



Šifra kandidata:

Državni izpitni center



M 2 5 1 4 5 1 1 2

SPOMLADANSKI IZPITNI ROK

INFORMATIKA

Izpitna pola 2

Četrtek, 12. junij 2025 / 90 minut

Dovoljeno gradivo in pripomočki:

Kandidat prinese nalivno pero ali kemični svinčnik in računalo.

Konceptni list je na perforiranem listu, ki ga kandidat pazljivo iztrga.

SPLOŠNA MATURA

NAVODILA KANDIDATU

Pazljivo preberite ta navodila.

Ne odpirajte izpitne pole in ne začenjajte reševati nalog, dokler vam nadzorni učitelj tega ne dovoli.

Prilepite kodo oziroma vpišite svojo šifro (v okvirček desno zgoraj).

Izpitna pola vsebuje 6 nalog. Število točk, ki jih lahko dosežete, je 44. Za posamezno nalogo je število točk navedeno v izpitni poli.

Rešitve pišite z nalivnim peresom ali s kemičnim svinčnikom v izpitno polo v za to predvideni prostor **znotraj okvirja**. Kadar je smiselno, narišite skico, čeprav je naloga ne zahteva, saj vam bo morda pomagala k pravilni rešitvi. Pišite čitljivo. Če se zmotite, napisano prečrtajte in rešitev zapišite na novo. Nečitljivi zapisi in nejasni popravki bodo ocenjeni z 0 točkami. Osnutki rešitev, ki jih lahko napišete na konceptni list, se pri ocenjevanju ne upoštevajo.

Zaupajte vase in v svoje zmožnosti. Želimo vam veliko uspeha.

Ta pola ima 20 strani, od tega 3 prazne.

V sivo polje ne pišite. V sivo polje ne pišite. V sivo polje ne pišite. V sivo polje ne pišite. V sivo polje ne pišite.

