# Matura iz informatike

## Spomladanski rok 2013, pola 2

**NALOGA 1**

V elektronskih dokumentih uporabljamo namesto fizičnega podpisa digitalni podpis.

1.1 Kaj pomeni verodostojnost podpisane listine?

_____

_____

1.2 Digitalni podpis določata _____ in _____ ključ.

1.3 Kako deluje digitalno podpisovanje?

_____

_____

_____

_____

1.4 Ali je digitalni podpis enakovreden lastnoročnemu podpisu? Pojasnite svoj odgovor.

_____

_____

**Rešitev:**

1.1 Verodostojnost podpisane listine zagotavlja, da je taka, kot smo jo podpisali, in da smo se strinjali z njeno vsebino. *(Podpis 1 točka, strinjanje 1 točka)*

1.2 Javni, zasebni

1.3 Avtor izračuna izvleček sporočila, ki ga podpiše s svojim zasebnim ključem (izvleček šifrira). Prejemnik izračuna izvleček sporočila in ga primerja z odšifriranim izvlečkom avtorja. Odšifrira ga z avtorjevim javnim ključem. *(Upoštevajo se tudi drugačni pravilni odgovori; opis podpisovanja 2 točki, opis preverjanja 2 točki)*

1.4 Da. Slovenska zakonodaja enači oba podpisa. *(Upoštevajo se tudi drugačni pravilni odgovori; odgovor 1 točka, pravilna razlaga 1 točka)*

# Matura iz informatike

## Spomladanski rok 2013, pola 2

**ACM skupina:**

- NC. Net Centric Computing (NC/NetworkSecurity) – Omrežno računalništvo
- *AL. Algorithms and Complexity (AL/CryptographicAlgorithms) – Algoritmi in zahtevnost*
- DS. Discrete Structures (DS/???) – Diskretne strukture

**Razlaga:**

Naloga zahteva poznavanje omrežne varnosti, digitalnega podpisovanja ter osnov kriptografije in kriptografskih algoritmov.

Osnovna matematična koncepta, ki ju srečamo v tej nalogi sta: javno-zasebni par ključev in kriptografska razpršilna funkcija. Pri javno-zasebnem paru ključev gre za preprost postopek uporabe primerne funkcije (npr. RSA, eliptične krivulje ipd.), ki ima dva parametra: podatke (dejansko neka zelooooo velika številka), ki jih želimo zakriptirati; in parameter (ponovno samo število), s pomočjo katerega izvedemo kriptiranje. Iz zornega kota uporabe, je funkcija samo matematična funkcija z dvema parametroma.

Drugemu parametru rečemo tudi ključ. Pri asimetričnih ključih velja, da lahko podatke, ki smo jih zakriptirali z enim ključem, z drugim ključem, ki je različen od prvega, odkriptiramo. Pri tem velja, da če nekdo pozna prvi ključe zelooooo težko naračuna drugi ključ. Matematično gledano sta oba ključa enaka, le kako ravnamo z njimi jima da svoje ime: javni in zasebni ključ. Posledica vsega povedanega je, da praktično nekaj, kar je zakriptiranega z zasebnim ključem lahko odkriptiramo samo z javnim ključem; ali drugače, če smo nekaj odkriptirali z javnim ključem je **moralo biti zakriptirano** z zasebnim ključem. Slednje dejstvo se uporablja pri postopku podpisovanja dokumentov, kot ga opisuje naloga.

Za zagotovitev verodostojnosti listine bi bilo torej dovolj, da bi s svojim zasebnim ključem zakriptirali listino in bi lahko vsakdo preveril, da smo jo res mi zakriptirali tako, da bi uporabil naš javni ključ (glej poudarjeno besedilo zgoraj). Če bi nepridiprav sedaj nekaj popravljal v dokumentu, ga ne bi mogel zakriptirati nazaj, saj ne pozna našega zasebnega ključa.

