



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

## Аудиториски вежби 7

### Функции

Концепти за развој на софтвер

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

Предавања, аудиториски вежби, соопштенија  
[courses.finki.ukim.mk](https://courses.finki.ukim.mk)

Изворен код на сите примери и задачи  
[bitbucket.org/tdelev/finki-krb](https://bitbucket.org/tdelev/finki-krb)

## Прашања ?