



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

Структурно програмирање

Аудиториски вежби 7

Верзија 1.0, 14 Ноември, 2016

Содржина

1. Функции	1
1.1. Потсетување од предавања	1
1.2. Функции од математичката библиотека <code>math.h</code>	2
1.3. Најчесто користени математички функции	2
1.4. Задача 1	3
1.5. Задача 2	4
1.6. Задача 3	5
1.7. Задача 4	6
1.8. Задача 5	6
1.9. Задача 6	7
2. Изворен код од примери и задачи	9

1. Функции

1.1. Потсетување од предавања

1.1.1. Дефиниција на функција во C

```
tip ime(lista_na_formalni_argumenti) {  
    telo_na_funkcijata  
}
```

- `tip` - типот на вредноста која ја враќа функцијата
- `ime` - името на функцијата
- `lista_na_formalni_argumenti` - листата со формални аргументи ги содржи аргументите заедно со нивните типови, разделени со запирка
- `telo_na_funkcijata` - телото на функцијата ги содржи истите елементи како и самата `main()` функција

1.1.2. Повик на функција

```
ime(lista_na_argumenti);
```

- `ime` - името на веќе дефинираната функција
- `lista_na_argumenti` - листата на аргументи е со вистински аргументи кои што ако се повеќе ако се одделуваат со запирка

1.1.3. Пример на кориснички дефинирана функција

Да се напише програма во која со посебна функција се пресметува куб n^3 за вчитан природен број n .

Пример ex7_1.c

```
#include <stdio.h>

double kub(int x) {
    return x * x * x;
}

int main() {
    int n;
    printf("Vnesete eden prirodan broj: ");
    scanf("%d", &n);
    double rezultat = kub(n);

    printf("Kubot na brojot %d e %.2f\n", n, rezultat);
    return 0;
}
```

1.2. Функции од математичката библиотека `math.h`

- Во C постои стандардна математичка библиотека `math.h` која што содржи многу готови математички функции.
- За да се употребува, треба претходно да се вклучи со: `#include <math.h>`
- Сите функции од стандардната библиотека `math.h` примаат аргументи од тип `double` и враќаат вредности од истиот тип.

1.3. Најчесто користени математички функции

Функција	Објаснување
<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>	закржува x на најмалиот цел број не помал од x
<code>floor(x)</code>	закржува 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)

1.3.1. Пример на користење на функција од `math.h`

Да се напише програма во која со посебна функција се пресметува куб n^3 за вчитан природен број n .

Пример ex7_2.c

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

int main() {
    int n;
    printf("Vnesete eden prirodan broj: ");
    scanf("%d", &n);
    double rezultat = pow(n, 3);

    printf("Kubot na brojot %d e %.2f\n", n, rezultat);
    return 0;
}
```

1.4. Задача 1

Да се напишат соодветни функции за пресметување на дијаметар, периметар и плоштина на круг чиј што радиус се предава како аргумент. Потоа да се напише и програма во која за внесен (од тастатура) радиус ќе се повикаат овие функции за да се пресметаат дијаметарот, периметарот и плоштината на соодветниот круг.

```
#include <stdio.h>
#define PI 3.14

double dijametar(double radius);
double perimetar(double radius);
double ploshtina(double radius);

int main() {
    float radius, D, L, P;
    printf("Vnesete radius na krugot: ");
    scanf("%f", &radius);

    D = dijametar(radius);
    L = perimetar(radius);
    P = ploshtina(radius);

    printf("Dijametar na krugot = %.2f\n", D);
    printf("Perimetar na krugot = %.2f\n", L);
    printf("Ploshtina na krugot = %.2f\n", P);
    return 0;
}

double dijametar(double radius) {
    return 2 * radius;
}

double perimetar(double radius) {
    return 2 * radius * PI;
}

double ploshtina(double radius) {
    return radius * radius * PI;
}
```

1.5. Задача 2

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 треба да отпечати и колку вакви броеви се пронајдени.

Пример:

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

Решение p7_5.c

```
#include <stdio.h>

int zbir_po_2cifri(int n) {
    return n % 100 + n / 100;
}

int main() {
    int i;
    int count = 0;
    for(i = 1000; i <= 9999; ++i) {
        if(i % zbir_po_2cifri(i) == 0) {
            printf("%d\n", i);
            ++count;
        }
    }
    printf("Vkupno: %d\n", count);
    return 0;
}
```

1.6. Задача 3

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 $577 - 573 = 4$

Решение p7_6.c

```
#include <stdio.h>

int prost(int n) {
    int i;
    for(i = 2; i * i <= n; ++i) {
        if(n % i == 0) {
            return 0;
        }
    }
    return 1;
}

int prv_pogolem_prost(int n) {
    ++n;
    while(!prost(n)) {
        ++n;
    }
    return n;
}

int main() {
    int n;
    scanf("%d", &n);
    int pogolem_prost = prv_pogolem_prost(n);
    printf("%d - %d = %d\n", pogolem_prost, n, pogolem_prost - n);
    return 0;
}
```

1.7. 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 се пронајдени.

Решение p7_7.c

```
#include <stdio.h>

int is_prime(int n) {
    if(n < 4) return 1;
    else {
        if(n % 2 == 0) return 0;
        else {
            int i;
            for(i = 3; i * i <= n; i += 2) {
                if(n % i == 0) {
                    return 0;
                }
            }
        }
    }
    return 1;
}

int sum_digits(int n) {
    int sum = 0;
    while(n != 0) {
        sum += n % 10;
        n /= 10;
    }
    return sum;
}

int main() {
    int i, count = 0;
    for(i = 2; i <= 9999; ++i) {
        if(is_prime(i) && is_prime(sum_digits(i))) {
            printf("%d\t", i);
            ++count;
        }
    }
    printf("\nVkupno: %d\n", count);
    return 0;
}
```

1.8. Задача 5

Да се напише програма што ќе ги отпечати сите парови прости броеви помали од 1000 што се разликуваат меѓу себе за 2. На крај да се отпечати и нивниот број.


```
#include <stdio.h>

int prost(int n) {
    int i;
    for(i = 2; i * i <= n; ++i) {
        if(n % i == 0) {
            return 0;
        }
    }
    return 1;
}

int main() {
    int i, count = 0;
    for(i = 1; i < 998; ++i) {
        if(prost(i) && prost(i + 2)) {
            printf("%d %d\n", i, i + 2);
            ++count;
        }
    }
    printf("Vкупно: %d\n", count);
    return 0;
}
```

1.9. Задача 6

Да се пресмета збирот:

$$1! + (1 + 2)! + (1 + 2 + 3)! + \dots + (1 + 2 + \dots + n)!$$



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

```
#include <stdio.h>

int suma(int n) {
    int i;
    int s = 0;
    for(i = 1; i <= n; ++i) {
        s += i;
    }
    return s;
}

int faktoriel(int n) {
    int result = 1;
    int i;
    for(i = 1; i <= n; ++i) {
        result *= i;
    }
    return result;
}

int main() {
    int n;
    scanf("%d", &n);
    if(n > 0) {
        int i;
        int result = 0;
        int s;
        for(i = 1; i < n; ++i) {
            s = suma(i);
            result += faktoriel(s);
            printf("%d! + ", s);
        }
        s = suma(n);
        result += faktoriel(s);
        printf("%d! = %d\n", s, result);
    } else {
        printf("Nevalidna vrednost\n");
    }
    return 0;
}
```

2. Изворен код од примери и задачи

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

Source code ZIP