Celotno rokovanje z dokumentom je malce nerodno. Namreč naša želja ni skrivati vsebine in tako nekdo, ki bi rad prebral dokument, mora najprej dokument odkriptirati ter ga lahko šele nato prebere. V stvarnem (papirnem svetu) to rešujemo na dva načina: če dokument sestoji iz več listov, jih povežemo z vrvico, ki jo zapečatimo ter lahko vsakdo vidi, če je pečat zlomljen in morda dokument potvorjen; ali številčimo strani tako, da napišemo vedno poleg številke strani še število vseh strani **in** vsak list podpišemo (zakaj oboje?). V digitalnem svetu se zatečemo h kriptografskim razpršilnim funkcijam (MD5, SHA1, SHA256, ...).

Slednje iz dokumenta naračunajo krajši podatek (število), ki mu rečemo izvleček, pri čemer velja, da je težkoooo naračunati drug dokument, ki bi imel isti izvleček; da vsak popravek dokumenta spremeni izvleček in še nekaj lastnosti. Sedaj namesto celotnega dokumenta podpišemo (zakriptiramo s svojim zasebnim ključem) samo izvleček. Če nekdo želi preveriti verodostojnost

dokumenta, mora: i.) sam naračunati izvleček; ii.) odkriptirati zakriptirani izvleček z našim javnim ključem; in iii.) preveriti, če se oba izvlečka ujemata.

V celotnem postopku je še nekaj podrobnosti, a ena najpomembnejših je, da moramo mi verjeti, da je javni ključ v koraku ii.) v resnici javni ključ podpisovalca. Slednje preverimo tako, da o tem povprašamo nekoga, komur zaupamo. Tehnično to pomeni, da neka institucija potrdi (certificira), da je nek ključ (dejansko neko število) res javni ključ neke osebe. To potrdi tako, da napiše dokument (certifikat, glej standard X500), v katerem potrjuje dejstvo in ga podpiše s svojim zasebnim ključem (opis podpisovanja je identičen zgornjemu). Kot uporabnik moramo sedaj poznati javni ključ potrjevalske agencije – ustvari se veriga zaupanja.

Tehnično gledano, javni ključ agencije (npr. SI-CERT) ročno vnesemo v naš računalnik.

# Matura iz informatike

## Spomladanski rok 2013, pola 2

**NALOGA 2**

Informacijski sistem.

2.1 Napišite, kaj je informacijski sistem.

_____

_____

2.2 Napišite cilje informacijskega sistema.

_____

_____

2.3 Napišite sestavine (elemente) informacijskega sistema.

_____

_____

_____

_____

2.4 Navedite tri primere informacijskih sistemov z različnih področij vsakdanjega življenja.

_____

_____

**Rešitev:**

2.1 Je skupek ljudi, podatkov, postopkov in naprav, zasnovan za zbiranje, obdelavo, shranjevanje in pošiljanje podatkov. *(Upoštevajo se tudi drugačni pravilni odgovori)*

2.2 Predložiti uporabnikom prave podatke ob pravem času. *(Upoštevajo se tudi drugačni pravilni odgovori)*

2.3 Strojna oprema, programska oprema, podatkovna baza, omrežje, postopki, ljudje *(Število pravilnih odgovorov – 2 = število točk)*

2.4 Bolnišnični IS, bančni IS, šolska evidenca, knjižnični IS *(Upoštevajo se tudi drugačni pravilni odgovori; trije pravilni primeri 2 točki, dva pravilna primera 1 točka)*

# Matura iz informatike

## Spomladanski rok 2013, pola 2

**ACM skupina:**

- IM. Information Management (IM/InformationModels) – Upravljanje informacij

**Razlaga:**

Naloga zahteva poznavanje informacijskih sistemov in njihovega delovanja. Sem sodi tudi shranjevanje, urejanje, priklic, predstavitev informacij ter poznavanje aplikacij oz. programov za upravljanje (management) informacij.

V RIN pogosto nastopajo sistemi: računalnik kot sistem, operacijski sistem, porazdeljeni sistem, informacijski sistem itd. Vsi ti sistemi so dejansko enaki sistemom, katere srečamo tudi drugje; npr. v elektrotehniki, v družboslovju itd. V vseh sistemih imamo opravka z dejavniki (akterji); snovjo/stvarmi, s katero se sistem ukvarja; pravili in postopki, po katerih sistem deluje; in zunanjimi dejavniki. Tudi pri informacijskih sistemih ni nič drugače, le da so dejavniki ljudje ter strojna in programska oprema in stvarmi, s katerimi se ukvarjajo, podatki/informacije. Seveda so informacijski sistemi lahko nato specializirani za posamezno področje.

