

Решени задачи од аудиториски вежби по
Концепти за развој на софтвер

December 8, 2011

Содржина

1	циклуси	24
2	наредба switch	32
3	Вектори (еднодимензионални полиња)	34
4	Матрици (дводимензионални полиња)	37
5		38

Вовед во програмскиот јазик C

- Развиен во лабораториите на Bell во периодот од 1969 од 1973 од страна на Dennis Ritchie
- Еден од најшироко употребуваните јазици за програмирање со општа намена на сите времиња
- Има огромно влијание во создавањето на многу други јазици за програмирање
 - C++
 - Objective C
 - PHP
 - Java

Синтакса на C

Азбуката е множество на следните дозволени симболи:

a-z, A-Z, 0-9 и ~!@#\$%^&*()-+={}[]:;'"<>?/_

Внимание!

Компајлерот разликува големи и мали букви!

Од азбуката на C се формираат зборови кои може да бидат:

1. Клучни зборови
2. Бројни и симболички константи
3. Идентификатори
4. Стрингови (низи од знаци)
5. Оператори

Синтакса на C

Множество на клучни зборови (32)

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
go	if	static	while

Структура на програма во C

Севкупниот изворен код кој се пишува во програмскиот јазик C е организиран во функции

Програма во C

```
int main() {  
    deklaracija_na_promenlivi;  
    programski_naredbi;  
}
```

Програма во Паскал

```
Program ime_na_programata;  
var deklaracija_na_promenlivi;  
begin  
    programski_naredbi;  
end.
```

Функции во C

main

Главна функција во C

()

Во мали загради се примаат влезните аргументи

int

Видот на податокот кој се враќа како резултат стои пред името на функцијата

{}

Телото на функцијата започнува со **{**, а завршува со **}**

;

Сите наредби се одделуваат меѓусебно со **;**

Употреба на коментари

- За дополнително до објаснување или документирање на изворниот код се користат коментари
- Во C постојат два видови на коментари
 1. коментари во еден ред
 2. коментари во повеќе редови

1. Коментар во еден ред

```
\\ komentar vo eden red
```

2. Коментар во повеќе редови

```
/* Komentar  
   vo povekje redovi */
```

Примери

Пример 1

```
#include <stdio.h>  
// glavna funckija  
int main() {  
    /* funckija za pecatenje na ekran */  
    printf("Dobredojdovte na FINKI!\n");  
    return 0;  
}
```

Структура на програма во C (проширена)

INCLUDE секција содржи **#include** изрази за вклучување на надворешни библиотеки, односно користење веќе дефинирани надворешни функции
DEFINE секција содржи **#define** изрази за декларирање на константи и податочни типови
... дефинирање на глобални променливи и функции
int main() главна функција

Претпроцесор

- Во C преведувањето (компајлирањето) на програмите го извршуваат:
 - претпроцесорот
 - компајлерот
- Претпроцесорот се управува со помош на т.н. директиви
 - Секоја директива започнува со **#**

Датотека со заглавја

- Една примена на претпроцесорски наредби е вклучување на „датотека со заглавија“ (анг. header file)
 - Се користи за декларација на функции и променливи на одредена предефинирана библиотека

- Корисниците ја вклучуваат „датотеката за заглавија“ со цел да ги користат функциите и надворешните променливи
- Вклучување се врши со претпроцесорската директива **#include**
 - Наредбата **#include** предизвикува копија од дадена датотека да се вклучи на местото каде што е испишана директивата

Форми на include

- Има две форми на оваа директива:
 - датотеката која се вклучува може да биде ставена во наводници (" "),
 - или во аголни загради (<>)

Пример

```
#include <imedatoteka.h>
#include "imedatoteka.h"
```

- Разликата е во локацијата во која препроцесорот ја бара датотеката која треба да ја вклучиме
 - Со аголни загради (се користат за датотеки од стандардните библиотеки)
 - Со наводници (препроцесорот прво ја бара датотеката во истиот директориум каде што се наоѓа C датотеката која треба да се компајлира)

Променливи (variables)

- Променливите се симболички имиња за места во меморијата во кои се чуваат некакви вредности
- Сите променливи пред да се користат треба да се *декларираат*
- Со секое ново сместување на вредност во променливата, старата вредност се брише

Начин	на	декларација	на	променливи:
Вид на променливата		Име на променливата	=	Почетна вредност
;				

Видови на променливи

Видови на променливи во C

Цели броеви	Знаковни	Децимални
int	char	float
short		double
long		

Дефинирање на имиња на променливи

- При именувањето на променливите може да се користат:
 - мали букви од а до z;
 - големи букви од А до Z;
 - цифри од 0 до 9 (не смее да започнува со цифра);
 - знак за подвлекување _ кој се третира како буква (не е препорачливо да започнува со _);

Треба да се внимава!

- најчесто должината на имињата на променливите е до 32 знаци
- С ги разликува малите и големите букви!

Примери

Пример 2

```
#include <stdio.h>

int main() {
    int a, b, c;
    a = 5;
    b = 10;
    c = a + b;
    return 0;
}
```

Константи

- Со помош на константи се означуваат вредности кои не се менуваат во текот на извршувањето на програмата
- Секоја константа припаѓа на некој од видовите на податоци
- Во C постојат неколку типови на константи:

- децимални: 1, -23, 15
- октални: 015, 035, 0205
- хексадецимални: 0x25, 0xA4C
- реални: 3.5F, -2.845F, 1.34e-9
- знаковни: 'a', '_', 'e'
- текстуални: " ", "Koncepti za razvoj na softver"

Одредување на типот на константите

- Одредувањето на типот на променливите е едноставно (се гледа од самата декларација на променливата)
- Константите не се декларираат и нивниот тип се одредува преку начинот на кој се напишани:
 - Броевите кои содржат "." или "e" се **double**: 3.5, 1e-7, -1.29e15
 - За наместо double да се користат **float** константи на крајот се додава "F": 3.5F, 1e-7F
 - За **long double** константи се додава "L": 1.29e15L, 1e-7L
 - Броевите без ".", "e" или "F" се **int**: 1000, -35
 - За **long int** константи се додава "L": 9000000L

Именувани константи (1)

Именуваните константи се креираат со користење на клучниот збор **const**

Пример 3

```
#include <stdio.h>

int main() {
    const long double pi = 3.141592653590L;
    const int denovi_vo_nedelata = 7;
    const nedela = 0; /* po default int */
    denovi_vo_nedelata = 1; /* greshka */
}
```

Именувани константи (2)

Именуваните константи може да се креираат и со користење на претпроцесорот и за нив по правило се користат големи букви

Пример 3

```

#include <stdio.h>
#define PI 3.141592653590L
#define DENOVI_VO_NEDELATA 7
#define NEDELA 7
int main() {
    long double pi = PI;
    int den = NEDELA;
}

```

Оператори

- Операторите се користат за градење на изрази, при што операциите се изведуваат од лево надесно со што се применува правилото на приоритет на операторите во нивното изведување
- Постојат три видови на оператори
 - Аритметички оператори
 - Релациони оператори
 - Логички оператори

Аритметички оператори

Се применуваат на броеви (цели или децимални)

Оператор	Операција
+	Собирање
-	Одземање
*	Множење
/	Делење
%	Делење по модул

Релациони оператори

Се применуваат над било кои споредливи типови на податоци, а резултатот е цел број 0 (неточно) или 1 (точно).

Оператор	Значење
<	Помало
<=	Помало еднакво
>	поголемо
>=	поголемо еднакво
==	еднакво
!=	различно

Логички оператори

Се користат најчесто во комбинација со релационите оператори за формирање на сложени логички изрази, кои повторно враќаат резултат 0 или 1

Оператор	Операција
&&	Логичко И
	Логичко ИЛИ
!	Негација

Дополнителни оператори

- Оператор за доделување =
- Оператори за инкрементирање и декрементирање (++ , -)
- Користење на операторите + и - на унарен начин
 - $X = + Y;$
 - $X = - Y;$
- Двојни оператори
 - Комбинација од оператор за доделување и друг оператор (+ = , - = , * = , / = ,

Оператор за доделување =

- Сите изрази имаат вредност, дури и оние кои содржат =
- Вредноста на таков израз е вредноста на изразот кој се наоѓа на десна страна
- Затоа е можно и доделување од следниот облик:

```
x = (y = 10) * (z = 5);  
x = y = z = 20;
```

Двојни оператори

Оператор +=

```
a += 5; // a = a + 5;  
a += b * c; // a = a + b * c;
```

Оператор -=

`a -= 3; // a = a - 3;`

Оператор `*=`

`a *= 3; // a = a * 3;`

Оператор `/=`

`a /= 3; // a = a / 3;`

Оператор `%=`

`a %= 3; // a = a % 3;`

Работа со променливи и оператори

Пример 4

```
#include <stdio.h>
int main() {
    int a;
    float p;
    p = 1.0 / 2.0; /* p = 0.5 */
    a = 5 / 2;     /* a = 2 */
    p = 1 / 2 + 1 / 8; /* p = 0.5 */
    p = 3.5 / 2.8; /* p = 1.25 */
    a = p; /* a = 1 */
    a = a + 1; /* a = 2; */
    return 0;
}
```

Печатење на стандарден излез

- Во C не постои наредба за печатење на екран
- Се користи готова функција од библиотеката за стандарден влез и излез `stdio.h` (standard input/output) `#include <stdio.h>`
- Функцијата која се употребува е:

`int printf(kontrolna_niza, lista_na_argumenti)`

- Контролната низа содржи било каков текст, ознаки за форматот на печатење на аргументите предводени со `%` или специјални знаци кои започнуваат со `\`.
- Ознаките за форматот на печатење се одредуваат според видот на променливата чија вредноста треба да се испише.

