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

Структурно програмирање

Аудиториски вежби 10

Содржина

1. Текстуални низи (стрингови)

1.1. Потсетување од предавања

1.1.1. Функции за работа со текстуални низи од библиотеката `<string.h>`

Функции за менување на текстуални низи:

- `strcpy()` - копирање на една текстуална низа во друга
- `strncpy()` - копирање на n бајти од една текстуална низа во друга
- `strcat()` - надоврзува една текстуална низа на крајот на друга
- `strncat()` - надоврзува n бајти од една текстуална низа на крајот на друга

Функции за менување на меморијата:

- `memset()` - пополнува низа со одреден бајт

Функции за претворање на текстуални низи во броеви:

- `atof()` - претвора текстуална низа во децимален број
- `atoi()` - претвора текстуална низа во цел број

Функции за испитување на текстуални низи:

- `strlen()` - ја враќа должината на дадена текстуална низа
- `strcmp()` - споредува две текстуални низи
- `strncmp()` - споредува n бајти од две текстуални низи
- `strchr()` - го наоѓа првото појавување на даден знак во текстуална низа
- `strrchr()` - го наоѓа последното појавување на даден знак во текстуална низа
- `strspn()` - во текстуална низа го наоѓа првото појавување на знак кој не

припаѓа на специфицирано множество од знаци

- `strcspn()` - во текстуална низа го наоѓа последното појавување на знак кој не припаѓа на специфицирано множество од знаци
- `strpbrk()` - во текстуална низа го наоѓа првото појавување на знак кој припаѓа на специфицирано множество од знаци
- `strstr()` - во текстуална низа го наоѓа првото појавување на дадена подниза

1.1.2. Функции за работа со знаци од библиотеката `<ctype.h>`

Функции за работа со единечен знак:

- `isalnum()` - проверува дали даден знак е алфанумерички (буква или цифра)
- `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 хексадекадна цифра
- `isprint()` - проверува дали даден знак може да се печати
- `ispunct()` - проверува дали даден знак е интерпункциски знак
- `isspace()` - проверува дали даден знак е празно место
- `islower()` - проверува дали даден знак е мала буква
- `isupper()` - проверува дали даден знак е голема буква
- `tolower()` - претвора дадена голема буква во соодветната мала буква
- `toupper()` - претвора дадена мала буква во соодветната голема буква
- `isgraph()` - проверува дали даден знак има локална графичка репрезентација

1.2. Задача 1

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

Пример

За стрингот

`HELLO FINKI`

знакот `L` се појавува 2 пати.

Решение `p10_1.c`

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

1.3. Задача 2

Да се напише функција што ќе ја одредува должината на една текстуална низа.

Да се даде итеративно и рекурзивно решение.

Пример

Ако на функцијата како аргумент и се предаде стрингот

`zdravo!`

тогаш таа треба да врати: `7`

Решение `p10_2.c`

```c
#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("Dolzhina: %d i %d\n", length(s), length_r(s));
    return 0;
}
```

1.4. Задача 3

Да се напише програма која ќе ја отпечати поднизата на дадена текстуална низа (што се внесува од тастатура) определена со позицијата и должината, што како параметри се внесуваат од тастатура. Поднизата започнува од знакот што се наоѓа на соодветната позиција во текстуалната низа, броејќи од лево.

Пример

Ако од тастатура се внесе:

banana

позиција: 3

должина: 4

тогаш програмата треба да отпечати: nana

Решение `p10_3.c`

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

int main() {
    char s[MAX], dest[MAX];
    int pozicija, dolzhina;
    gets(s);
    scanf("%d %d", &pozicija, &dolzhina);
    if (pozicija <= strlen(s)) {
        strncpy(dest, s + pozicija - 1, dolzhina);
        dest[dolzhina] = '\0';
        printf("Rezultat: ");
        puts(dest);
    } else
        printf("Nevaliden vnes, prochitaniot string ima samo %d znaci.\n",
               strlen(s));
    return 0;
}
```

1.5. Задача 4

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

Пример

`face` е подниза на `Please faceAbook`

Решение `p10_4.c`

```c
#include <stdio.h>
#include <string.h>
#define MAX 100
int podstring(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 (podstring(s1, s2))
        printf("%s e podstring na %s\n", s1, s2);
    else
        printf("%s NE e podstring na %s\n", s1, s2);
    return 0;
}
```

1.6. Задача 5

Да се напише функција која ќе проверува дали дадена текстуална низа е палиндром.

Една текстуална низа е **палиндром** ако таа се чита исто од лево на десно и од десно на лево.

Примери за палиндроми

```
dovod
ana
kalabalak
```

Решение `p10_5.c`

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

int main() {
    char s[MAX];
    gets(s);
    printf("%s ", s);
    if (e_pal(s, 0, strlen(s) - 1))
        printf("e palindrom.");
    else
        printf("NE e palindrom.");
    return 0;
}
```

1.6.1. Задача 5-а (за дома)

Да се напише функција која ќе проверува дали дадена реченица е палиндром. При проверката да се игнорираат празните места, интерпункциските знаци, а соодветните мали и големи букви да се сметаат за еднакви (`A == a`, `B == b`, итн.).

Примери за реченици - палиндроми

```
Jadejne i pienje daj!
A man, a plan, a canal, Panama.
Never odd or even.
Rise to vote sir!
```

1.7. Задача 6

Да се напише функција која за дадена текстуална низа ќе одредува дали таа е доволно сложена за да биде лозинка.

Секоја лозинка мора да содржи барем една буква, барем една цифра и барем еден специјален знак.

Пример

`zdr@v0!` е валидна лозинка.

`zdravo` не е валидна лозинка.

Решение `p10_6.c`

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

int e_validna_lozinka(char *str) {
    int bukvi = 0, cifri = 0, spec = 0;
    for (; *str; str++) {
        if (isalpha(*str))
            bukvi++;
        else if (isdigit(*str))
            cifri++;
        else
            spec++;
    }
    return (bukvi > 0 && cifri > 0 && spec > 0);
}

int main() {
    char s[MAX];
    gets(s);
    printf("%s ", s);
    if (e_validna_lozinka(s))
        printf("e validna lozinka.");
    else
        printf("NE e validna lozinka.");
    return 0;
}
```

1.8. 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 отстрани сите цифри

и специјални знаци.

Пример

За низата:

```
0v@ePr1m3R
```

треба да се добие:

```
VEpRMr
```

Решение `p10_7.c`

```
#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]);
            j++;
        }
        i++;
    }
    str[j] = '\0';
}
int main() {
    char s[MAX];
    gets(s);
    filter(s);
    printf("%s\n", s);
    return 0;
}
```

1.9. Задача 8

Да се напише функција која во дадена текстуална низа ќе ги отстранува празните места на почетокот и крајот од низата.

Пример

За низата:

```
"   make trim   "
```

треба да се добие:

```
"make trim"
```

Решение `p10_8.c`

```c
#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;
}
```

2. Изворен код од примери и задачи

https://github.com/finki-mk/SP/

Source code ZIP