# Matura iz informatike

## Spomladanski rok 2013, pola 2

**NALOGA 3**

Jure je napisal svojo prvo spletno stran. Želi, da je vidna na spletu, zato jo bo prenesel na spletni strežnik z operacijskim sistemom Linux.

V tabelo zapišite napake, poleg napak pa pravilen zapis.

<HTML><HEAD><TITLE> Moja spletna stran </TITLE>

<BODY></HEAD>

<H1><I> Pozdravljeni! </I></H2>

<FONT  SIZE="4" COLOR="#FF">

<IMG SRC="c:/moje_slike/tulipan.jpg">

<I><U> To je moje besedilo!</U></I></P>

</BODY>

</HTML>

| Napaka | Pravilen zapis |
|---|---|
| | |
| | |
| | |
| | |
| | |

**Rešitev:**

| Napaka | Pravilen zapis |
|---|---|
| <BODY></HEAD> | </HEAD><BODY> |
| </H2> | </H1> |

# Matura iz informatike

## Spomladanski rok 2013, pola 2

| | |
|---|---|
| Ni značke <P> | dodan <P> ali  izbris </P> |
| "c:/moje_slike/tulipan.jpg" | "moje_slike/tulipan.jpg" |
| color="#FF" | color="#FF0000" |

**ACM skupina:**

NC. Net Centric Computing (NC/WebOrganisation) – Omrežno računalništvo

**Razlaga:**

Naloga zahteva poznavanje strukture zapisa v jeziku HTML.

Pri definiranju jezika HTML bi lahko načrtovalci sestavili poljubno zahtevno slovnico. Odločili so se za kar se da preprosto slovnico, ker je potem razpoznava besed iz takšnega jezika preprostejša in računsko manj zahtevna. V preprostejših besedah; beseda v jeziku HTML je vsaka spletna stran in brskalnik mora besedo razpoznati ter pravilno predstaviti – pokazati spletno stran. Tu postaja jasno, zakaj mora biti slovnica preprosta, saj bi sicer naš brskalnik predolgo preračunaval vsebino, predno bi jo prikazal.

Ena osnovnih odločitev je bila, da bo jezik HTML zasnovan na osnovi značk, ki jih odpiramo in zapiramo (prim. oklepaje), kot je bil pred tem zasnovan že jezik SGML. Tri od zgornjih napak izvirajo iz napačne rabe odpiranja in zapiranja značk. Ob tem moramo dodati, da so načrtovalci pri nekaterih značkah dovolili opuščanje zapiranja, čeprav je zaradi tega možna dvoumnost. Na primer: <P> nekaj <P> drugo </P>.  Podobna dvoumnost nastopa v programskih jezikih: IF pogoj1 THEN IF pogoj2 THEN nekaj1 ELSE nekaj2 ENDIF, kjer ne vemo, kam sodi ELSE pri nekaj2. Poiščite načine razreševanja te dvoumnosti.

Četrta napaka izhaja iz dejstva, da je lokalni datotečni sistem pri različnih operacijskih sistemih različen – prim. *File system hierarchy*. Zadnja, peta napaka je ponovno plod jezika HTML, saj zahteva zapis barv v 24-bitni obliki ter je zapis FF dvoumen.