Ознаки за форматот на печатење

Ознака	Објаснување
%d	за цели броеви
%i	за цели броеви
%c	за знаци
%s	за низа од знаци
%e	реален број во технички формат (e)
%E	реален број во технички формат (E)
%d	реален број во децимален формат
%f	реален број во пократкиот од форматите %e и %f
%g	реален број во пократкиот од форматите %E и %f
%u	цел број без предзнак
%o	октален цел број без предзнак
%x	хексадецимален цел број без предзнак (мали букви)
%X	хексадецимален цел број без предзнак (мали букви)
%p	прикажува покажувач
%n	бројот на испишани знаци се доделува на аргументот
%%	испишување на знакот %

Примена на функцијата printf

Пример 5

```
#include <stdio.h>
int main() {
    printf(" e zbor dolg %d bukvi.\n", printf("Makedonija"));
    return 0;
}
```

Задача 1

Да се напише програма која ќе ја пресметува вредноста на математичкиот израз: $x = \frac{3}{2} + (5 - \frac{46x5}{12})$

Решение

```
#include <stdio.h>
int main() {
    float x = 3.0 / 2 + (5 - 46 * 5 / 12.0);
    printf("x = %.2f\n", x);
    return 0;
}
```

Задача 2

Да се напише програма која за зададена вредност на x (при декларација на променливата) ќе го пресмета и отпечати на екран x^2 .

Решение

```
#include <stdio.h>
int main() {
    int x = 7;
    printf("Brojot %d na kvadrat e: %d\n", x, x * x);
    return 0;
}
```

Задача 3

Да се напише програма која за дадени страни на еден триаголник ќе ги отпечати на екран периметарот и квадратот од плоштината (нека се работи со $a=5$, $b=7.5$, $c=10.2$).

Решение

```
#include <stdio.h>
int main() {
    float a = 5;
    float b = 7.5;
    float c = 10.2;
    float L = a + b + c;
    float s = L / 2;
    float P = s * (s - a) * (s - b) * (s - c);
    printf("Perimetarot e: %.2f\n", L);
    printf("Plostinata na kvadrat e: %.2f\n", P);
    return 0;
}
```

Задача 4

Да се напише програма за пресметување на аритметичката средина на броевите 3, 5 и 12.

Решение

```
#include <stdio.h>
int main() {
    int a = 3, b = 5, c = 12;
    float as = a + b + c / 3.0;
    printf("Aritmetickata sredina e: %.2f\n", as);
    return 0;
}
```

Задача 5

Да се напише програма која ќе ги отпечати на екран остатоците при делењето на бројот 19 со 2, 3, 5 и 8.

Решение

```
#include <stdio.h>
int main() {
    int a = 19;
    printf("Ostatok pri delenje na %d so 2: %d\n", a, a % 2);
    printf("Ostatok pri delenje na %d so 3: %d\n", a, a % 3);
    printf("Ostatok pri delenje na %d so 5: %d\n", a, a % 5);
    printf("Ostatok pri delenje na %d so 8: %d\n", a, a % 8);
    return 0;
}
```

Потсетување од предавања

- Оператори

- Аритметички
- Релациски
- Логички
- Излез на податоци `printf()`
- Влез на податоци `scanf()`

Внес на податоци во C Функцијата `scanf`

`int scanf(Контролна_низа_од_знаци, arg1, arg2, ..., argn)`

- Контролната низа од знаци е всушност низа од знаци која ја содржи потребната информација за форматирање
- `arg1, arg2, ..., argn` се аргументите кои ги претставуваат индивидуалните податоци

Употреба на `scanf`

Пример 1

```
#include <stdio.h>
int main() {
    char del;
    int delbroj;
    float cena;
    scanf("%c%d%f", &del, &delbroj, &cena);
    return 0;
}
```

Задача 1

Да се напише програма за пресметување и печатење на плоштината и периметарот на круг. Радиусот на кругот се чита од тастатура како децимален број.

Решение

```
#include <stdio.h>
#define PI 3.1415
int main() {
    float r;
    float P = 0, L = 0;
    printf("Vnesete go radiusot na krugot: ");
    scanf("%f", &r);
    L = 2 * r * PI;
    P = r * r * PI;
    printf("P = %f\n", P);
    printf("L = %f\n", L);
    return 0;
}
```

Задача 2

Да се напише програма која од тастатура ќе прочита два цели броја и ќе ја испечати нивната сума, разлика, производ и остатокот при делењето.

Решение

```
#include <stdio.h>
int main() {
    int a, b;
    printf("a = ");
    scanf("%d", &a);
    printf("b = ");
    scanf("%d", &b);
    printf("a + b = %d\n", a + b);
    printf("a - b = %d\n", a - b);
    printf("a * b = %d\n", a * b);
    printf("a %% b = %d\n", a % b);
    return 0;
}
```

Задача 3

Да се напише програма која чита голема буква од тастатура и ја печати истата како мала буква. Помош: Секој знак се претставува со ASCII број. Пр. 'A' = 65, 'a' = 97

Решение

```
#include <stdio.h>
int main() {
    char c;
    printf("Vnesete golema bukva: ");
    scanf("%c", &c);
    printf("%c malo se pishuva %c\n", c, c + ('a' - 'A'));
    return 0;
}
```

Задача 4

Да се напише програма која чита знак од тастатура и во зависнот од тоа дали е мала или голема буква печати 1 или 0, соодветно. Помош: Користете логички и релациони оператори за тестирање на ASCII вредноста на знакот. **Бонус:** Направете проверка дали знакот е број

Решение

```
#include <stdio.h>
int main() {
    char ch;
    int rez;
    printf("Vnesete znak: ");
    scanf("%c", &ch);
    rez = (ch >= 'a') && (ch <= 'z');
    printf("%d\n", rez);
    return 0;
}
```

Задача 5

Да се напише програма која ќе чита два цели броеви (x, y) од тастура и на компјутерскиот екран ќе го испечати резултатот (z) од следниот израз $z = x++ + -y + (x < y)$ Каква вредност ќе има z за x=1, y=2?

Решение

```
#include <stdio.h>
int main() {
    int x, y, z;
    printf("Vnesete x i y, soodvetno: ");
    scanf("%d%d", &x, &y);
    z = x++ + --y + (x < y);
    printf("z = %d\n", z);
    return 0;
}
```

Задача 6

Нека е дадено: $r = (x < y \ || \ y < z++)$ Каква вредност ќе има r за x=1, y=2, z=3? Каква вредност ќе има z?

Решение

r = 1 z = 3

Нека е дадено: $r = (x > y \ \&\& \ y < z++)$ Каква вредност ќе има r за x=1, y=2, z=3? Каква вредност ќе има z?

Решение

r = 0 z = 3

Задача 7

Нека е дадено:

```
int x, y;
y = scanf("%d", &x);
```

Каква вредност ќе има y за x=5?

Решение

y = 1

Нека е дадено:

```
int x, y, z;
z = scanf("%d%d", &x, &y);
```

Каква вредност ќе има z за x=5, y=6?

Решение

z = 2

Задача 8

Да се напише програма каде од тастатура ќе се внесе цена на производ, а потоа ќе ја испечати неговата цена со пресметан ДДВ. Помош: ДДВ е 18% од почетната цена

Решение

```
#include <stdio.h>
int main() {
    float cena;
    printf("Vnesete ja cenata na proizvodot: ");
    scanf("%f", &cena);
    printf("Vkupnata cena e %.2f\n", cena * 1.18);
    return 0;
}
```

Задача 9

Да се напише програма каде од тастатура ќе се внесе цена на производ, број на рати на кои се исплаќа и камата (каматата е број изразен во проценти од 0 до 100). Програмата треба да го испечати износот на ратата и вкупната сума што ќе се исплати за производот. Помош: Пресметајте ја целата сума, па потоа ратата.

Задача 9 Решение

```
#include <stdio.h>
int main() {
    float cena, kamata, rata, vkupno;
    int brRati;
    printf("Vnesete ja cenata na proizvodot: ");
    scanf("%f", &cena);
    printf("Vnesete go brojot na rati: ");
    scanf("%d", &brRati);
    printf("Vnesete ja kamata: ");
    scanf("%f", &kamata);
    vkupno = cena * (1 + kamata / 100);
    rata = vkupno / brRati;
    printf("Edna rata ke iznesuva: %.3f\n", rata);
    printf("Vkupnata isplatena suma ke bide %.3f\n", vkupno);
    return 0;
}
```

Задача 10

Да се напише програма каде од тастатура ќе се внесе трицифрен цел број. Програмата ќе ја испечати најзначајната и најмалку значајната цифра од бројот Пример: Ако се внесе следниот бројот 795, програмата ќе испечати: Najznasajna cifra e 7, a najmalku znasajna e 5. Помош: Искористете целобројно делење и остаток од делење.

Задача 10 Решение

```
#include <stdio.h>
int main() {
    int broj;
    printf("Vnesete tricifren broj: ");
    scanf("%d", &broj);
    printf("Najznacajna cifra e %d, a najmalku znacajna e %d\n", broj
        / 100, broj % 10);
    return 0;
}
```

Задача 11

Да се напише програма каде од тастатура ќе се внесе датумот на раѓање во формат (ddmmggggg). Програмата на компјутерскиот екран ќе го испечати денот и месецот на раѓање. Пример: Ако се внесе следниот број 18091992, програмата ќе испечати: 18.09 Помош: Искористете целобројно делење и остаток од делење.