L



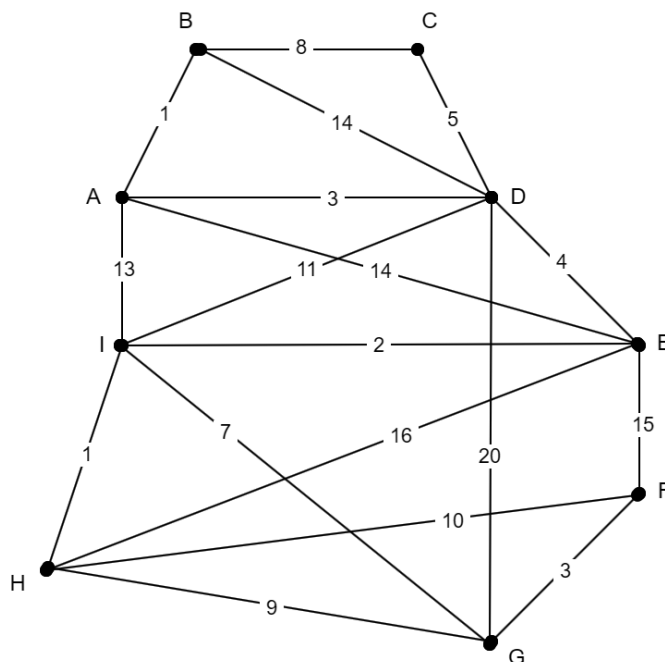
**Konceptni list**

**OBRNITE LIST.**

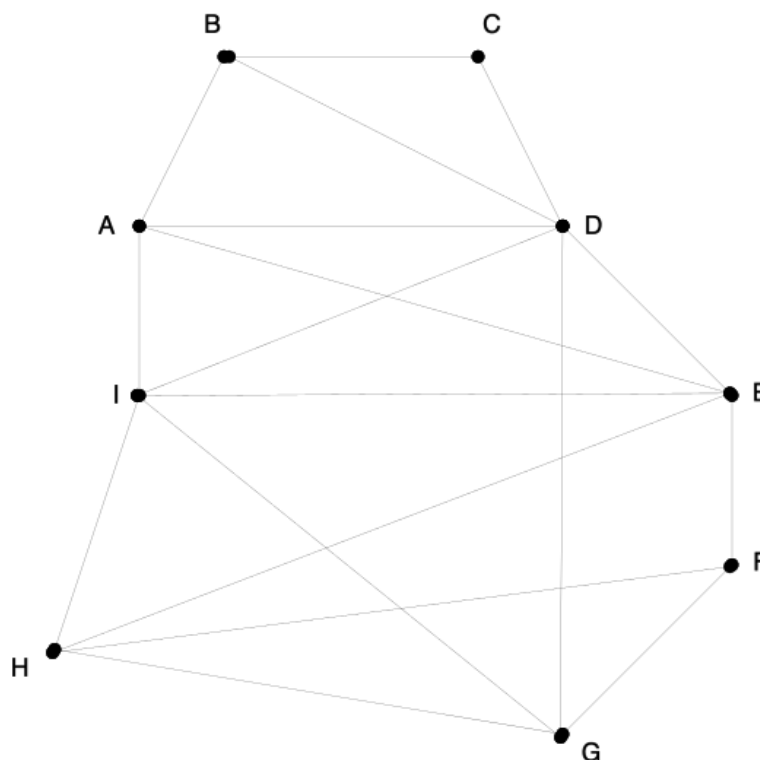


M 2 5 1 4 5 1 1 2 0 6

1. Butalci želijo zgraditi optično omrežje, v katerega bi povezali vse hiše v Butalah. Shema omrežja z možnimi povezavami med posameznimi hišami in ceno izgradnje povezave je prikazana spodaj.



- 1.1. Pomagajte Butalcem pri načrtovanju optičnega omrežja na podlagi možnih povezav na zgornji sliki. Gradnja omrežja naj bo čim cenejša, hkrati pa naj omogoča pošiljanje sporočil med vsemi hišami v Butalah. Narišite to omrežje.



(4 točke)



(1 točka)



2. Na spletni strani želimo imeti prijavi obrazec, kot je prikazan spodaj:

Prijavite se

E-poštni naslov
Geslo

☐ Zapomni si me

Prijava

- 2.1. Dopolnite manjkajoče dele kode HTML.

```
<_____ action="/login" method="post">
  <h1 class="h3 mb-3 fw-normal">Prijavite se<_____>

  <div class="form-floating">
    <_____ type="email" class="form-control"
      id="eposta" placeholder="name@example.com">
    <label for="eposta">E-poštni naslov</label>
  </div>
  <div class="form-floating">
    <input type="password" class="form-control" id="geslo"
      placeholder="Geslo">
    <label for="geslo">Geslo</label>
  <___div>

  <div class="form-check text-start my-3">
    <input class="form-check-input" type="checkbox"
      value="zapomni-si-me" id="zapomni">
    <label class="form-check-label" for="zapomni">
      Zapomni si me
    </label>
  </div>
  <_____ class="btn btn-primary w-100 py-2"
    type="submit">Prijava</button>
</form>
```

(5 točk)



Prazna stran

OBRNITE LIST.



3. Podjetje Spletko, d. o. o., ki se ukvarja s spletno prodajo, je v zadnjem času doživelo skokovito povečanje prometa. Zato želijo posodobiti informacijski sistem, da bi tako izboljšali učinkovitost poslovanja in izkušnjo svojih strank. Trenutno uporabljajo program za preglednice in v njem zavihek, ki vsebuje vse informacije o uporabnikih, naslovih in naročilih, kot je prikazano na sliki.

UporabnikID	Ime	Priimek	Email	Ulica	Mesto	PoštnaŠtevilka	Država	DatumNaročila	Znesek
1	Tina	Novak	tina.novak@example.com	Glavna 10	Ljubljana	1000	Slovenija	10.01.2024	150.00
2	Marko	Zupan	marko.zupan@example.com	Cesta svobode 15	Maribor	2000	Slovenija	14.02.2024	75.00
3	Ana	Mrak	ana.mrak@example.com	Trg republike 5	Celje	3000	Slovenija	22.03.2024	200.00
4	Luka	Hribar	luka.hribar@example.com	Obala 12	Koper	6000	Slovenija	5.04.2024	50.00
5	Maja	Koren	maja.koren@example.com	Poljanska 8	Novo mesto	8000	Slovenija	7.05.2024	120.00
6	Marko	Zupan	marko.zupan@example.com	Cesta svobode 15	Maribor	2000	Slovenija	28.05.2024	140.00

Informatik Miha se je odločil, da bo sistem nadgradil. Najprej načrtuje razdelitev v več tabel in na koncu za vse skupaj uporabo relacijske podatkovne baze.