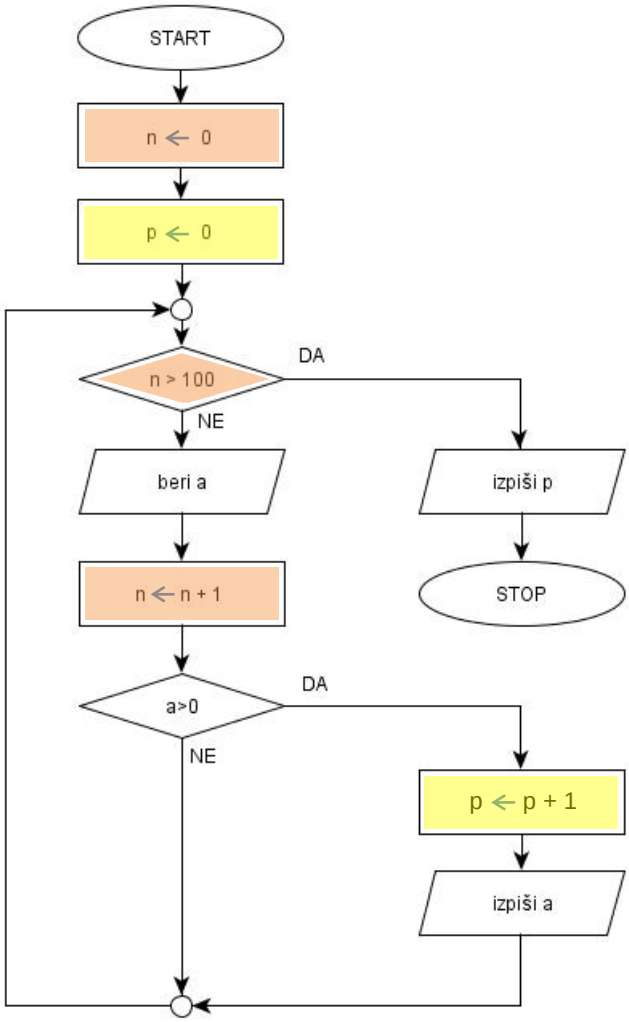
# Matura iz informatike

## Spomladanski rok 2013, pola 2

**NALOGA 4**

Sestavite diagram poteka za postopek, ki omogoča vnos 100 številskih podatkov.
Med vnosom izpiše vsa pozitivna števila.
Na koncu izpiše še število izpisanih števil.

**Rešitev:**




**ACM skupina:**

- PF. Programming Fundamentals(PF/FundamentalProgrammingConcepts) – Osnove programiranja

# Matura iz informatike

## Spomladanski rok 2013, pola 2

**Razlaga:**

Naloga zahteva implementacijo naslednjih osnovnih programskih konceptov:

- 100 kratno izvajanje zanke: za slednjo potrebujemo najprej spremenljivko, v kateri beležimo, kolikokrat se je zanka že izvedla. V naši rešitvi ima spremenljivka ime n. Spremenljivko na začetku nastavimo na vrednost 0, ker se ni zanka še nikoli izvedla in ob vsakem izvajanju zanke jo povečamo za 1 – pozor!! n šele po povečanju predstavlja število ponovitev zanke. Preverjanje ali smo zanko že dovoljkrat izvedli opravi pogojna vejitev n >= 100. Ob tem bi želeli poudariti, da bi semantično pravilen bil pogoj ali je n = 100, kar pomeni, da se je zanka stotič izvedla. Vendar je razširjeno preverjanje >= običajno.
- Štetje in izpisovanje pozitivnih prebranih števil ter na koncu izpis njihovega skupnega števila: to nalogo razdelimo na dva dela: štetje pozitivnih števil in oba končna izpisa. Za štetje števil ponovno potrebujemo spremenljivko (podobno kot za štetje ponovitev zanke), ki jo imenujemo v rešitvi p ter jo na začetku nastavimo na 0 – saj nismo prebrali še nobenega pozitivnega števila. Pri branju števil, ki jih beremo v spremenljivko a, sproti preverjamo, če je število pozitivno ter, če je, povečamo števec pozitivnih števil p za 1. Kot zahteva naloga, najdeno pozitivno število tudi izpišemo ter ob zaključku izpišemo število najdenih pozitivnih števil.

Preprosta razširitev: izpiši število ne pozitivnih števil. Malce bolj zapletena razširitev: izpiši spektrum števil: število števil med 1 in 100, med 101 in 200, ... do 1000.

# Matura iz informatike

## Spomladanski rok 2013, pola 2

**NALOGA 5**

Knjižničarka je za posamezne dijake izpisala vse gradivo, ki ga imajo še izposojenega.