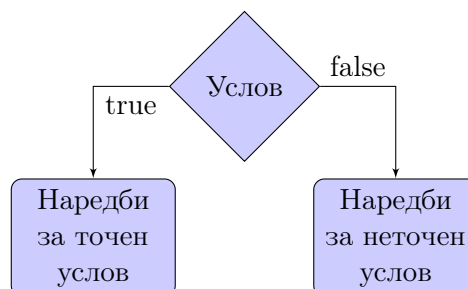
Задача 11 Решение

Решение

```
#include <stdio.h>
int main() {
    int datum;
    int den, mesec;
    printf("Vnesete datum na raganje: ");
    scanf("%d", &datum);
    den = datum / 1000000;
    mesec = (datum / 10000) % 100;
    printf("Vasata data na raganje e %02d.%02d\n", den, mesec);
    return 0;
}
```

Бонус: Задачата може да се реши и со употреба на `scanf("%2d%2d", &den, &mesec)`

Потсетување од предавања



```
if(uslov) {
    naredbi_za_vistinit_uslov;
} else {
    naredbi_za_nevistinit_uslov;
}
```

Употреба на if

Пример 1

```
#include <stdio.h>
int main() {
    int i;
    printf("Vnesete cel broj\n");
    scanf("%d", &i);
    if(i > 0)
        printf("Vnesen e pozitiven broj\n");
    if(i < 0)
        printf("Vnesen e negativen broj\n");
    if(i == 0)
        printf("Vnesena e nula\n");
    return 0;
}
```

Со употреба на if-else

Пример 2

```
#include <stdio.h>
int main() {
    int i;
    printf("Vnesete cel broj\n");
    scanf("%d", &i);
    if(i > 0)
        printf("Vnesen e pozitiven broj\n");
    else if(i < 0)
        printf("Vnesen e negativen broj\n");
    else
        printf("Vnesena e nula\n");
    return 0;
}
```

Едноставни примери

Што ќе отпечати?

Пример 3

```
#include <stdio.h>
int main() {
    int m = 5, n = 10;
    if(m > n)
        ++m;
    ++n;
    printf("m = %d, n = %d\n", m, n);
    return 0;
}
```

m = 5, n = 11

Задача 1

Да се напише програма со која ќе се отпечати максимумот од два броја чији вредности се читаат од тастатура.

Решение

```
#include <stdio.h>
int main() {
    int a, b;
    printf("Vnesete gi vrednostite na a i b: \n");
    scanf("%d %d", &a, &b);
    if(a > b)
        printf("Vrednosta na maksimumot e %d.\n", a);
    else
        printf("Vrednosta na maksimumot e %d.\n", b);
    return 0;
}
```

Задача 2

Да се напише програма која проверува дали дадена година која се вчитува од тастатура е престапна или не и на екран печати соодветна порака. Пример: 1976, 2000, 2004, 2008, 2012...

Решение

```
#include <stdio.h>
int main() {
    int godina;
    printf("Vnesete ja godinata: \n");
    scanf("%d", &godina);
    if((godina % 4 == 0 && godina % 100 != 0) || godina % 400 == 0)
        printf("%d godina e prestopna.\n", godina);
    else
        printf("%d godina ne e prestopna.\n", godina);
    return 0;
}
```

Задача 3

Од тастатура се внесуваат координати на една точка. Да се напише програма со која ќе се испечати на кој квадрант или оска припаѓа внесената точка. Ако станува збор за точка која лежи на координатниот почеток, да се испечати соодветна порака.

Решение 1 дел

```
#include <stdio.h>
int main () {
    float x, y;
    printf("Vnesete kootdinati \n");
    scanf ("%f %f", &x, &y);
    if(x > 0) {
        if(y > 0)
            printf("I Kvadrant.\n");
    }
```

```

else if(y < 0)
    printf("IV kvadrant.\n");
else printf("Poz. x oska.\n");

```

Задача 3 Решение

Решение 2 дел

```

    } else if(x < 0) {
if(y > 0)
    printf("II kvadrant.\n");
else if(y < 0)
    printf("III kvadrant.\n");
else
    printf("Neg. x oska.\n");
} else {
    if(y > 0)
        printf("Poz. y oska.\n");
    else if(y < 0)
        printf("Neg. y oska.\n");
    else
        printf("Koord. pocetok\n");
}
return 0;
}

```

Задача 4

Да се напише програма која за внесен број на поени од испит ќе генерира соодветна оценка според следната табела:

Поени	Оценка
0 - 50	5
51 - 60	6
61 - 70	7
71 - 80	8
81 - 90	9
91 - 100	10

Задача 4 Решение

Решение

```

#include <stdio.h>
int main () {
    int i, oценка = 0;
    printf("Vnesete poeni: \n");
    scanf("%d", &i);
    if(i >= 0 && i <= 50) oценка = 5;
    else if(i > 50 && i <= 60) oценка = 6;
    else if(i > 60 && i <= 70) oценка = 7;
    else if(i > 70 && i <= 80) oценка = 8;
    else if(i > 80 && i <= 90) oценка = 9;
    else if(i > 90 && i <= 100) oценка = 10;
    else printf("Vnesen e pogreshen broj za poenite!!\n");
}

```

```

        if(ocenka)
            printf("Studentot dobil ocena %d.\n", ocenka);
        return 0;
    }

```

Задача 5

Да се промени претходната програма, така што покрај оценките ќе се испечатат и знаците + и – во зависност од вредноста на последната цифра на поените:

последна цифра	печати
1 - 3	-
4 - 7	<prazno mesto>
8 - 0	+

Пример: 81 = 9-, 94 = 10, 68 = 7+. За оценката 5 не треба да се додава + или –, а за оценката 10 не треба да се додава знакот +.

Задача 5 Решение

Решение

```

...
// isto kako od prethodnata zadaca (zadaca 4)
char znak = ' ';
if(ocenka) {
    int p = i % 10;
    if(ocenka != 5) {
        if(p >= 1 && p <= 3) znak = '-';
        else if(ocenka != 10 && (p >= 8 || p == 0))
            znak = '+';
    }
    printf("Studentot dobil ocena %d%c.\n", ocenka, znak);
}
return 0;
}

```

Задача 6

Да се напише програма која ќе претставува едноставен калкулатор. Во програмата се вчитуваат два броја и оператор во формат:

broj1 operator broj2

По извршената операција во зависност од операторот, се печати резултатот во формат:

broj1 operator broj2 = rezultat

Задача 6 Решение

Решение

```
#include <stdio.h>
int main() {
    char op; float br1, br2, rezultat;
    printf("Vnesete dva broja i operator vo format\n");
    printf(" broj1 operator broj2\n");
    scanf("%f %c %f", &br1, &op, &br2);
    if(op == '*') rezultat = br1 * br2;
    else if(op == '+') rezultat = br1 + br2;
    else if(op == '-') rezultat = br1 - br2;
    else if(op == '/') {
        if(br2) rezultat = br1 / br2;
        else {
            printf("Ne se deli so 0!\n");
            return 0;
        }
    } else {
        printf("Nevaliden operator!\n");
        return 0;
    }
    printf("%f %c %f = %f\n", br1, op, br2, rezultat);
    return 0;
}
```

Задача 7

Од тастатура се внесуваат три броја кои не мора да се сортирани. Внесените броеви претставуваат должини на страните на правоаголен триаголник. Да се напише програма која што ќе проверува дали може да се конструира триаголник од дадените должини, при што ако може, треба да се провери дали истиот е правоаголен и да се пресмета неговата плоштина. Во спротивно, треба да се испечатат соодветни пораки.

Задача 7 Решение

Решение

```
#include <stdio.h>
int main() {
    int a, b, c, tmp;
    printf("Vnesi gi dolzinite na stranite: \n");
    scanf("%d %d %d", &a, &b, &c);
    if((a + b <= c) || (a + c <= b) || (b + c <= a))
        printf("Ne moze da se konstruira triagolnik so tie strani.\n");
    else {
        if(a >= b) {
            tmp=a; a = b; b = tmp;
        }
        if(a >= c) {
            tmp = a; a = c; c = tmp;
        }
        if(b >= c) {
            tmp = b; b = c; c = tmp;
        }
    }
}
```

```

        if(c * c == a * a + b * b) {
            printf("Triagolnikot e pravoagolen.\n");
            printf("Plostinata e %7.3f.\n", a * b / 2.0);
        }
        else { printf("Triagolnikot e ne pravoagolen.\n"); }
    }
    return 0;
}

```

1 циклуси

Задача 1a

Да се напише програма за пресметување на сумата на сите парни двоцифрени броеви. Добиената сума се печати на екран.

Решение

```

#include <stdio.h>
int main () {
    int i = 10, suma = 0;
    while (i <= 98) {
        suma = suma + i;
        i+=2;
    }
    printf("Sumata na site parni dvocifreni broevi e %d\n", suma);
    return 0;
}

```

Задача 1б

Да се напише програма за пресметување на сумата на сите непарни двоцифрени броеви. Програмата ја печати сумата на екран во следниот формат: 11 + 13 + 15 + 17 + ... + 97 + 99 = 2475 **Забелешка: Програмата да се направи без користење на наредбата if**

Решение - Верзија 1

```

#include <stdio.h>
int main () {
    int i = 11, suma = 0;
    printf("%d", i);
    suma = i;
    i=i+2;
    while (i <= 99){
        printf(" + %d", i);
        suma = suma + i;
        i+=2;
    }
    printf(" = %d\n", suma);
    return 0;
}

```

Решение - Верзија 2

```

#include <stdio.h>
int main () {
    int i = 11, suma = 0;
    while (i <= 97) {
        printf("%d + ", i);
        suma = suma + i;
        i+=2;
    }
    printf(" %d", i);
    suma = suma + i;
    printf(" = %d\n", suma);
    return 0;
}

```

Задача 2 Решение со употреба на while и do...while

Да се напише програма за пресметување на $y = x^n$ за даден природен број $n, n \geq 1$ и реален број x .

