



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

## Структурно програмирање

Аудиториски вежби 3

# Содржина

1. Потсетување од предавања .....	1
1.1. Релациски оператори .....	1
1.2. Логички оператори .....	1
1.3. Дополнителни оператори .....	2
1.4. Оператор за доделување = .....	2
1.5. Оператори за инкрементирање и декрементирање .....	2
1.6. Двојни оператори .....	3
2. Примери .....	3
2.1. Работа со променливи и доделување вредности .....	3
2.2. Промена на вид на вредност – cast операција .....	4
2.3. Претворање во друг тип .....	4
2.4. Кастирање .....	5
3. Задачи .....	5
3.1. Задача 1 .....	5
3.2. Задача 2 .....	6
3.3. Задача 3 .....	7
3.4. Задача 4 .....	7
3.5. Задача 5 .....	8
3.6. Задача 6 .....	8
3.7. Задача 7 .....	9
3.8. Задача 8 .....	9
4. Изворен код од примери и задачи .....	11

# 1. Потсетување од предавања

- Оператори
  - Аритметички
  - Релациски
  - Логички
- Промена на вид на вредност – **cast** операција

## 1.1. Релациски оператори

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

Оператор	Значење
<	Помало
<=	Помало или еднакво
>	Поголемо
>=	Поголемо или еднакво
==	Еднаквост
!=	Не еднаквост (различно)

## 1.2. 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 <b>И</b> (AND)
	Логичко <b>ИЛИ</b> (OR)
!	Негација

Пример:

```
int a = 5 && 0; // a = 0;
a = 2 && 5;    // a = 1;
a = 0 || 5;    // a = 1;
a = !0;        // a = 1;
a = !5;        // a = 0;
```

## 1.3. Дополнителни оператори

- Оператор за доделување =
- Оператори за инкрементирање и декрементирање (++ , --)
  - ++ инкрементирање (зголемување на вредноста на операндот за 1)
  - -- декрементирање (намалување на вредноста на операндот за 1)
- Користење на операторите + и – на унарен начин

```
x = + y;  
x = - y;
```

- Двојни оператори
  - Комбинација од оператор за доделување и друг оператор (+ = , - = , \* = , / = , % =)

## 1.4. 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;
```

## 1.5. Оператори за инкрементирање и декрементирање

- Оператор за инкрементирање ++ (зголемување на вредноста на операндот за 1)
- Оператор за декрементирање -- (намалување на вредноста на операндот за 1)
- Можат да бидат зададени во префикс или постфикс нотација:

### 1.5.1. Префикс

Вредноста на променливата се зголемува пред да се пресмета изразот во која

таа учествува

```
a = ++b;
```

## 1.5.2. Постфикс

Вредноста на променливата се зголемува откако ќе се пресмета изразот

```
a = b++;
```

## 1.6. 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;
```

## 2. Примери

### 2.1. Работа со променливи и доделување вредности

```
#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; */
    p = 3.5 / 2.8; /* p = 1.25 */
    a = p; /* a = 1 */
    a = a + 1; /* a = 2; */
    return 0;
}
```

## 2.2. Промена на вид на вредност – cast операција

### Формат

(podatocen tip) vrednost

### Пример

```
int i;
double d = 7.28;
i = (int) d;
```

### Пример

```
(int) 5.56          // 5.56 во 5
(long) 8.28         // 8.28 во 8L
(double) 2          // 2 во 2.0
(char) 70           // 70 во char чиј код е 70 ('F')
(unsigned short) 3.14 //3.14 во 3 (unsigned short)
```

## 2.3. Претворање во друг тип

Манипулирање со спецификатор за конверзија **%f** и со оператор за делење

```
#include <stdio.h>
int main() {
    int integer1; /* првиот број што кје го внесува корисникот */
    int integer2; /* вториот број што кје го внесува корисникот */
    int sum; /* променлива во која кје се зачува сумата */
    float kol; /* променлива во која кје се зачува коликиот */

    printf("Vnesi prv cel broj\n");
    scanf("%d", &integer1);

    printf("Vnesi vtor cel broj\n"); /* procitaj cel broj */
    scanf("%d", &integer2); /* procitaj cel broj */

    sum = integer1 + integer2; /* dodeli go zbirot na sum */
    kol = (float) integer1 / integer2; /* dodeli go kolicnikot na kol */

    printf("Nivnata suma e %d\n", sum); /* pecati sum */
    printf("Nivniot kolicnik e %.2f\n", kol); /* pecati kol */

    return 0;
}
```