**IZPOSOJA**

Knjižnica: Sanje
Naslov: Sanjska cesta 100 Kraj: 1000 Ljubljana

Datum: 17. 11. 2011

Št. obiskovalca: 87
Ime: Janez Novak Naslov: Cigaletova 1 Kraj: 1215 Medvode
Skupina: dijak Član od: 1. 9. 2011
Ima izposojeno enot: 3 Članarina: 10,00 €

Gradivo št.: 8 KOMAC, Darko Zvrst: kriminalke
Vrsta gradiva: m (monografija) Trije so se potepali
ISSN: 886.3-312.4 Ljubljana, 1991

| Št. gradiva | Št. avtorja | Avtor | Naslov | Izposojeno dne | Vrniti | Zamuda |
|---|---|---|---|---|---|---|
| 8 | 9 | KOMAC, Darko | Trije so se potepali | 2. 11. 11 | 23. 11. 11 | |
| 18366 | 410 | NERY, Gerard | Bolečina ljubezni | 10. 10. 11 | 31. 10. 11 | 17 |
| 25965 | 233 | SMRDU, Andrej | Fluor in flour, zbirka nalog | 5. 9. 11 | 26. 9. 11 | 52 |

5.1. Določite entitete in njim ustrezne atribute, in sicer na podlagi izpiska izposoje.

**Rešitev:**

Gradivo (ID gradiva, naslov, ID avtorja, ISSN, vrsta, zvrst, kraj, leto izdaje)
Obiskovalec (ID obiskovalca, priimek, ime, skupina, kraj, ulica, datum včlanitve, članarina)
Izposoja (ID gradiva, ID avtorja, datum izposoje)
Avtor (ID avtorja, priimek, ime)

5.2. Določite ključe entitet in opišite, za kakšno vrsto ključa (primarni, tuji ali sestavljeni) gre.

**Rešitev:**

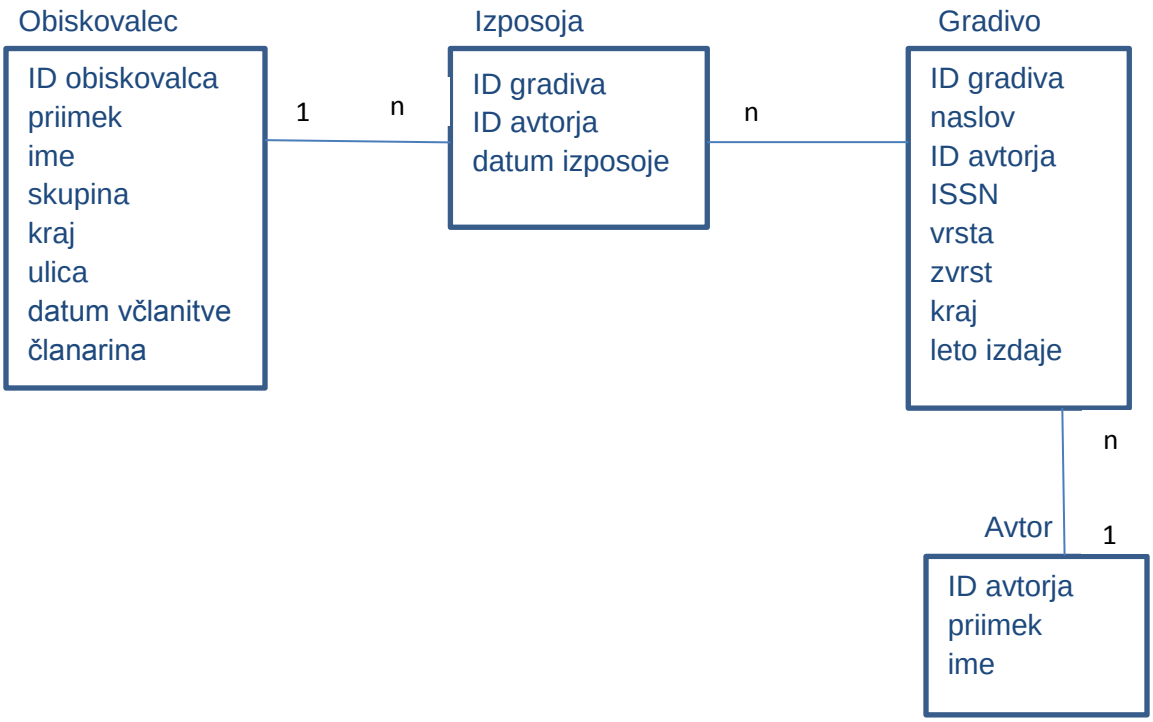
Gradivo (ID gradiva) – primarni ključ
Obiskovalec (ID obiskovalca) –primarni ključ
Izposoja (ID gradiva, ID avtorja, datum izposoje) – sestavljeni ključ
Avtor (ID avtorja) – primarni ključ