Решение - со употреба на while

```

#include <stdio.h>
int main () {
    int brojac = 0, n;
    float x, y = 1;
    printf("vnesi ja osnovata: ");
    scanf("%f", &x);
    printf("vnesi go eksponentot: ");
    scanf("%d", &n);
    while (brojac < n) {
        y *= x;
        brojac++;
    }
    printf("%f^%d = %f\n", x, n, y);
    return 0;
}

```

Решение - со употреба на do...while

```

#include <stdio.h>
int main () {
    int brojac = 0, n;
    float x, y = 1;
    printf("vnesi ja osnovata: ");
    scanf("%f", &x);
    printf("vnesi go eksponentot: ");
    scanf("%d", &n);
    do {
        y *= x;
        brojac++;
    } while (brojac < n);
    printf("%f^%d = %f\n", x, n, y);
    return 0;
}

```

Задача 2 Решение со употреба на for

Да се напише програма за пресметување на $y = x^n$ за даден природен број $n, n \geq 1$ и реален број x .

Решение со употреба на for

```
#include <stdio.h>
int main () {
    int brojac = 0, n;
    float x, y = 1;
    printf("vnesi ja osnovata: ");
    scanf("%f", &x);
    printf("vnesi go eksponentot: ");
    scanf("%d", &n);
    for(brojac = 1, y = x; brojac < n; brojac++) {
        //for( brojac < n; brojac++) {
            x *= y;
        }
    printf("%f^%d = %f\n", x, n, y);
    return 0;
}
```

Задача 3

Да се напише програма која од n броеви (внесени од тастатура) ќе го определи бројот на броеви што се деливи со 3, при делењето со 3 имаат остаток 1, односно 2. **Забелешка:** Задачата да се реши со **while**, **do...while** и **for**

Решение на задача 3 while

Решение на задачата со употреба на while

```
#include <stdio.h>
int main () {
    int n = 1, i = 0, broj, del, os1, os2;
    del = os1 = os2 = 0;
    printf("Kolku broevi treba da se proveruvaat za delivost so 3?\n");
    ;
    scanf("%d", &n);
    while (i < n) {
        printf("Vnesete broj za proverka: ");
        scanf("%d", &broj);
        if (broj % 3 == 0)
            del++;
        else if (broj % 3 == 1)
            os1++;
        else os2++;
        i++;
    }
    printf("%d broj(a) se delivi so 3.\n", del);
    printf("%d broj(a) imaat ostatok 1, pri delenje so 3.\n", os1);
    printf("%d broj(a) imaat ostatok 2, pri delenje so 3.\n", os2);
    return 0;
}
```

Решение на задача 3 do while

Решение на задачата со употреба на *do...while*

```
#include <stdio.h>
int main () {
    int n = 1, i = 0, broj, del, os1, os2;
    del = os1 = os2 = 0;
    printf("Kolku broevi treba da se proveruvaat za delivost so 3?\n");
    ;
    scanf("%d", &n);
    do {
        printf("Vnesete broj za proverka: ");
        scanf("%d", &broj);
        if (broj % 3 == 0)
            del++;
        else if (broj % 3 == 1)
            os1++;
        else
            os2++;
        i++;
    } while (i < n);
    printf("%d broj(a) se delivi so 3.\n", del);
    printf("%d broj(a) imaat ostatok 1, pri delenje so 3.\n", os1);
    printf("%d broj(a) imaat ostatok 2, pri delenje so 3.\n", os2);
    return 0;
}
```

Решение на задача 3 for

Решение на задачата со употреба на *for*

```
#include <stdio.h>
int main () {
    int n = 1, i = 0, broj, del, os1, os2;
    del = os1 = os2 = 0;
    printf("Kolku broevi treba da se proveruvaat za delivost so 3?\n");
    ;
    scanf("%d", &n);
    for (i = 0; i < n; i++) {
        printf("Vnesete broj za proverka: ");
        scanf("%d", &broj);
        if (broj % 3 == 0)
            del++;
        else if ( broj % 3 == 1)
            os1++;
        else
            os2++;
    }
    printf("%d broj(a) se delivi so 3.\n", del);
    printf("%d broj(a) imaat ostatok 1, pri delenje so 3.\n", os1);
    printf("%d broj(a) imaat ostatok 2, pri delenje so 3.\n", os2);
    return 0;
}
```

Задача 4

Да се напише програма која која на екран ќе ги испечати сите четири-цифрени броеви кај кои збирот на трите најмалку значајни цифри е еднаков со најзначајната цифра.

Пример

4031 ($4=0+3+1$), 5131 ($5=1+3+1$)

Решение на задача 4

Решение

```
#include <stdio.h>
int main() {
    int m, i, n, suma, prva_cifra, cifra;
    i = 1000;
    while (i <= 9999) {
        prva_cifra = i / 1000;
        n = i % 1000;
        suma = 0;
        while (n > 0) {
            cifra = n % 10;
            suma += cifra;
            n /= 10;
        }
        if (suma == prva_cifra) printf("%d\t", i);
        i++;
    }
    return 0;
}
```

Задача 5

Да се напише програма која ќе ги испечати сите броеви од зададен опсег кои исто се читаат и одлево надесно и оддесно налево.

Пример

12345 54321

Решение 5

Решение

```
#include <stdio.h>
int main () {
    int i, odb, dob, pom, prev, cifra;
    printf("Vnesete vrednost za opsegot.\n");
    printf("Od koj broj?\n"); scanf("%d", &odb);
    printf("Do koj broj?\n"); scanf("%d", &dob);
    for (i = odb; i <= dob; i++) {
        pom = i;
        prev = 0;
        while (pom > 0) {
            cifra = pom % 10;
            prev = prev * 10 + cifra;
            pom /= 10;
        }
        if (prev == i) printf("%d\t", i);
    }
    return 0;
}
```

Задача 6

Да се напише програма која од непознат број на цели броеви кои се внесуваат од тастатура ќе го определи бројот со максимална вредност. Програмата завршува ако наместо број се внесе знак што не е цифра.

Решение

```
#include <stdio.h>
int main() {
    int broj, max;
    if (scanf("%d", &max)){
        while(scanf("%d", &broj)){
            if(max < broj){
                max = broj;
            }
        }
        printf("Maksimalniot broj e %d", max);
    } else {
        printf("Treba da vnesete najmalku eden cel broj");
    }
    return 0;
}
```

Задача 7

Да се напише програма која од непознат број на цели броеви кои се внесуваат од тастатура ќе го определи бројот со максимална вредност. Притоа броевите поголеми од 100 не се земаат предвид т.е. се игнорираат. Програмата завршува ако наместо број се внесе знак што не е цифра.

Решение на задача 7

Решение

```
#include <stdio.h>
int main() {
    int broj, max;
    if (scanf("%d", &max)) {
        while(scanf("%d", &broj)) {
            if (broj > 100) continue;
            if(max < broj){
                max = broj;
            }
        }
        printf("Maksimalniot broj e %d", max);
    } else {
        printf("Treba da vnesete najmalku eden cel broj");
    }
    return 0;
}
```

Задача 8

Да се напише програма која од непознат број на цели броеви кои се внесуваат од тастатура ќе ги определи двата броја со најголеми вредности. Програмата завршува ако наместо број се внесе знак што не е цифра.

Пример

Ако се внесат броевите 2 4 7 4 2 1 8 6 9 7 10 3 програмата ќе отпечати 10 и 9.

Решение на задача 8

Решение

```
#include <stdio.h>
int main() {
    int broj, max1, max2, pom;
    if (scanf("%d%d", &max1, &max2) == 2) {
        if (max2 > max1) {
            pom = max1;
            max1 = max2;
            max2 = pom;
        }
        while (scanf("%d", &broj)) {
            if (broj > max1) {
                max2 = max1;
                max1 = broj;
            } else if (broj > max2) {
                max2 = broj;
            }
        }
        printf("Brojot so najgolema vrednost e %d\n", max1);
        printf("Brojot so vтора najgolema vrednost e %d\n", max2);
    } else {
        printf("Treba da vnesete najmalku dva celi broja");
    }
    return 0;
}
```

Задача 9

Да се напише програма која од N цели броеви внесени од тастатура ќе ја определи разликата од сумите на броевите на парни и непарни позиции (според редоследот на внесување). Ако оваа разлика е помала од 10 на екран се печати "Dvete sumi se slicni" а во спротивно на екран се печати "Dvete sumi mnogu se razlikuvaat".

Пример

За броевите внесени од тастатура: 2 4 3 4 2 1 1 6 1 7 suma_neparni_pozicii = 9 suma_parni_pozicii = 22 На екран ќе се испечати: Dvete sumi mnogu se razlikuvaat

Решение на задача 9

Решение

```
#include <stdio.h>
// #include <math.h>
int main() {
    int razlika, i, n = 0, broj = 0;
    int suma_neparni_pozicii = 0, suma_parni_pozicii = 0;
```

```

scanf("%d", &n);
for (i = 1; i <= n; i++){
    scanf("%d", &broj);
    if (i % 2){
        suma_neparni_pozicii += broj;
    } else {
        suma_parni_pozicii += broj;
    }
}
razlika = suma_parni_pozicii - suma_neparni_pozicii;
//printf("razlikata e %d", razlika);
//if(abs(razlika) < 10){
if(razlika < 10 && razlika > -10){
    printf("Dvete sumi se slicni");
} else {
    printf("Dvete sumi mnogu se ralikuvaat");
}
return 0;
}

```

Задача 10

Да се напише програма која од непознат број на цели броеви кои се внесуваат од тастатура ќе ги определи позициите (редните броеви на внесување) на двата последователни броеви кои ја имаат најголемата сума. Програмата завршува ако едно по друго (последователно) се внесат два негативни цели броја.

