



Универзитет „Св. Кирил и Методиј“ - Скопје  
**ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ  
И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО**

## Аудиториски вежби 9

### Текстуални низи (Стрингови)

Структурно програмирање

# Функции за работа со текстуални низи

`<string.h>`

## Функции за менување на текстуални низи

- `strcpy` - копирање на една текстуална низа во друга
- `strncpy` - копирање на `n` бајти во тесктуална низа, се копираат од `src` или се додаваат `nulls`
- `strcat` - додава една текстуална низа на крајот на друга
- `strncat` - додава `n` бајти од една текстуална низа во друга

## Функции за менување на меморијата

- `memset` - пополнува низа со одреден бајт

## Функции за претворање на низи од знаци во броеви

- `atof` - претвора низа од знаци во децимален број
- `atoi` - претовра низа од знаци во цел број

# Функции за работа со текстуални низи

## <string.h>

### Функции за испитување на текстуални низи

- `strlen` - ја враќа должината на низата
- `strcmp` - споредува две текстуални низи
- `strncmp` - споредува одреден број бајти од две текстуални низи
- `strchr` - го наоѓа првото појавување на даден знак во текстуална низа
- `strrchr` - го наоѓа последното појавување на знак во низа
- `strspn` - го наоѓа првото појавување во текстуална низа на знак кој не е во множество од знаци
- `strcspn` - го наоѓа последното појавување во текстуална низа на знак кој не е во множество од знаци
- `strpbrk` - го наоѓа првото појавување во текстуална низа на знак од множество од знаци
- `strstr` - го наоѓа во текстуална низа првото појавување на подниза
- `strtok` - го наоѓа во текстуална низа следното појавување на токен

# Функции за работа со знаци

<ctype.h>

- `isalnum` - проверува дали даден знак е алфанумерички (буква или број)
- `isalpha` - проверува дали даден знак е буква
- `iscntrl` - проверува дали даден знак е контролен знак
- `isdigit` - проверува дали даден знак е декадна цифра
- `isxdigit` - проверува дали даден знак е хексадецимална цифра
- `isprint` - проверува дали знакот може да се печати
- `ispunct` - проверува дали даден знак е интерпукциски
- `isspace` - проверува дали дали даден знак е празно место
- `islower` - проверува дали даден знак е мала буква
- `isupper` - проверува дали даден знак е голема буква
- `tolower` - претвора голема во мала буква
- `toupper` - претвора мала во голема буква
- `isgraph` - проверува дали даден знак има локална графичка репрезентација

# Задача 1

Да се напише функција што ќе одреди колку пати знак се наоѓа во даден стринг. Знакот за споредување и стрингот се внесуваат од тастатура.

## Пример

За стрингот

`'hello FINKI'`

знакот `'l'` се наоѓа 2 пати

# Задача 1

## Решение

---

```
#include <stdio.h>
#define MAX 100
int count_char(char *str, char c) {
    int total = 0;
    while (*str != '\0') {
        total += (*str == c);
        str++;
    }
    return total;
}
int main() {
    char s[MAX], c;
    gets(s);
    c = getchar();
    printf("%d\n", count_char(s, c));
    return 0;
}
```

---

## Задача 2

Да се напише функција што ќе ја одреди должината на една текстуална низа. Да се понуди и рекурзивно решение.

### Пример

Ако се внесе: “zdravo!”

Треба да врати: 7

## Задача 2

### Решение

---

```
#include <stdio.h>
#define MAX 100

int length(char *s) {
    int i, len = 0;
    for (i = 0; s[i] != '\0'; i++)
        len++;
    return len;
}

int length_r(char *s) {
    if (*s == '\0')
        return 0;
    return 1 + length_r(s + 1);
}

int main() {
    char s[MAX];
    gets(s);
    printf("Length: %d and %d\n", length(s), length_r(s));
    return 0;
}
```

---



## Задача 3

Да се напише програма која ќе врати подниза од зададена текстуална низа определена со позицијата и должината што како параметри се вчитуваат од тастатура. Поднизата започнува од карактерот што се наоѓа на соодветната позиција во текстуалната низа броено од лево.

### Пример

Ако се внесе:

‘banana’, позиција = 3, должина = 4

Треба да се добие: nana

# Задача 3

## Решение

---

```
#include <stdio.h>
#include <string.h>
#define MAX 100

int main() {
    char s[MAX], dest[MAX];
    int position, length;
    gets(s);
    scanf("%d %d", &position, &length);
    if (position <= strlen(s)) {
        strncpy(dest, s + position - 1, length);
        dest[length] = '\0';
        printf("Result: ");
        puts(dest);
    } else
        printf("Invalid input, the read string has only %d characters.\n",
            strlen(s));
    return 0;
}
```

---

## Задача 4

Да се напише функција која ќе одреди дали една текстуална низа е подниза на друга текстуална низа.