# Matura iz informatike

## Spomladanski rok 2013, pola 2

5.3. Narišite diagram E-R, označite relacije med entitetami in določite številnost.

**Rešitev:**




**ACM skupina:**

- IM. Information Management - Upravljanje informacij

**Razlaga:**

Preverja razumevanje pojmov "entiteta", "ključ", "številnost", "relacija" in zahteva, uporabo znanja gradnje modela E-R.

R-E model sestoji iz dveh ključnih sestavin: entitet z lastnostmi in odnosov (relacij) med njimi. Vsebinsko je tej nalogi bilo potrebno najti entitete, določiti njihove lastnosti, določiti, katere lastnosti predstavljajo ključe, ter na koncu entitete povezati v R-E model. Ker sta v konceptualnem modelu bili entiteti *Obiskovalec* in *Gradivo* v odnosu mnogo na mnogo, je bilo potrebno izvesti še normalizacijo z vmesno tabelo *Izposoja*. Dejansko je slednja že bila izpisana v besedilu naloge.

# Matura iz informatike

## Spomladanski rok 2013, pola 2

**NALOGA 6**

V preglednici imamo podatke o reševanju testa in podatke o statistični obdelavi testa.

| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | **Ime in Priimek** | **Št. točk** | **Ocena** | | **Ocena** | **oznaka** | **št.ocen** | **Delež v %** | |
| 2 | Janez Novak | 55 | 2 | | negativno | 1 | | | |
| 3 | Miha Božič | 89 | 5 | | zadostno | 2 | | | |
| 4 | Ana Kovač | 72 | 3 | | dobro | 3 | | | |
| 5 | Jana Nagode | 91 | 5 | | prav dobro | 4 | | | |
| 6 | Bine Horvat | 68 | 3 | | odlično | 5 | | | |
| 7 | Ana Lampe | 80 | 4 | | **skupaj** | | | | |
| 8 | Simon Koren | 74 | 3 | | | | | | |
| 9 | Tone Hočevar | 81 | 4 | | | | | | |
| 10 | **Povprečje** | | | | | | | | |
| 11 | | | | | | | | | |
| 12 | | | | | | | | | |

6.1. Izračunajte povprečno število točk v testu in povprečno oceno tega testa. Zapišite ustrezno funkcijo v celici B10, tako da jo lahko kopirate tudi v celico C10 in bo delovalo pravilno.

**Rešitev:**

AVERAGE(B2:B9)

6.2. Izračunajte število posameznih ocen v testu. Zapišite izraz v celici G2, tako da jo lahko kopirate še v celice G3 do G6 in bo delovalo pravilno. Uporabite funkcijo COUNTIF (COUNTIF(območje; pogoj)), ki vam v izbranem območju prešteje celice, ustrezne pogoju.

**Rešitev:**

COUNTIF($C$2: $C$9;F2)

6.3. V celici G7 zapišite funkcijo, ki izračuna število vseh ocen testa.

**Rešitev:**

SUM(G2:G6)

6.4. Izračunajte deleže ocen. Zapišite izraz v celico H2 tako, da jo lahko kopirate v celice od H3 do H6 in bo delovalo pravilno.