Решение на задача 10

Решение

```

#include <stdio.h>
int main() {
    int pol_pozicija, pozicija, max_suma, suma, prethoden, sleden;
    scanf("%d%d", &prethoden, &sleden);
    pol_pozicija = pozicija = 2;
    max_suma = suma = prethoden + sleden;
    while(1){
        if (prethoden < 0 && sleden < 0){
            break;
        }
        suma = prethoden + sleden;
        if (suma > max_suma){
            max_suma = suma;
            pol_pozicija = pozicija;
        }
        prethoden = sleden;
        scanf("%d", &sleden);
        pozicija++;
    }
    if(pozicija > 2)
        printf("broevite se naogaat na pozicija %d i %d a nivanata
            suma e %d",
            pol_pozicija - 1, pol_pozicija, max_suma);
    return 0;
}

```

2 наредба switch

Задача 1

Да се напише програма што ќе овозможи претворање на двоцифрените броеви во зборови на следниот начин: За двоцифрениот број 89 на екран ќе се испечати "osum devet".

Решение на задача 1

Решение прв дел

```
#include <stdio.h>
int main() {
    int broj, mala, golema;
    printf("Vnesete dvocifren broj:");
    scanf("%d", &broj);
    mala = broj % 10;
    golema = broj / 10;
    switch (golema) {
        case 0:
            printf("nula ");
            break;
        case 1:
            printf("eden ");
            break;
        case 2:
            printf("dva ");
            break;
        case 3:
            printf("tri ");
            break;
        case 4:
            printf("cetiri ");
            break;
        case 5:
            printf("pet ");
            break;
        case 6:
            printf("sest ");
            break;
        case 7:
            printf("sedum ");
            break;
        case 8:
            printf("osum ");
            break;
        case 9:
            printf("devet ");
            break;
        default:
            break;
    }
}
```

Решение втор дел

```
switch (mala) {
    case 0:
        printf("nula\n");
```



```

        break;
    case 1:
        printf("eden\n");
        break;
    case 2:
        printf("dva\n");
        break;
    case 3:
        printf("tri\n");
        break;
    case 4:
        printf("cetiri\n");
        break;
    case 5:
        printf("pet\n");
        break;
    case 6:
        printf("sest\n");
        break;
    case 7:
        printf("sedum\n");
        break;
    case 8:
        printf("osum\n");
        break;
    case 9:
        printf("devet\n");
        break;
    default:
        break;
}
printf("%d %d\n", golema, mala);
return (0);
}

```

Задача 2

Да се напише програма која ќе претставува едноставен калкулатор. Во програмата се вчитуваат два броја и оператор во формат: broj1 operator broj2 По извршената операција во зависност од операторот, се печати резултатот во формат: broj1 operator broj2 = rezultat

Решение на задача 2

Решение

```

#include <stdio.h>
int main() {
    char op;
    float br1, br2, rez = 0;
    printf("Vnesete dva broja i operator vo format:\n");
    printf("broj1 operator broj2\n");
    scanf("%f %c %f", &br1, &op, &br2);
    switch (op) {
        case '+':
            rez = br1 + br2;
            break;
        case '-':
            rez = br1 - br2;

```

```

        break;
    case '*':
        rez = br1 * br2;
        break;
    case '/':
        if (br2 == 0) {
            printf("Greshka: Delenje so 0\n");
            printf(" operacijata ke se ignorira\n");
        }
        else {
            rez = br1 / br2;
        }
        break;
    default:
        printf("Nepoznat operator %c\n", op);
        break;
}
if(res) printf("Rezultatot od operacijata: %.2f %c %.2f = %f", br1
, op, br2,
rez); return (0);
}

```

3 Вектори (еднодимензионални полиња)

Задача 1

Да се напише програма која за две низи кои се внесуваат од тастатура ќе провери дали дали се еднакви или не. На екран да се испачати резултатот од споредбата. Максимална големина на низите е 100.

Решение 1 дел

```

#include<stdio.h>
#define MAX 100
int main() {
    int n1, n2, element, i;
    int a[MAX], b[MAX];
    printf("Golemina na prvata niza: ");
    scanf("%d", &n1);
    printf("Golemina na vtorata niza: ");
    scanf("%d", &n2);
    if(n1 != n2)
        printf("Nizite ne se ednakvi\n");
}

```

Задача 1 Решение 2 дел

Решение 2 дел

```

    else {
        printf("Vnesi gi elementite od prvata niza: \n");
        for(i = 0; i < n1; ++i) {
            printf("a[%d] = ", i);
            scanf("%d", &a[i]);
        }
        printf("Vnesi gi elementite od vtorata niza: \n");
        for(i = 0; i < n2; ++i) {
            printf("b[%d] = ", i);
        }
    }
}

```

```

        scanf("%d", &b[i]);
    }
    //proverka dali nizite se ednakvi:
    for(i = 0; i < n1; ++i)
        if(a[i] != b[i])
            break;
    if(i == n1)
        printf("Nizite se ednakvi \n");
    else
        printf("Nizite ne se ednakvi \n");
}
return 0;
}

```

Задача 2

Да се напише програма која за низа, чии што елементи се внесуваат од тестатура, ќе го пресмета збирот на парните елементи, збирот на непарните елементи, како и односот помеѓу бројот на парни и непарни елементи. Резултатот да се испечати на екран.

Пример

За низата: 3 2 7 6 2 5 1 На екран ќе се испечати: suma_parni = 8
 suma_neparni = 16 odnos = 0.75

Задача 2 Решение

Решение

```

#include <stdio.h>
#define MAX 100
int main() {
    int i, n, a[MAX], brNep = 0, brPar = 0, sumNep = 0, sumPar = 0;
    printf("Vnesi ja goleminata na nizata: \n");
    scanf("%d", &n);
    printf("Vnesi gi elementite od nizata: \n");
    for(i = 0; i < n; ++i)
        scanf("%d", &a[i]);
    for(i = 0; i < n; ++i) {
        if(a[i] % 2) {
            brNep++;
            sumNep += a[i];
        } else {
            brPar++;
            sumPar += a[i];
        }
    }
    printf("Sumata na parni elementi: %d\nSumata na neparni elementi: %d\n", sumPar, sumNep);
    printf("Odnosot na parnite so neparnite elementi e %.2f\n", (float)brPar / brNep);
    return 0;
}

```

Задача 3

Да се напише програма која ќе го пресмета скаларниот производ на два вектори со по n координати. Бројот на координати n , како и координатите на векторите се внесуваат од тастатура. Резултатот да се испечати на екран.

Задача 3 Решение

Решение

```
#include <stdio.h>
#define MAX 100
int main() {
    int a[MAX], b[MAX], n, i, scalar = 0;
    printf("Vnesi ja goleminata na vektorite: ");
    scanf("%d", &n);
    printf("Vnesi gi koordinatite na prviot vector: \n");
    for(i = 0; i < n; ++i)
        scanf("%d", &a[i]);
    printf("Vnesi gi koordinatite na вториот vector: \n");
    for(i = 0; i < n; ++i)
        scanf("%d", &b[i]);
    for(i = 0; i < n; ++i)
        scalar += a[i] * b[i];
    printf("Scalarmiot proizvod na vektorite e: %d\n", scalar);
    return 0;
}
```

Задача 4

Да се напише програма која ќе провери дали дадена низа од n елементи која се внесува од тастатура е строго растечка, строго опаѓачка или ниту строго растечка ниту строго опаѓачка. Резултатот да се испечати на екран.

Задача 4 Решение

Решение

```
#include <stdio.h>
#define MAX 100
int main() {
    int n, element, a[MAX], i;
    short rastecka = 1, opagacka = 1;
    printf("Vnesi ja goleminata na nizata: \n");
    scanf("%d", &n);
    printf("Vnesi gi elementite od nizata: \n");
    for(i = 0; i < n; ++i)
        scanf("%d", &a[i]);
    for(i = 0; i < n-1; ++i) {
        if(a[i] >= a[i+1]) {
            rastecka = 0;
            break;
        }
    }
    for(i = 0; i < n-1; ++i) {
```

```

        if(a[i] <= a[i+1]) {
            opagacka = 0;
            break;
        }
    }
    if(!opagacka && !rastecka)
        printf("Nizata ne e nitu strogo rastecka nitu strogo opagacka\n");
    else if(opagacka)
        printf("Nizata e strogo opagacka\n");
    else if(rastecka)
        printf("Nizata e strogo rastecka\n");
    return 0;
}

```

Задача 5

Да се напише програма која што ќе ги избрише дупликатите од една низа. На крај, да се испечати на екран новодобиената низа. Елементите од низата се внесуваат од тестатура.

