajdującą się o trzy pozycje dalej w alfabecie (osobno litery małe i wielkie). Na przykład `a` należy zastąpić przez `d`, `b` przez `e`, `A` przez `D`. Trzy ostatnie litery (`x`, `y`, `z`) zastąpić przez kolejne początkowe litery (`a`, `b`, `c`). Znaki, które nie są literami należy pozostawić niezmiennymi.

Zadanie 6

Napisać funkcję obliczającą liczbę wyrazów w napisie podanym jako argument. Przyjąć, że wyraz to ciąg kolejno następujących po sobie znaków złożony z samych liter. Można skorzystać z funkcji bibliotecznej `int isalpha(int c)`, która zwraca wartość różną od zera, gdy jej argument jest kodem wielkiej lub małej litery.

Zadanie 7

Napisać funkcję, która sprawdza, czy przekazywany do niej napis jest palindromem. Wykorzystać funkcję w przykładowym programie.

Zadanie 8

Napisz i przetestuj funkcję, która odwraca kolejność znaków w łańcuchu będącym jej argumentem.

Zadanie 9

Napisz program, który czyta ze standardowego wejścia kolejne linie tekstu wprowadzane przez użytkownika, aż do linii o treści „stop”. Program powinien wypisać najdłuższą linię, liczbę jej znaków i numer.

Zadanie 10

Utwórz program, który pobiera od użytkownika tekst i pewną frazę. Wykorzystując funkcję `strstr()`, program powinien wyszukać pierwsze wystąpienie tej frazy w podanym wierszu tekstu, a następnie znalezione położenie przypisać do zmiennej `search_ptr` typu `char*`. Jeżeli szukana fraza zostanie znaleziona, należy wyświetlić pozostałą część wiersza, począwszy od znalezionej frazy. Następnie funkcja `strstr()` ma zostać użyta do znalezienia kolejnych wystąpień frazy w wierszu tekstu. Jeżeli zostanie znalezione drugie (i kolejne) wystąpienie szukanej frazy, należy wyświetlić pozostałą część tekstu począwszy od znalezionej drugiego (kolejnych) wystąpienia szukanej frazy. Wskazówka: drugie wywołanie funkcji `strstr()` powinno zawierać pierwszy argument w postaci `search_ptr + 1`.

Zadanie 11

Utwórz program wykorzystujący generator liczb pseudolosowych do utworzenia zdań. Program powinien używać czterech tablic (`article`, `noun`, `verb` i `preposition`) wskaźników do `char`. Działanie programu ma polegać na utworzeniu zdania przez losowy wybór słowa z każdej tablicy w podanej kolejności: `article`, `noun`, `verb`, `preposition`, `article` i `noun`. Po pobraniu każdego słowa ma zostać ono połączone z poprzednimi w tablicy wystarczająco dużej do przechowywania całego zdania (dokonaj dynamicznej alokacji pamięci dla tej tablicy). Poszczególne słowa mają być rozdzielone spacjami.

W tablicy `article` mogą być umieszczone elementy takie jak "ten", "ta", "to" itd. W tablicy `noun` mogą być umieszczone elementy takie jak "chłopak", "dziewczyna", "pies", "miasto", "samochód" itd. W tablicy `verb` mogą być umieszczone elementy takie jak "prowadził",

"skoczył", "uciekł", "szedł", "przeskoczył" itd. Z kolei w tablicy `preposition` mogą być umieszczone elementy takie jak "do", "z", "na", "nad", "pod" itd. Program powinien wygenerować 20 takich „zdań”.

Zadania dodatkowe

Zadanie 1

Napisz i przetestuj funkcję, która jako argument pobiera łańcuch i usuwa z niego znaki odstępu ' '. Funkcja nie powinna wykorzystywać dodatkowego bloku pamięci do wykonania tej operacji. Można założyć, że słowa w łańcuchu rozdzielone są pojedynczym znakiem odstępu. Wykorzystaj funkcję w przykładowym programie.

Zadanie 2

Napisz funkcję posiadającą dwa parametry będące wskaźnikami typu `char*`. Funkcja powinna usuwać w tekście znajdującym się w tablicy podanej jako pierwszy argument wszystkie wystąpienia łańcucha będącego drugim argumentem. Wykorzystaj funkcję w przykładowym programie.

Zadanie 3

Utwórz program pobierający ciąg tekstowy numeru telefonu w postaci (555) 555-5555. Program powinien używać funkcji `strtok()` do wyodrębniania tokenów w postaci numeru kierunkowego, pierwszych trzech cyfr numeru telefonu i ostatnich czterech cyfr numeru telefonu. Następnie siedem cyfr tworzących numer telefonu ma zostać połączonych w jeden ciąg tekstowy. Program powinien skonwertować na wartość typu `int` ciąg tekstowy numeru kierunkowego oraz na wartość typu `long int` ciąg tekstowy numeru telefonu (można wykorzystać funkcje biblioteczne). W wyniku działania programu mają zostać wyświetlone numer kierunkowy i numer telefonu (jako liczby)

Zadanie 4

Utwórz program, który pobiera jeden wiersz tekstu (słowa w tekście rozdzielone są spacjami), tokenizuje go za pomocą funkcji `strtok()` oraz wyświetla tokeny w odwrotnej kolejności. Wykorzystaj tablicę wskaźników do typu `char` do „przechowania” wydzielonych tokenów.