

# Аудиториски вежби 5

## Функции и рекурзија



М-р Ѓорѓи Маџаров  
М-р Томче Делев

Структурирано програмирање

Факултет за информатички  
науки и компјутерско  
инженерство - Скопје 2011

# Содржина

---

Функции

Рекурзија



# Задача 1

---

Пример за void функции и функции без параметри.

## Решение

---

```
#include <stdio.h>
/* Deklaracija na funkcii */
void printMax(int broj);
void printPozdrav();
int main() {
    int k = 10;
    printPozdrav(); /* Pecati Pozdrav */
    printMax(k); /* Ja pecati maksimalnata vrednost */
    return 0;
}
/* Definicija na funkciiite */
void printMax(int broj) {
    printf("Maksimalniot broj e %d\n", broj);
}
void printPozdrav() {
    printf("Dobar Den. Kako se cuvstvuvate denes?\n");
}
```

---



## Задача 2

---

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

### Пример

3417 е делив со  $34 + 17$ , 5265, 6578, ....



# Задача 2

## Решение

---

```
#include <stdio.h>
int zb2cif(int n);
int main() {
    int br=0,i;
    for (i=1000; i<=9999; i++) {
        if (i%zb2cif(i)==0) {
            printf("Brojot %d go zadovoluva uslovot\n",i);
            br++;
        }
    }
    printf("Pronajdeni se %d broevi koi go zadovoluvaat uslovot\n",br);
    return 0;
}

int zb2cif(int n) {
    int zbir;
    zbir=(n%100)+(n/100);
    return zbir;
}
```

---



## Задача 3

---

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



# Задача 3

## Решение

---

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



## Задача 4

---

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 биле пронајдени.  
пример: 23, 179, 9613, ...





# Задача 4

## Решение

---

```
#include <stdio.h>
int eprost(int n);
int zbircif(int n);
int main () {
    int br=0,i;
    for (i=2; i<=9999; i++) {
        if (eprost(i) && eprost(zbircif(i))) {
            printf("Brojot %d go zadovoluva uslovot\n",i);
            br++;
        }
    }
    printf("Pronajdeni se %d broevi koi go zadovoluvaat uslovot\n",br);
    return 0;
}
int eprost(int n) {
    int i, prost;
    if (n<4) prost=1;
    else
    if ((n%2)==0) prost =0;
    else {
        i=3; prost=1;
        while ((i*i<=n) && prost) {
            if (n%i==0) prost=0;
            i+=2;
        }
    }
    return prost;
}
int zbircif(int n) {
    int zbir=0;
    while (n>0) {
        zbir+=(n%10);
        n/=10;
    }
    return zbir;
}
```

---



## Задача 5

---

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.



# Задача 5

## 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 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);
float stepen(float i, int j);
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;
        while (x<=4) {
            printf("x=%3.1f, f(x)=%10.4f\n", x, f(x,n));
            x+=0.1;
        }
        return 0;
    }
}
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;
}
```



# Содржина

---

Функции

Рекурзија



## Задача 7

---

Да се напише програма која пресметува факториел на даден број. Факториел да се пресметува во посебна рекурзивна функција.

$$n! = n * (n - 1) * (n - 2) \dots * 2 * 1$$

---

```
#include <stdio.h>
int factorial(int n) {
    if(n == 0) return 1;
    return n * factorial(n - 1);
}
int main () {
    int n;
    printf("Vnesi broj:\n");
    scanf("%d", &n);
    printf("%d! = %d\n", n, factorial(n));
    return 0;
}
```

---



## Задача 8

---

Да се напише програма која ја пресметува сумата на следната низа:

$$1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$

---

```
#include <stdio.h>
int sum(int n) {
    if(n == 1) return 1;
    return 1.0 / n + sum(n - 1);
}
int main () {
    int n;
    printf("Vnesi broj:\n");
    scanf("%d", &n);
    printf("sum(%d) = %.2f\n", n, sum(n));
    return 0;
}
```

---





## Задача 9

---

Да се напише програма која за дадено  $N$  ќе го испише соодветниот Фибоначиев број. Фибоначиевите броеви се дефинирани на следниов начин:

$$a_1 = 1$$

$$a_2 = 1$$

$$\vdots$$

$$a_n = a_{n-1} + a_{n-2}$$

### Пример

1, 1, 2, 3, 5, 8, 13, 21, 34, ...



# Задача 9

## Решение

---

---

```
#include <stdio.h>
int fibonaci(int n) {
    if(n == 0 || n == 1) return 1;
    return fibonaci(n - 1) + fibonaci(n - 2);
}
int main () {
    int n;
    printf("Vnesi broj:\n");
    scanf("%d", &n);
    printf("fib(%d) = %d\n", n, fibonaci(n));
    return 0;
}
```

---



## Задача 10

---

Да се напише програма што ќе ја испишува вредноста на произволен 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 10

## Решение

---

---

```
#include <stdio.h>
float xnn(float x1, float x2, int n) {
    if(n == 1) return 1;
    if(n == 2) return 2;
    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, 1, %d) = %.2f\n", n, xnn(1, 1, n));
    return 0;
}
```

---



# Материјали и прашања

---

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

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

Прашања и одговори  
[qa.finki.ukim.mk](https://qa.finki.ukim.mk)

