

# Zadania z programowania proceduralnego w języku C do wykładu 8

dr hab. Andrzej Zbrzezny, prof. nadzw. AJD

## C-napisy

W poniższych zadaniach nie używaj funkcji z pliku nagłówkowego `string.h`. Przetestuj wszystkie napisane funkcje. Przykładowo, aby przetestować funkcję `strcat`, zadeklaruj i zainicjalizuj napisami dwie zmienne (powiedzmy `a` oraz `b`) typu `char []`, zaalokuj *odpowiednią* liczbę bajtów pamięci dla tablicy znaków `p`, wykonaj ewentualnie przypisanie `*p = '\0'` (zastanów się w zależności od funkcji użytej do alokacji pamięci czy i dlaczego jest ono potrzebne), a następnie wywołaj funkcję `strcat` dwa razy; albo tak:

```
strcat(p, a.c); strcat(p, b.c);
```

albo tak:

```
strcat(strcat(p, a.), b.c);
```

Przed rozwiązaniem poniższych zadań zapoznaj się dokładnie z podanym na końcu przykładowym rozwiązaniem.

1. Napisz tablicowe wersje poniższych funkcji bibliotecznych podanych na wykładzie (wersja tablicowa używa operatora indeksowania a nie używa operatora dereferencji).

(a) `char* strcpy(char* t, const char* s)`

(b) `size_t strlen(const char* s)`

(c) `char* strncpy(char* t, const char* s, size_t n)`

(d) `char* strcat(char* t, const char* s)`

(e) `char* strncat(char* t, const char* s, size_t n)`

2. Napisz wskaźnikowe wersje funkcji z poprzedniego zadania (wersja wskaźnikowa używa operatora dereferencji, a nie używa operatora indeksowania).

3. Napisz tablicową wersję funkcji `strend`, która sprawdza czy C-napis `t` występuje na końcu C-napisu `s`.

`bool strend(const char* s, const char* t)`

4. Napisz wskaźnikową wersję funkcji z poprzedniego zadania.

5. Napisz tablicową wersję funkcji `strindex`, która zwraca pozycję wystąpienia C-napisu `t` w C-napisie `s`, a `-1` gdy C-napis `t` nie występuje w C-napisie `s`.

`int strindex(const char* s, const char* t)`

6. Napisz wskaźnikową wersję funkcji z poprzedniego zadania.

### Przykładowe rozwiązanie zadania 1a:

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

size_t strlen(const char *s);
char *strcpy(char *t, const char *s);

int main(void)
{
    char *s = "Oto rzekł Mistrz Programista:\n\n\
    \"Po trzech dniach bez programowania, \
    życie staje się pozbawione sensu.\"\n";
    char *p = calloc(strlen(s) + 1, sizeof(char));
    strcpy(p, s);
    printf("%s\n", p);
    free(p);
    return 0;
}

size_t strlen(const char *s)
{
    size_t len = 0;
    while (*s++) {
        ++len;
    }
    return len;
}

char *strcpy(char *t, const char *s)
{
    char *p = t;
    while ((*t++ = *s++))
        ;
    return p;
}
```