### Пример

“face” е подниза на “Please faceAbook”

# Задача 4

## Решение

```
#include <stdio.h>
#include <string.h>
#define MAX 100
int substring(char *s1, char *s2) {
    int i;
    int d1 = strlen(s1);
    int d2 = strlen(s2);
    if (d1 > d2)
        return 0;
    for (i = 0; i < d2 - d1; i++)
        if (strncmp(s1, s2 + i, d1) == 0)
            return 1;
    return 0;
}
int main() {
    char s1[MAX], s2[MAX];
    gets(s1);
    gets(s2);
    if (substring(s1, s2))
        printf("%s is substring of %s\n", s1, s2);
    else
        printf("%s is NOT substring of %s\n", s1, s2);
    return 0;
}
```

## Задача 5

Да се напише функција која ќе провери дали дадена текстуална низа е палиндром. Текстуална низа е палиндром ако се чита исто од лево на десно и од десно на лево.

### Пример зборови палиндроми

dovod  
ana  
kalabalak

*За дома:* Да се напише функција која ќе проверува дали одредена реченица е палиндром. Да се игнорираат празните места, интерпункциските знаци и мали и големи букви при споредбата.

### Пример реченици палиндроми

Jadenje i pienje daj  
A man, a plan, a canal, Panama  
Never odd or even  
Rise to vote sir

# Задача 5

## Решение

```
#include <stdio.h>
#include <string.h>
#define MAX 100
int is_is_palindrome(char *str) {
    int i, len = strlen(str);
    for (i = 0; i < len / 2; i++)
        if (*(str + i) != *(str + len - 1 - i))
            return 0;;
    return 1;
}
// RECURSIVE
int is_pal(char *str, int start, int end) {
    if (start >= end) return 1;
    if (str[start] == str[end])
        return is_pal(str, start + 1, end - 1);
    return 0;
}

int main() {
    char s[MAX];
    gets(s);
    printf("%s ", s);
    if (is_pal(s, 0, strlen(s) - 1))
        printf("is palindrome.");
    else
        printf("is NOT palindrome.");
    return 0;
}
```

## Задача 6

Да се напише функција која за дадена текстуална низа ќе одреди дали е доволно сложена да биде лозинка. Секоја лозинка мора да има барем една буква, барем еден број и барем еден специјален знак.

### Пример

`zdr@v0!` е валидна лозинка.

`zdravo` не е валидна лозинка.

## Задача 6

### Решение

```
#include <stdio.h>
#include <string.h>
#define MAX 100

int is_valid_password(char *str) {
    int letters = 0, digits = 0, spec = 0;
    for (; *str; str++) {
        if (isalpha(*str))
            letters++;
        else if (isdigit(*str))
            digits++;
        else
            spec++;
    }
    return (letters > 0 && digits > 0 && spec > 0);
}

int main() {
    char s[MAX];
    gets(s);
    printf("%s ", s);
    if (is_valid_password(s))
        printf("is valid password.");
    else
        printf("is NOT valid password.");
    return 0;
}
```



## Задача 7

Да се напише функција која во стринг што и се предава како влезен параметар ќе ги промени малите букви во големи и обратно и ќе ги отфрли сите цифри и специјални знаци.

### Пример

Ако се внесе: “0v@ePr1m3R”

Треба да се добие: “VEpRMr”

# Задача 7

## Решение

```
#include <stdio.h>
#include <string.h>
#define MAX 100

void filter(char *str) {
    int i = 0, j = 0;
    while (str[i] != '\0') {
        if (isalpha(str[i])) {
            if (islower(str[i]))
                str[j] = toupper(str[i]);
            else if (isupper(str[i]))
                str[j] = tolower(str[i]);
            else
                str[j] = str[i];
            j++;
        }
        i++;
    }
    str[j] = '\0';
}

int main() {
    char s[MAX];
    gets(s);
    filter(s);
    printf("%s\n", s);
    return 0;
}
```

## Задача 8

Да се напише функција која за дадена текстуална низа ќе ги исфрли празните места на почетокот и крајот на низата.

### Пример

Ако се внесе:

```
“  make trim  ”
```

Треба да се добие: “make trim”

# Задача 8

## Решение

---

```
#include <stdio.h>
#include <string.h>
#define MAX 100

void trim(char *s) {
    char *d = s;
    while (isspace(*s++))
        ;
    s--;
    while (*d++ = *s++)
        ;
    d--;
    while (isspace(*--d))
        *d = 0;
}

int main() {
    char s[MAX];
    gets(s);
    printf("[%s] -> ", s);
    trim(s);
    printf("[%s]", s);
    return 0;
}
```

---

# Материјали и прашања

Предавања, аудиториски вежби, соопштенија  
`courses.finki.ukim.mk`

Изворен код на сите примери и задачи  
`https://github.com/tdelev/SP/tree/master/latex/src`

Прашања и дискусија  
`forum.finki.ukim.mk`