3.1. Navedite in utemeljite dva razloga za takšno preoblikovanje.

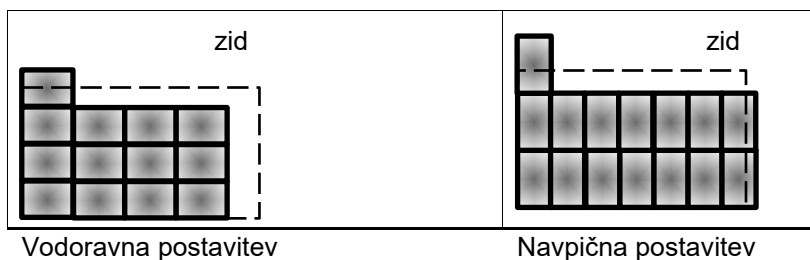
(2 točki)



- (3 točke)



4. Pečar Luka Kratkohlačnica polaga ploščice velikosti 3×2 enoti. Polaga jih v vodoravni ali navpični postavitvi (glejte sliko). Pri polaganju mora pogosto zadnjo ploščico v vrsti odrezati. Ker pa rez praviloma ni najboljši, ostanka ploščice ne more uporabiti. Da bi si olajšal izdelavo ponudb, kjer mora vedeti, koliko ploščic bo potreboval, je začel pisati preprost program.



Luka poda programu višino in dolžino zidu v istih enotah, kot so dimenzije ploščic, program pa mu izpiše, koliko celih ploščic potrebuje za vodoravno in koliko za navpično postavitev:

```
luka> python3 ploscic
Višina zidu > 9
Dolžina zidu > 8
Število ploščic v vodoravni postavitvi: 15
Število ploščic v navpični postavitvi: 12
luka> python3 ploscic
Višina zidu > 8
Dolžina zidu > 9
Število ploščic v vodoravni postavitvi: 12
Število ploščic v navpični postavitvi: 15
luka>
```

Luka je za pisanje programa uporabil jezik Python. Spodnja slika prikazuje izpis izvirne kode programa, ki ga je Luka odložil na mizo. Žal je nanj položil ploščico, ki je na izpisu prekrila funkcijo `CelePloscice`. Funkcija vrne, koliko celih ploščic je potrebnih v eni vrsti oziroma stolpcu. Da bi bila mera polna, je bil preširok še izpis programa in manjkal je del stavka `print`.

```

plo_a = 3          # velikost ploščice ...
plo_b = 2          # ... a x b

def ...):
    ...
    # Cel ...

# preberemo velikost zidu
zid_vis = int(input('Višina zidu > '))
zid_dol = int(input('Dolžina zidu > '))

print ('Število ploščic v vodoravni postavitvi: ', CelePloscice(zid_vis, zid_dol, plo_a, plo_b))
print ('Število ploščic v navpični postavitvi: ', CelePloscice(zid_dol, zid_vis, plo_a, plo_b))

```



4.1. Napišite prekrito funkcijo `CelePloscice` in dopolnite manjkajoči del stavka `print`.

(4 točke)

4.2. Luka je program s pridom uporabljal pri poslu, toda kmalu je ugotovil, da so njegovi predračuni nenatančni in predvsem prenizki. Po preudarku je ugotovil, da je rezanje ploščic drag postopek, zato bi moral pri pripravi predračuna upoštevati še število ploščic, ki jih mora prerezati. Zato si je za najpogostejšo ponujeno vodoravno postavitve med letnim dopustom dodelal program tako, da mu je poleg potrebnega števila ploščic izpisal še število ploščic, ki jih bo moral prerezati:

```
Luka> python3 ploscice
Višina zidu > 9
Dolžina zidu > 8
Potrebno število ploščic: 15
Število rezanih ploščic: 7
Luka > python3 ploscice
Višina zidu > 8
Dolžina zidu > 9
Potrebno število ploščic: 12
Število rezanih ploščic: 0
Luka >
```