Задача 5 Решение

Решение

```

#include <stdio.h>
#define MAX 100
int main() {
    int a[MAX], n, i, j, k, izbrisani = 0;
    printf("Vnesi ja goleminata na nizata: \n");
    scanf("%d", &n);
    printf("Vnesi gi elementite od nizata: \n");
    for(i = 0; i < n; ++i)
        scanf("%d", &a[i]);
    for(i = 0; i < n - izbrisani; ++i)
        for(j = i + 1; j < n - izbrisani; ++j)
            if(a[i] == a[j]) {
                for(k = j; k < n - 1 - izbrisani; ++k)
                    a[k] = a[k + 1];
                izbrisani++;
            }
    n -= izbrisani;
    printf("Rezultantnata niza e: \n");
    for(i = 0; i < n; ++i)
        printf("%d\t", a[i]);
    return 0;
}

```

4 Матрици (дводимензионални полиња)

Задача 6

Да се напише програма која ќе испечати на екран дали дадена матрица е симетрична во однос на главната дијагонала. Димензиите и елементите на матрицата се внесуваат од тестатура.

Задача 6 Решение

Решение

```
#include <stdio.h>
#define MAX 100
int main() {
    int a[MAX][MAX], n, i, j;
    printf("Vnesi dimenzii na matricata: \n");
    scanf("%d", &n);
    printf("Vnesi gi elementite na matricata: \n");
    for(i = 0; i < n; ++i)
        for(j = 0; j < n; ++j)
            scanf("%d", &a[i][j]);
    for(i = 0; i < n - 1; ++i) {
        for(j = i + 1; j < n; ++j)
            if(a[i][j] != a[j][i])
                break;
        if(i != j) break;
    }
    if(i == j)
        printf("Matricata e simetricna vo odnos na glavnata dijagonala\n");
    else
        printf("Matricata ne e simetricna vo odnos na glavnata dijagonala\n");
    return 0;
}
```

5

Дефинирање на функции Потсетување од предавања

Дефиниција на функции

```
return_type function_name(arguments_list) {
    /* function body */
}
```

- return_type - типот на вредноста која што ја враќа функцијата
- function_name - името на функцијата
- arguments_list - листата со формални аргументи ги содржи аргументите заедно со нивните типови разделени со запирка телото на функцијата ги содржи истите елементи како и самата main() функција

Повикување на функции Потсетување од предавања

Повикување на функции

```
function_name(arguments_list);
```

- function_name – името на веќе дефинираната функција

- `arguments_list` - листата на аргументи е со вистински аргументи кои што ако се повеќе се одделуваат со запирка

Функции од математичката библиотека `math.h`

- Во C постои стандардна математичка библиотека `math.h` која што содржи многу готови математички функции
- Може да се употребуваат доколку претходно се уклучи математичката библиотека `#include<math.h>`
- Сите функции од стандардната библиотека `math.h` примаат аргументи од типот `double` и враќаат вредност од истиот тип

Вклучување на математичката библиотека

```
#include <math.h>
```

Најчесто користени математички функции `math.h`

<code>sqrt(x)</code>	квадратен корен од <code>x</code>
<code>exp(x)</code>	експоненцијална функција <code>ex</code>
<code>log(x)</code>	природен логаритам од <code>x</code> (со основа <code>e</code>)
<code>log10(x)</code>	логаритам од <code>x</code> со основа <code>10</code>
<code>fabs(x)</code>	апсолутна вредност од <code>x</code>
<code>ceil(x)</code>	заокружува <code>x</code> на најмалиот цел број не помал од <code>x</code>
<code>floor(x)</code>	заокружува <code>x</code> на најголемиот цел број не поголем од <code>x</code>
<code>pow(x, y)</code>	<code>x</code> на степен <code>y</code>
<code>fmod(x, y)</code>	остаток од <code>x/y</code> како реален број
<code>sin(x)</code>	синус од <code>x</code> (во радијани)
<code>cos(x)</code>	косинус од <code>x</code> (во радијани)
<code>tan(x)</code>	тангенс од <code>x</code> (во радијани)

Задача 1

Да се напише програма која што ќе ги отпечати сите четирицифрени природни броеви кои се деливи со збирот на двата броја составени од првите две цифри и од последните две цифри на четирицифрениот број. На крајот да отпечати колку вакви броеви се пронајдени.

Пример

3417 е делив со 34 + 17 5265 е делив со 52 + 65 6578 е делив со 65 + 78

Задача 1 Решение

```
#include <stdio.h>
int zbir_po_2cifri(n) {
    return (n % 100) + (n / 100);
}
int main() {
    int i, br = 0;
    for(i = 1000; i <= 9999; i++) {
        if (i % zbir_po_2cifri(i) == 0) {
            printf("Brojot %d go zadovoluva uslovot\n", i);
            br++;
        }
    }
    printf("Pronajdeni se %d broevi koi go zadovoluvaat uslovot\n", br);
    ;
    return 0;
}
```

Задача 2

Да се напише програма која за даден природен број ја пресметува разликата помеѓу најблискиот поголем од него прост број и самиот тој број.

Пример

Ако се внесе 573, програмата треба да испечати $577 - 573 = 4$

Задача 2 Решение

```
#include <stdio.h>
int prost(int n);
int prv_pogolem_prost(int n);
int main() {
    int broj;
    printf("Vnesi broj\n");
    scanf("%d", &broj);
    int prost = prv_pogolem_prost(broj);
    printf("%d - %d = %d\n", prost, broj, prost - broj);
    return 0;
}
int prost(int n) {
    int k;
    k = 2;
    while (k * k <= n) {
        if (n % k == 0)
            return 0;
        k++;
    }
    return 1;
}
int prv_pogolem_prost(int n) {
    do
        n++;
    while (!(prost(n)));
    return n;
}
```

Задача 3

Да се напише програма што ќе ги отпечати сите прости броеви помали од 10000 чиј што збир на цифри е исто така прост број. На крајот да се отпечати колку вакви броеви се пронајдени.

Пример

23 -> 2+3=5 179 -> 1+7+9=17 9613 -> 9+6+1+3=19

Задача 3 Решение

```
#include <stdio.h>
int e_prost(int n) {
    int i;
    if (n < 4)
        return 1;
    else if ((n % 2) == 0)
        return 0;
    else {
        i = 3;
        while (i * i <= n) {
            if (n % i == 0)
                return 0;
            i += 2;
        }
        return 1;
    }
}

int zbiri_cifri(int n) {
    int zbiri = 0;
    while (n > 0) {
        zbiri += (n % 10);
        n /= 10;
    }
    return zbiri;
}

int main() {
    int br = 0, i;
    for (i = 2; i <= 9999; i++) {
        if (e_prost(i) && e_prost(zbiri_cifri(i))) {
            printf("Brojot %d go zadovoluva uslovot\n", i);
            br++;
        }
    }
    printf("Pronajdeni se %d broevi koi go zadovoluvaat uslovot\n", br);
    return 0;
}
```

Задача 4

Да се напише програма што ќе ги отпечати сите парови прости броеви што се разликуваат меѓу себе за 2. На крај да се отпечати и нивниот број.

Пример

11 и 13 101 и 103 617 и 619 881 и 883

Задача 4 Решение

```
#include <stdio.h>
int eprost(int n);
int main () {
    int br=0,i;
    for (i=1; i<=(1000-2); i++) {
        if (eprost(i) && eprost(i+2)) {
            printf("Prostire broevi %d I %d se razlikuvaat za 2\n", i,
                (i+2));
            br++;
        }
    }
    printf("Pronajdeni se vkupno %d parovi prosti broevi koi go
        zadovoluvaat uslovot\n",br);
    return 0;
}
int eprost(int n) {
    int i;
    if (n < 4) return 1;
    else
    if ((n%2)==0) return 0;
    else {
        i=3;
        while (i*i<=n){
            if (n%i==0) return 0;
            i+=2;
        }
    }
    return 1;
}
```

Задача 5

Да се пресмета збирот: $1! + (1+2)! + (1+2+3)! + \dots + (1+2+\dots+n)!$ Помош:

- Користете функција за пресметување на збирот на првите k природни броеви
- Користете функција за пресметување факториел на еден природен број k

Задача 5 Решение

```
#include <stdio.h>
int factoriel(int k) {
    int i, fact_num = 1;
    for (i = 1; i <= k; i++)
        fact_num *= i;
    return fact_num;
}
int suma(int k) {
    int i, zbir = 0;
    for (i = 1; i <= k; i++)
        zbir += i;
    return zbir;
}
int main() {
    int i, n, rezultat = 0;
```

```

printf("Vnesete eden pozitiven cel broj \n");
if (scanf("%d", &n) && n > 0) {
    for (i = 1; i < n; i++) {
        rezultat += factoriel(suma(i));
        printf("%d! + ", suma(i));
    }
    rezultat += factoriel(suma(n));
    printf("%d! = %d", suma(n), rezultat);
} else
    printf("Vnesena e pogresna vrednost \n");
return 0;
}

```

Задача 6

Да се напише функција што прима два параметра x и n и враќа:

$$f(n) = \begin{cases} x + \frac{x^n}{n} + \frac{x^{n+2}}{n+2} & , x \geq 0 \\ -\frac{x^{n-1}}{n-1} + \frac{x^{n+1}}{n+1} & , x < 0 \end{cases}$$

Потоа да се состави програма што ќе ја табелира оваа функција за прочитано n во интервал $x \in [-4, 4]$, со чекор 0.1.

