



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

## Аудиториски вежби 7

### Рекурзивни функции

Структурно Програмирање

# 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
- Таквиот 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:

$$x_1 = 1$$

$$x_2 = 2$$

$$\vdots$$

$$x_n = \frac{n-1}{n}x_{n-1} + \frac{1}{n}x_{n-2}$$



# 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("Внеси 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 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 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 ovaa 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 ovaa 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 ovaa 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;
    }
}
```

# 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`