Napišite funkcijo `RezanihPloscic`, ki vrne število ploščic, ki jih mora Luka prerezati. Prva parametra sta dolžina in višina zidu ter drugi par parametrov velikost ploščice.

```
def RezanihPloscic(zid_dol, zid_vis, plo_a, plo_b):
```

(5 točk)



5. Butale so zgledna srenja in prav ponosno povedo, da so pri njih vse ceste dvosmerne. Vse prepogosto se je dogajalo, da se ta ali oni Butalec preprosto ni znašel v labirintu enosmernih cest. Naslednji korak je bil, da so izdelali register cest. V ta namen so najprej oštevilčili vsa križišča, ki niso bila samo križišča, ampak tudi krožišča ali kakršnakoli razpotja, s številkami od 1 pa tja do n . Nakar so oštevilčili od 1 do m še vse ceste, ki tudi niso bile samo ceste, ampak še ulice, kolovozi ali kakršnekoli gazi. Sledila je potreba po informacijski podpori registru, ki bo omogočala učinkovite poizvedbe o tem:
- kateri križišči povezuje dana cesta in
 - katere ceste so priključene na križišče.
- 5.1. Prva butalska ideja za informacijsko rešitev je bila, da shranijo cestni register v relacijsko podatkovno bazo. Predlagajte model z entitetami in atributi ter glavnimi in tujimi ključi, ki bo omogočal zahtevani poizvedbi.

(3 točke)



- 5.2. Na koncu so se Butalci odločili, da poskusijo še s povsem lastno rešitvijo, za kar so naprosilil Petra Zmedo. Peter je pripravil knjižnico `registerCest.py` s štirimi funkcijami. Prvi sta `steviloKrizisc()` in `steviloCest()`, ki vrmeta število križišč (zgoraj n) in število cest (zgoraj m) v registru. Poleg tega je dodal funkcijo `PovezujeKrizisci(idCeste)`, ki vrne tabelo z dvema elementoma, ki sta številki križišč, ki ju povezuje cesta s številko `idCeste`. Zadnja funkcija v knjižnici je `PovezujeCeste(idKrizisca)`, ki vrne tabelo cest, ki izhajajo iz križišča `idKrizisca`. Dodal je še primer uporabe knjižnice:

```
from registerCest import *
n= steviloKrizisc()
m= steviloCest()
print("v registru je ", n, " križišč in", m, " cest.")
for idCeste in range(m): #izpišemo vse ceste in kateri križišči povezujejo
    krizisci= PovezujeKrizisci(idCeste)
    print("cesta ", idCeste, " povezuje križišči ",
          krizisci[0], " in ", krizisci[1])
for idKrizisca in range(n): #izpišemo vsa križišča in katere ceste se
                           srečujejo v križišču
    ceste= PovezujeCeste(idKrizisca)
    print("V križišču ", idKrizisca, " se sreča ",
          len(ceste), " cest.")
```

Velikost križišča definiramo kot število cest, ki se stikajo v njem. Z uporabo Petrove knjižnice napišite program, ki poišče največji križišči in ju vrne kot rezultat. Če boste poiskali samo največje križišče, boste dobili nekaj točk.

```
def NajvecjiKrizisci():
```

(4 točke)



- 5.3. Toda Butalci so kmalu ugotovili, da je s Petrovo knjižnico nekaj narobe. Namreč, ponosni kot so bili na dvosmernost svojih cest, so želeli preveriti, če je cesta, ki je zabeležena, da povezuje križišči p in q , v resnici tudi zavedena pri obeh križiščih med cestami, ki jih povezujeta križišči. Napišite program, ki bo izpisal takšne ceste – to je ceste, ki naj bi povezovale križišči p in q , niso pa navedene pri enem od križišč p ali q .

(3 točke)



6. Pomlad je v butalski šoli poskrbela za prve simpatije in zaljubljenosti. Seveda so eni pari bolj zaljubljeni kot drugi. Stopnjo zaljubljenosti, ki je vrednost med 0 in 100, izračunajo na sila čuden, Butalam primeren način. Nebutalcem in predvsem Tepanjcem ga je silno težko razložiti, zato ga bomo razložili v dveh delih.