## 2.4. Кастирање

- Во претходниот пример беше искористен каст оператор (оператор кој овозможува преминување од еден во друг тип):

```
kol = (float) integer1 / integer2;
```

- Бидејќи `integer1` и `integer2` се цели броеви, резултатот кој ќе се добиеше по целобројното делење не ќе беше она што го очекувавме. Значи, барем една од тие вредности на `integer1` или `integer2` требаше некако да се претвори во реален број. Тоа ни го овозможи операторот `(float)`, кој `integer1` го претвори во реален број, а со тоа и целото делење се претвори во делење на реални броеви.
- Истиот ефект на кастирање на `int` во `double` може да се постигне и со множење на променливата со `double` константа `1.0`

```
kol = 1.0 * integer1 / integer2;
```

## 3. Задачи

### 3.1. Задача 1

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.



Користете логички и релациони оператори за тестирање на ASCII вредноста на знакот.

- Бонус: Направете проверка дали знакот е цифра

#### Решение 1

```
#include <stdio.h>

int main() {
    char ch;
    int rez;
    printf("Enter char: ");
    scanf("%c", &ch);
    rez = (ch >= 'a') && (ch <= 'z');
    printf("%d\n", rez);
    return 0;
}
```

#### Решение на бонусот

```
rez = (ch >= '0') && (ch <= '9');
```

## 3.2. Задача 2

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 2

```
#include <stdio.h>

int main() {
    int x, y, z;
    printf("Enter x and y: ");
    scanf("%d%d", &x, &y);
    z = x++ + --y + (x < y);
    printf("z = %d\n", z);
    return 0;
}
```

*Што ќе се случи во оваа ситуација:*

```
z = x++ + --y + x < y;
```



### 3.3. Задача 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?

- 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 3

```
#include <stdio.h>

int main() {
    int x = 1, y = 2, z = 3, r;
    r = (x < y || y < z++);
    printf("r = %d, z = %d\n", r, z);

    r = (x > y && y < z++);
    printf("r = %d, z = %d\n", r, z);

    return 0;
}
```

#### Излез

```
r=1, z=3
```

```
r=0, z=3
```

### 3.4. Задача 4

- Нека е дадено:

```
#include <stdio.h>

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

Каква вредност ќе има y за x=5?

*Излез*

y=1

- Нека е дадено:

```
#include <stdio.h>

int main() {
    int x, y, z;
    z = scanf("%d%d", &x, &y);
    printf("z = %d\n", z);
    return 0;
}
```

Каква вредност ќе има z за x=5, y=6?

*Излез*

z=2

### 3.5. Задача 5

Да се напише програма каде од тастатура ќе се внесе цена на производ, а потоа ќе ја испечати неговата цена со пресметан ддв.



ДДВ е 18% од почетната цена

*Решение 5*

```
#include <stdio.h>

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

### 3.6. Задача 6

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



Пресметајте ја целата сума, па потоа ратата.

#### Решение 6

```
#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 kje iznesuva: %.3f\n", rata);
    printf("Vкупната isplatena suma ke bide %.3f\n", vkupno);
    return 0;
}
```

### 3.7. Задача 7

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:

Najznacajna cifra e 7, a najmalku znacajna e 5.



Искористете целобројно делење и остаток од делење.

#### Решение 7

```
#include <stdio.h>

int main() {
    int broj;
    printf("Vnesete go brojot:\n");
    scanf("%d", &broj);
    printf("Najznacajната cifra e ");
    printf("%d, a najmalku znacajната e %d.\n", (broj / 100),
        (broj % 10));
    return 0;
}
```

### 3.8. Задача 8

Да се напише програма каде од тастатура ќе се внесе датумот на раѓање во формат (ddmmgggg). Програмата на компјутерскиот екран ќе го испечати денот

и месецот на раѓање.

*Пример:*

Ако се внесе следниот број 18091992, програмата ќе испечати: 18.09



Искористете целобројно делење и остаток од делење.

*Решение 8*

```
#include <stdio.h>
int main() {
    long int datum;
    int den, mesec;
    printf("Vnesete datum na ragjanje:\n");
    scanf("%ld", &datum);
    den = datum / 1000000;
    mesec = (datum / 10000) % 100;
    printf("Denot i mesecot na ragjanje se %02d.%02d\n", den, mesec);
    return 0;
}
```

## 4. Изворен код од примери и задачи

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

Source code ZIP