# Matura iz informatike

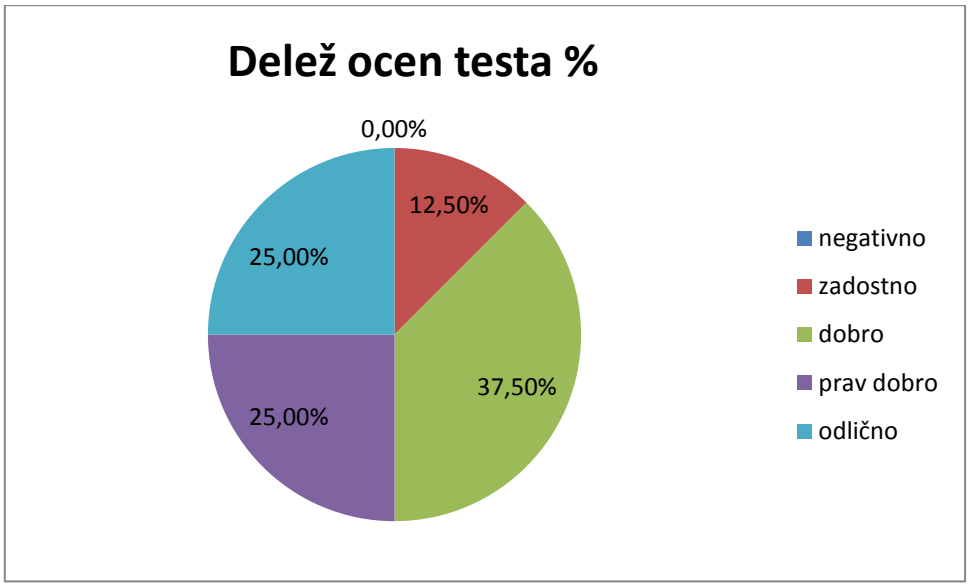
## Spomladanski rok 2013, pola 2

**Rešitev:**

=G2/$G$7

6.5. Skicirajte grafikon, ki predstavlja deleže posameznih ocen. Grafikon opremite z vsemi potrebnimi podatki.

**Rešitev:**




**ACM skupina:**

- *AL. Algorithms and Complexity – Algoritmi in zahtevnost*
- PF. Programming Fundamentals – Osnove programiranja

**Razlaga:**

Naloga zahteva poznavanje razlikovanja med absolutnim in relativnim naslavljanjem ter uporabo le-tega. Poleg tega pričakuje razumevanje in uporabo pojma funkcija, argument, pogoj, naslovi prostor. Kar je zanimivo pri tej nalogi je to, da je to naloga iz programiranja in ne toliko iz uporabe preglednic. Namreč koncept posrednega in absolutnega naslova spremenljivke je eno temeljnih orodij pri programiranju.

Zanimiva je tudi funkcija COUNTIF, ki ima dva parametra: območje, na katerem se izvede štetje; in pogojno funkcijo, ki se odloča za vsak element območja posebej, ali se naj upošteva pri štetju ali ne.

# Matura iz informatike

## Spomladanski rok 2013, pola 2

Opis območja v programskih jezikih ne predstavlja posebnosti, je pa prenos funkcije kot spremenljivke izredno močno orodje. Naj ga pojasnimo na primeru.

Recimo, da imamo funkcijo ODŠTEJ(int x, y), ki od x odšteje y in vrne rezultat. Ko pokličemo funkcijo na primer ODŠTEJ(3, 2), se spremenljivki x priredi vrednost 3 in spremenljivki vrednost 2 ter se opravi preostanek izračuna. V primeru funkcije COUNTIF(območje, pogoj) pa se spremenljivki pogoj priredi vrednost, ki je funkcija. To ima dolgoročne posledice, saj sedaj lahko sedaj spremenljivki vedno priredimo vrednost fukcije in predvsem funkcija kot rezultat lahko vrne novo funkcijo – zato govorimo o funkcijah višjega reda.

Pri sodobnih jezikih, ki zahtevajo ujemanje tipov pri prirejanju vrednosti spremenljivk (typing), slednje zahteva, da tudi definiramo tip spremenljivke, ki je lahko funkcija. Ta je definiran kot n-terica (p1, p2, p3, ..., pn-1, r), kjer so pi tipi parametrov funkcije in r tip rezultata funkcije. Sedaj lahko prirejamo vrednosti samo v primeru, ko se funkcija, ki je prirejena spremenljivki, ujema s spremenljivko.