Najprej drugi del, ki izvira iz stare butalske igre dva na dva. Pri tej igri imamo seznam števk (med 0 in 9) in nasprotnika, ki izmenično ponavljata korak dva na dva:

1. Tekmovalec vzame prvo in zadnjo števko ter ju sešteje. Seštevek predstavlja prvo ali prvi (odvisno od velikosti seštevka) števk seznam za naslednji korak.
2. Tekmovalec nadaljuje z drugo in predzadnjo števko, kar ustvari naslednjo ali naslednji števk seznam, in tako naprej.
3. Če je bila dolžina seznama na začetku soda, se seštevanje lepo izteče, v nasprotnem primeru pa preostalo števko tekmovalec samo pripiše na konec seznama.
4. Če je v novem seznamu samo ena ali dve števki, je tekmovalec zmagal, sicer pa nasprotnik izvede korak dva na dva.

Poglejmo si konkreten primer: začetni seznam naj bo 2911121, imamo belega in sivega igralca, ki je tudi zmagal:

1. 2911121 → 3
2. 2911121 → 311
3. 2911121 → 3112
4. 2911121 → 31121

1. 31121 → 4
2. 31121 → 43
3. 31121 → 431

1. 431 → 5
2. 431 → 53

Še razlaga prvega dela, ki bo ustvaril seznam števk, na katerem bomo igrali igro dva na dva:

1. Najprej par na listek z velikimi tiskanimi črkami napiše svoji imeni.
2. Nato združi vse črke iz obeh imen in jih abecedno uredi.
3. Prešteje, kolikokrat se pojavlja posamezna črka, in dobi iskani seznam števk.
4. Na seznamu izvede igro dva na dva in izidu prišteje 1 – namreč zaljubljenost, kar predstavlja dobljeno število, pač ne more biti med 0 in 99, ampak med 100 in 1.

Oglejmo si postopek še na primeru, kjer Mihec želi izračunati, kakšna je zaljubljenost med njim in Manco:

1. Na listek napiše imeni: MIHEC in MANCA.
2. Prešteje pojavitve posameznih črk v imenih ter jih zapiše v abecednem vrstnem redu (A = 2, C = 2, E = 1, H = 1, I = 1, M = 2 in N = 1):

A	B	C	Č	D	E	F	G	H	I	J	K	L	M	N	O	P	R	S	Š	T	U	V	Z	Ž
2		2			1			1	1				2	1										

3. Tako dobi število 2211121, ki se v igri dva na dva najprej preoblikuje v 3421 in na koncu v 46, kar pomeni, da je njuna stopnja zaljubljenosti 47.



- 6.1. Mihec se čez en teden na novo zaljubi v sošolko Petro. Pomagajte mu izračunati stopnjo zaljubljenosti.

Stopnja zaljubljenosti med Mihcem in Petro je: _____ (3 točke)

- 6.2. Računanje stopnje zaljubljenosti je zelo popularno in sošolci neprestano prosijo Mihca, naj jim jo izračuna. Da bo pisanje kalkulatorja lažje, najprej pomagajte Mihcu napisati funkcijo `dva_na_dva`, ki kot argument dobi seznam števk, vrne pa izid igre kot število med 0 in 99.

```
def dva_na_dva (seznam):
```

(3 točke)



- 6.3. Dajmo še malo pomoči Mihcu, da bo napisal funkcijo `kalkulator_zaljubjenosti`. Funkcija kot argument dobi imeni `ime1` in `ime2` obeh zaljubljenцев, vrne pa stopnjo njune zaljubljenosti v obliki naravnega števila med 1 in 100. Pri pisanju lahko predpostavite, da je funkcija `dva_na_dva` že napisana in da pravilno deluje. Morda boste želeli porabiti v rešitvi tudi funkcijo `ord(crka)`, kjer je `crka` črka slovenske abecede. Vrednosti, ki jih funkcija vrne pri slovenski abecedi, so med 0 in 512.

```
abeceda = 'ABCČDEFGHIJKLMNOPRSŠTUVZŽ'
def kalkulator_zaljubjenosti (ime1, ime2):
    imeni = ime1.upper() + ime2.upper()
```

(4 točke)



Prazna stran