Задача 6 Решение

```

#include <stdio.h>
double f(float i, int j) {
    double vrednost;
    if (i > 0)
        vrednost = i + stepen(i, j) - stepen(i, j + 2);
    else
        vrednost = -stepen(i, j - 1) + stepen(i, j + 1);
    return vrednost;
}
float stepen(float i, int j) {
    int k;
    double vrednost;
    if (i == 0)
        vrednost = 0.0;
    else {
        vrednost = 1.0;
        for (k = 1; k <= j; ++k)
            vrednost *= i;
    }
    return vrednost;
}

```

```

int main() {
    int n;
    float x;
    printf("Vnesi broj:\n");
    scanf("%d", &n);
    if ((n >= -2) && (n <= 1))
        printf("Neodredeno.\n");
    else {
        x = -4.0;

```

```

        for (x = -4; x <= 4; x += 0.1) {
            printf("x=%3.1f, f(x)=%10.4f\n", x, f(x, n));
        }
    }
    return 0;
}

```

Рекурзивни функции Потсетување од предавања

- Функциите во C може да повикуваат други функции
 - Ова повикување може да оди во произволна длабочина
- Една функција може да се повикува самата себе
- Такиот начин на повикување се нарекува **рекурзија**

Пример на рекурзивна функција

```

int faktoriel(int n) {
    if(n == 0) return 1;
    else return n * faktoriel(n - 1);
}

```

Задача 1

Да се пресмета збирот: $1! + (1+2)! + (1+2+3)! + \dots + (1+2+\dots+n)!$ Овој пат:

- Користете **рекурзивна** функција за пресметување на збирот на првите k природни броеви
- Користете **рекурзивна** функција за пресметување факториел на еден природен број k

Задача 1 Решение

```

#include <stdio.h>

int faktoriel(int n) {
    if (n == 0)
        return 1;
    else
        return n * faktoriel(n - 1);
}

int suma(int k) {
    if (k == 0)
        return 0;
    else
        return k + suma(k - 1);
}

```

Задача 1 Решение

```
int main() {
    int i, n, rezultat = 0;
    printf("Vnesete eden pozitiven cel broj \n");
    scanf("%d", &n);
    if (n > 0) {
        for (i = 1; i < n; i++) {
            rezultat += faktoriel(suma(i));
            printf("%d! + ", suma(i));
        }
        rezultat += faktoriel(suma(n));
        printf("%d! = %d", suma(n), rezultat);
    } else
        printf("Vnesena e pogreshna vrednost \n");
    return 0;
}
```

Задача 2

Да се напише програма која за даден природен број ја пресметува разликата помеѓу најблискиот поголем од него прост број и самиот тој број. Програмата треба да користи **рекурзивна** функција за наоѓање на соодветниот прост број.

Пример

Ако се внесе 573, програмата треба да испечати $577 - 573 = 4$

Задача 2 Решение

```
#include <stdio.h>

int e_prost(int n, int i);
int prv_pogolem_prost(int n);

int main() {
    int broj, razlika;
    printf("Vnesi broj\n");
    scanf("%d", &broj);
    razlika = prv_pogolem_prost(broj) - broj;
    printf("Razlikata pomegju prostiot broj %d i %d e %d\n",
        prv_pogolem_prost(broj), broj, razlika);
    return 0;
}

int e_prost(int n, int i) {
    if (n < 4)
        return 1;
    else if ((n % 2) == 0) return 0;
    else if (n % i == 0) return 0;
    else if (i * i > n) return 1;
    else return e_prost(n, i + 2);
}

int prv_pogolem_prost(int n) {
    if (e_prost(n + 1, 3)) return n + 1;
    else return prv_pogolem_prost(n + 1);
}
```

Задача 3

Да се напише програма што ќе ја испишува вредноста на n-тиот член на низата дефинирана со:

$$\begin{aligned}x_1 &= 1 \\x_2 &= 2 \\&\vdots \\x_n &= \frac{n-1}{n}x_{n-1} + \frac{1}{n}x_{n-2}\end{aligned}$$

Задача 3 Решение

```
#include <stdio.h>
float xnn(float x1, float x2, int n) {
    if (n == 1)
        return x1;
    if (n == 2)
        return x2;
    return (n - 1) * xnn(x1, x2, n - 1) / n + xnn(x1, x2, n - 2) / n;
}
int main() {
    int n;
    printf("Vnesi n:\n");
    scanf("%d", &n);
    printf("xnn(1, 2, %d) = %.2f\n", n, xnn(1, 2, n));
    return 0;
}
```

Задача 4

Да се напише рекурзивна функција која ќе го пресметува збирот на цифрите на еден број.

Пример

```
sumDigits(126) -> 9
sumDigits(49) -> 13
sumDigits(12) -> 3
```

```
#include <stdio.h>
int sumDigits(int n) {
    if(n == 0) return 0;
    return n % 10 + sumDigits(n / 10);
}
```

Задача 5

За даден број n, да се напише рекурзивна функција која ќе ги изброи појавувањата на цифрата 8, со тоа што ако се појави цифра 8 со уште една цифра 8 веднаш лево од неа таа се брои двојно, така 8818 дава 4.

Пример

```
count8(8) -> 1
count8(818) -> 2
count8(8818) -> 4
```

```
#include <stdio.h>
int count8(int n) {
    if (n == 0)
        return 0;
    if ((n / 10) % 10 == 8 && n % 10 == 8)
        return 2 + count8(n / 10);
    if (n % 10 == 8)
        return 1 + count8(n / 10);
    return count8(n / 10);
}
```

Задача 6

Да се напише програма која што за дадена низа од природни броеви која што се внесува од тастатура) ќе го отпечати најголемиот заеднички делител (НЗД) на нејзините елементи. Програмата задолжително треба да содржи рекурзивна функција за пресметување на НЗД на два природни броја.

Пример

48 36 120 72 84 На екран треба да се отпечати: NZD na elementite na ovaа niza e 12

Евклидов алгоритам Задача 6

- НЗД за два броја може да се пресмета со користење на Евклидовиот алгоритам
- За да се пресмета НЗД за броевите m и n , се пресметува остатокот при делење на m со n
 - Ако остатокот не е 0, се пресметува остатокот при делење на n со $m \% n$
 - Постапката се повторува сè додека се добиваат ненулни остатоци
 - Ако остатокот е 0, НЗД за двата броја е последниот пресметан ненулни остаток

Пример

```
НЗД(20, 12)
20 % 12 = 8
12 % 8 = 4
8 % 4 = 0
НЗД(20, 12) = 4
```

Задача 6 Решение

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

int nzd(int m, int n) {
    if (!n)
        return m;
    return nzd(n, m % n);
}

int main() {
    int i, n, a[MAX];
    printf("Vnesi ja goleminata na nizata:");
    scanf("%d", &n);
    printf("Vnesi gi elementite na nizata: \n");
    for (i = 0; i < n; i++)
        scanf("%d", &a[i]);
    int NZD = nzd(a[0], a[1]);
    for (i = 2; i < n; i++)
        NZD = nzd(NZD, a[i]);
    printf("NZD na elementite na ova niza e %d", NZD);
    return 0;
}
```

Задача 7 За дома

Да се напише програма која што за дадена низа од природни броеви (која што се внесува од тастатура) ќе го отпечати најмалиот заеднички содржател (НЗС) на нејзините елементи. Програмата задолжително треба да содржи рекурзивна функција за пресметување на НЗС на два природни броја.

Пример

За низата:

18 12 24 36 6

На екран треба да се отпечати:

NZS na elementite na ova niza e 72

Задача 8

Да се напише програма која за дадена низа од цели броеви (која што се внесува од тастатура) ќе го отпечати најмалиот елемент. Програмата треба да содржи рекурзивна функција за наоѓање на најмал елемент во дадена низа.

Пример

За низата:

5 8 3 12 9 6

На екран треба да се отпечати:

Najmal element vo nizata e 3

Задача 8 Решение

```
#include <stdio.h>
int minElement(int niza[], int n);
int main() {
    int i, n, a[100];
    printf("Vnesi ja goleminata na nizata: \n");
    scanf("%d", &n);
    printf("Vnesi gi elementite na nizata: \n");
    for (i = 0; i < n; i++)
        scanf("%d", &a[i]);
    printf("Najmal element vo nizata e: %d \n", minElement(a, n - 1));
    return 0;
}
int minElement(int niza[], int n) {
    if (n == 0)
        return niza[n];
    else {
        int pom = minElement(niza, n - 1);
        if (niza[n] < pom)
            return niza[n];
        return pom;
    }
}
```

Покажувачи (Pointers) Потсетување од предавања

Што е покажувач (pointer)?

Покажувач е *податочен тип* кој што чува (покажува кон) некоја мемориска локација. Оваа мемориска локација се чува преку нејзината адреса (некаков број).

Како се декларира покажувач?

`pointer_type *name`; Пример: `float *p`; Вака декларираниот покажувач не покажува никаде!

Зошто служат покажувачите?

- За брзо и ефикасно изминување на сложени податочни структури како низи и дрва, ...
- За ефикасно пренесување на сложени аргументи во функции. Пренесување на низа, структура и слично
- За пренесување на аргументи чии што вредности сакаме да останат во онаа состојба во која се наоѓаат по извршувањето на функцијата

Задача 1

Да се напише функција која за низа од N цели броеви ќе ги пронајде почетокот и должината на најголемата растечка подниза.

Пример

За низата 2 3 1 4 7 12 7 9 1 ќе врати 2 4

Задача 1 Решение

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

void maxRastecka(int x[], int n, int *pos, int *len) {
    int i, start, currLen;
    start = 0;
    currLen = 1;
    *pos = 0;
    *len = 1;
    for(i = 0; i < n - 1; i++) {
        start = i;
        currLen = 1;
        while ((x[i] < x[i + 1])) {
            currLen++;
            i++;
            if(i >= n) break;
        }
        if (currLen > *len) {
            *len = currLen;
            *pos = start;
        }
    }
}

int main() {
    int a[MAX];
    int i, n, pos, len;

    printf("Dolzina na nizata: ");
    scanf("%d", &n);
    for (i = 0; i < n; i++)
        scanf("%d", &a[i]);

    maxRastecka(a, n, &pos, &len);

    printf("Pocetok: %d, dolzina: %d\n", pos, len);
    return 0;
}
```

