

Uputstva i pravila za predmet Uvod u programiranje

1 Održavanje ispita – opšta pravila

- Na ispit je potrebno doneti indeks.
- **Ispit traje 3 sata.** Sastoji se od teorijskog dela i praktičnog dela.
- **Teorijski deo** se sastoji iz **15 kratkih pitanja**. Ukupno se radi **45 minuta**. Radi se na papiru. Teorijski deo nosi **45 poena**. Da bi se ispit položio potrebno je imati **bar 50%** poena teorijskog dela.
- **Praktični deo** se radi na računaru i ukupno se radi **2 sata i 15 minuta**. Sastoji se od **5 zadataka** koja se pišu na programskom jeziku C++. Svaki zadatak nosi **11 poena**. Ukupno praktični deo nosi **55 poena**. Da bi se ispit položio potrebno je imati **bar 50%** poena praktičnog dela.
- Da bi se ispit položio potrebno je imati ukupno **bar 51 poen**.
- U prvom ispitnom roku se vrši **skaliranje** prema najboljem studentu. Taj faktor skaliranja se koristi kasnije u svim ispitnim rokovima.
- Kada ispit počne (odnosno kada počne bilo koji deo ispita) student **ne sme da napušta učionicu prvih pola sata**.
- Student je obavezan da preda potpisan papir sa teorijskim delom čak i ako nije popunjen nijedan odgovor.
- Student može poneti zadatke sa praktičnog dela.
- Između praktičnog i teorijskog dela postoji kratka pauza. Za vreme te pauze studenti se mogu osvežiti, a nekada je potrebno i zameniti učionicu. Ipak, student se ne sme zadržati duže nego što mu je rekao dežurni asistent jer ga dežurni neće čekati već će početi sa ispitom bez obzira da li se student vratio nakon pauze ili ne.

2 Tekst zadatka

- Za svaki zadatak, pored teksta zadatka, dati su i test primeri. Najčešće je dato *4 test primera*.
- **Uvek se podrazumeva da sve moguće unose student treba da predvidi i da obezbedi da se program uvek ispravno ponaša.** Student treba sam da predvidi sve moguće pogrešne unose i specijalne slučajeve. U test primerima se mogu uočiti neke pogrešne situacije, ali neće biti istaknute i zapisane sve koje su moguće.
- Ukoliko u zadatku stoji napomena *“Podrazumevati da je ulaz korektno zadat”*, onda (samo tada) nije potrebno obrađivati slučaj pogrešnog unosa podataka.

3 Način pregledanja i bodovanja C++ programa

- Program se automatski testira pregledačem. Test primeri nisu viđeni u postavci zadatka.
- Automatskim testiranjem student može dobiti **9 od 11 poena**.
- Ukoliko program **nije sintaksički** ispravan dobija se **0 poena**.
- Od 9 poena, **većina se dodeljuje za glavnu funkcionalnost programa**. Ostali poeni se dodeljuju za specijalne slučajeve. Specijalni slučajeve nisu pogrešan unos, ali se ponekad moraju obraditi nezavisno/odvojeno od glavnog/opšteg slučaja.
- Test primeri mogu biti **grupisani u pakete** od kojih svaki testira neku specifičnu funkcionalnost i nosi određeni broj poena. **Poeni koje nosi paket dodeljuju se samo ukoliko program korektno radi za sve primere tog paketa**.
- Ukoliko je student dobio 9 poena testiranjem zadatka moguće je dobiti još **2 poena na lepotu koda**. Lepotom koda se ocenjuje, između ostalog:
 - nazubljenost koda
 - jednostavnost rešenja
 - imenovanje promenljivih i funkcija
 - da li su datoteke ispravno zatvorene
 - pravilno čuvanje zadataka
- Na ispitu je neophodno **striktno poštovati formu ispisa koja se zahteva**. Na primer, "... program ispisuje jedan ceo broj...", onda program treba da ispisuje

```
cout << broj;
```

a **pogrešno** bi bilo ispisati:

```
cout << "Resenje je " << broj;
```

Kao što se iz primera može videti, pogrešno je ispisivati ikakav dodatni tekst koji se prema tekstu zadatka ne traži.

Sa druge strane, dozvoljeno je ispisivati praznine, odnosno blanko, tab (`\t`) i znak za novi red (`\n`).

Ukoliko se ne ispoštuje tražena forma, pregledač vraća 0 poena. Nakon žalbi, ukoliko je jedina greška neispravan ispis moguće je dobiti maksimalno 9 poena (u zavisnosti koliko test primera prolazi), ali ne i 2 poena na lepotu koda.

- Ukoliko je zadatak rešen tako što su **podaci učitani na pogrešan način, broj osvojenih poena je 0**. Na primer, ako se čita iz datoteke, a trebalo je sa standardnog ulaza ili ako se čita iz argumenata komandne linije, a trebalo je iz datoteke itd.
- **Neophodno je voditi računa o inicijalizaciji promenljivih**. Na primer, može se dogoditi da izostavi inicijalizacija (recimo $s = 0$), a da se program ispravno ponaša. Ipak, tokom testiranja ova greška će biti uočena i broj osvojenih poena je 0.

- **Važno je program sačuvati na način kako piše na papiru sa zadacima** i kako je naveo dežurni asistent. Ukoliko se to ne učini može se dogoditi da se zadatak izgubi (i samim tim ne može da se oce- ni). Takođe, može se dogoditi da se loše sačuvan program ne testira ispravno (i da se dobije 0 poena). Ukoliko se loše sačuvani program pronađe, moguće je nakon žalbi osvojiti maksimalno 9 poena (ali ne i dodatna 2 poena).

Najčešće se zadaci čuvaju pod svojim rednim brojem (1.c, 2.c, itd..) u folderu čiji naziv je:

InicijaliAsistenta_ImePrezime_brojIndeksa_Grupa

Na primer, AS_MihajloPupin_mi16235_1

mi16235 – je oznaka indeksa; mi je oznaka i-smera, ai je oznaka za astroinformatiku itd. 16 označava godinu upisa, a 235 označava broj indeksa. Ako je indeks recimo 9, godina upisa 2015, a smer astroin- formatika, onda je oznaka ai15009.

- Ukoliko je student ostavio **samo izvršivu verziju programa**, ali ne i kod, broj osvojenih **poena je 0**.
- **Ukoliko student napravi rešenje u kome su izlazi ispravni, ali ne na način na koji se traži u zadatku broj osvojenih poena je 0**. Na primer, ako se u zadatku traži da se napravi funkcija koja nešto radi, a takve funkcije u kodu nema, onda je broj poena 0. Ili, ako u zadatku piše da treba rešiti korišćenjem niza ili matrice itd..., a to nije urađeno onda je broj osvojenih poena 0.
- **U posebnim slučajevima, asistent može dodeliti 0 poena za neki zadatak ukoliko ličnim uvidom ustanovi da je, iako neispravan, automatskim testiranjem dobio neke poene**. Pri- mera radi, ukoliko program koji treba da ispiše odgovor „da” ili „ne” ignoriše ulaz i uvek ispisuje samo „da”, asistent će za taj program dodeliti 0 poena.