Задача 2

Да се напише функција која влезната низа a_0, b_1, \dots, b_{n-1} ќе ја трансформира во излезна низа b_0, b_1, \dots, b_{n-1} на следниот начин

$$\begin{aligned} b_0 &= a_0 + a_{n-1} \\ b_1 &= a_1 + a_{n-2} \\ &\vdots \\ b_n &= a_{n-1} + a_0 \end{aligned}$$

Пример

Влезната низа 1 2 3 5 7 треба да се трансформира во 8 7 6 7 8

Задача 2 Решение

```
#include "stdio.h"
#define MAX 100

void promena(int *a, int n) {
    int i, j;
    for (i = 0, j = n - 1; i < j; i++, j--) {
        *(a + i) += *(a + j);
        *(a + j) = *(a + i);
    }
    if (n % 2) {
        *(a + n / 2) *= 2;
    }
}

int main() {
    int i, n;
    int a[MAX];
    printf("Kolku elementi ima nizata: ");
    scanf("%d", &n);
    printf("Vnesi gi elementite na nizata\n");
    for (i = 0; i < n; i++)
        scanf("%d", &a[i]);
    for (i = 0; i < n; i++)
        printf("a[i] = %d\n", a[i]);
    promena(a, n);
    for (i = 0; i < n; i++)
        printf("b[i] = %d\n", a[i]);
    return 0;
}
```

Задача 3

Да се напишат следните функции за пребарување во низа:

- Линеарно пребарување
- Бинарно пребарување

Потоа да се напише главна програма во која ќе се пополнува низа со броевите од 1 до 1 000 000, а потоа се генерира случаен број во овој опсег чија што позиција треба да се пронајде со повикување на двете функции за пребарување. **За дома:** За двете функции избројте го и споредето го бројот на потребни итерации за проаноѓање на бројот.

Задача 3 Решение

```
int linearSearch(int *a, int n, int key) {
    int i;
    for(i = 0; i < n; i++) {
        if(*(a + i) == key) return i;
    }
    return -1;
}

int binarySearch(int *a, int n, int key) {
    int start = 0;
    int end = n - 1;
    while(start <= end) {
```

```

        int mid = (start + end) / 2;
        if(*(a + mid) == key) return mid;
        else if(*(a + mid) > key) end = mid - 1;
        else start = mid + 1;
    }
    return -1;
}

```

За дома: Да се напише рекурзивна функција за бинарно пребарување

Задача 3 Решение

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define MAX 1000000

int main() {
    int i;
    int *a = malloc(sizeof(int) * MAX);
    for(i = 0; i < MAX; i++){
        *(a + i) = i + 1;
    }
    srand(time(NULL));
    int key = rand() % MAX + 1;
    printf("Element што се бара: %d\n", key);
    int found = linearSearch(a, MAX, key);
    printf("Najden so linearno preberavanje na pozicija: %d\n", found);
    ;
    found = binarySearch(a, MAX, key);
    printf("Najden so binarno preberavanje na pozicija: %d\n", found);
    return 0;
}

```

Задача 4

Да се напишат функции за сортирање на низа со помош на следните методи за сортирање:

- Метод на меурче (Bubble sort)
- Метод со избор на елемент (Selection sort)
- Метод со вметнување (Insertion sort)

Да се напишат функции за внесување и печатење на елементите на една низа и да се напише главна програма во која се тестираат сите методи за сортирање.

Bubble sort Задача 4

Се започнува од првиот елемент и се споредуваат секои два соседни додека не се дојде до последниот елемент. При секое споредување, ако претходниот има поголема вредност, тогаш си ги заменуваат местата. Така најголемиот елемент се доведува на последната позиција во низата. Се повторува истата постапка од 1-от до претпоследниот елемент во низата, така

што сега на претпоследната позиција ќе исплива елемент помал од најголемиот елемент во низата итн. На крајот се споредуваат само 1-от и 2-от елемент од низата.

```
void bubbleSort(int *a, int n) {
    int i, j;
    for (i = 0; i < n; i++) {
        for (j = 0; j < n - i - 1; j++) {
            if (a[j] > a[j + 1])
                swap(&a[j], &a[j + 1]);
        }
    }
}
```

Selection sort Задача 4

Се пронаоѓа најмалиот елемент во низата и истиот се заменува со првиот елемент. Потоа, првиот елемент на низата се игнорира (бидејќи се знае дека тој е најмал) и рекурзивно се сортира преостанатата поднiza (од вториот елемент, па до крајот). Постапката се повторува сè дури не остане само еден елемент. Тоа е граничниот случај – се престанува со сортирањето

```
void selectionSort(int niza[], int n, int m) {
    if (n - m == 1)
        return;
    else {
        int najmal = niza[m];
        int indeksNajmal = m;
        int i;
        for (i = m; i < n; ++i)
            if (niza[i] < najmal) {
                najmal = niza[i];
                indeksNajmal = i;
            }
        swap(&niza[m], &niza[indeksNajmal]);
        selectionSort(niza, n, m + 1);
    }
}
```

Insertion sort Задача 4

Со овој метод се сортира на тој начин што секој елемент се вметнува на соодветната позиција, од што и доаѓа самото име. Во првата итерација, вториот елемент $a[1]$ се споредува со првиот елемент $a[0]$. Во втората итерација третиот елемент се споредува со првиот и вториот. Генерално, во секоја итерација елементот се споредува со сите елементи пред него. Ако при споредбата се покаже дека тој елемент треба да се вметне на соодветната позиција, тогаш се создава простор со поместување на сите елементи десно од тој елемент за еден и се вметнува елементот. Оваа процедура се повторува за секој елемент во низата.

```
void insertionSort(int a[], int n) {
    int i, j;
    for (i = 1; i < n; i++) {
        int temp = a[i];
        j = i - 1;
        while (temp < a[j] && j >= 0) {
            a[j + 1] = a[j];
            j--;
        }
    }
}
```

```
        a[j + 1] = temp;
    }
}
```

Задача 4 Решение

```
void insert(int a[], int n) {
    int i;
    for(i = 0; i < n; i++) {
        printf("a[%d] = ", i);
        scanf("%d", &a[i]);
    }
}

void print(int *a, int n) {
    int i;
    for(i = 0; i < n; i++) {
        printf("%d\t", *(a + i));
    }
    printf("\n");
}

int main() {
    int a[MAX], n;
    printf("Vnesi dolzina na nizata: ");
    scanf("%d", &n);
    insert(a, n);
    bubbleSort(a, n);
    //selectionSort(a, n, 0);
    //insertionSort(a, n);
    print(a, n);
    return 0;
}
```

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 текстуална 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 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 broiZnak(char *str, char znak) {
    int vkupno = 0;
    while (*str != '\0') {
        vkupno += (*str == znak);
        str++;
    }
    return vkupno;
}
int main() {
    char s[MAX], znak;
    printf("Vnesete string:");
    gets(s);
    printf("Vnesete znak koj treba da se bara vo vneseniot sting: \n");
    ;
    znak = getchar();
    printf(
        "Vo vneseniot string \"%s\", znakot \"%c\" se pojavuva
        vkupno %d pati.\n",
        s, znak, broiZnak(s, znak));
    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 dolzina(char *s) {
    int i, len = 0;
    for (i = 0; s[i] != '\0'; i++)
        len++;
    return len;
}

int length(char *s) {
    if (*s == '\0')
        return 0;
}
```



```

    return 1 + length(s + 1);
}

int main() {
    char s[MAX];
    printf("Vnesete ja tekstualna niza:\n");
    gets(s);
    printf("Dolzina: %d i %d\n", dolzina(s), length(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 poz, dolz;
    printf("Vnesete string: ");
    gets(s);
    printf("Vnesete pozicija i broj na znaci za podnizata: \n");
    scanf("%d %d", &poz, &dolz);
    if (poz <= strlen(s)) {
        strncpy(dest, s + poz - 1, dolz);
        dest[dolz] = '\0';
        printf("Novo dobienata tekstualna niza e: ");
        puts(dest);
    } else
        printf("Vnesena e nevalidna pozicija za podnizata, vnesenata\n niza ima samo %d znaci.\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 podniza(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];
    printf("Vnesete ja prvata tekstualna niza:\n");
    gets(s1);
    printf("Vnesete ja vtorata tekstualna niza:\n");
    gets(s2);
    if (podniza(s1, s2))
        printf("%s e podniza na %s\n", s1, s2);
    else
        printf("%s ne e podniza na %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, интерпунктиските 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 palindrom(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;
}
```

```

}
// REKURZIVNO
int pal(char *str, int start, int end) {
    if(start >= end) return 1;
    if(str[start] == str[end])
        return pal(str, start + 1, end - 1);
    return 0;
}

int main() {
    char s[MAX];
    printf("Vnesete string: ");
    gets(s);
    printf("Vneseniot string %s ", s);
    if (pal(s, 0, strlen(s) - 1))
        printf("e palindrom.");
    else
        printf("ne e palindrom.");
    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 <ctype.h>
#include <string.h>
#define MAX 100

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

int main() {
    char s[MAX];
    printf("Vnesete string: ");
    gets(s);
    printf("Vneseniot string %s ", s);
    if (eLozinka(s))
        printf("moze da bide lozinka.");
    else
        printf("ne moze da bide lozinka.");
}

```

```
    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];
    printf("Vnesete string: ");
    gets(s);
    filter(s);
    printf("Novo dobieniot string e: [%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

`\input{code.c}`
