

1. Princip lokalnosti (angl. *principle of locality*):

- ✗ Pravi, da je zelo verjetno, da bo program znova uporabil isti podatek ali ukaz.
- ✗ Pravi, da je zelo verjetno, da bo program uporabil ukaz ali podatek, ki je blizu v pomnilniku trenutno uporabljanemu podatku /izvajanemu ukazu.
- ✗ Je osnova za učinkovito delovanje predpomnilnika.
- ✓ Drži vse zgoraj naštet.

2. Moore-ov zakon pravi, da se število tranzistorjev na tiskanem vezju (angl. *printed circuit board*) podvoji cca. vsake dve leti.

- ✗ Zakon nima nobenega pomena za hitrost računalnikov, le za kapaciteto različnih tipov vezij, predvsem pomnilniških, ki se povečujejo skladno s številom tranzistorjev.
- ✗ Večja gostota tranzistorjev nima pomembnega vpliva na hitrost računalnika, ker elektrika potuje s svetlobno hitrostjo in je zakasnitev med dvema tranzistorjema zanemarljivo majhna.
- ✓ Zaradi večje gostote in posledično manjše razdalje med tranzistorji, se poveča tudi hitrost računalnikov.
- ✗ Nič od zgoraj naštetega ne drži.

3. Zbirnik pri prevajanju izvornega programa (angl. *source program*) v strojni jezik:

- ✗ zgradi tabelo ukazov (angl. *op code table*), ki so bili uporabljeni v izvornem programu.
- ✗ zgradi tabelo simbolov (angl. *symbol table*), ki vsebuje uporabljene simbole in naslove kjer so le-ti bili uporabljeni v izvornem programu.
- ✓ zgradi tabelo simbolov, ki vsebuje uporabljene simbole in naslove oznak (angl. *label*) kjer so le-ti bili uporabljeni v izvornem programu.
- ✗ V prvem prehodu zgradi tabelo ukazov, v drugem prehodu pa tabelo simbolov.
- ✗ oba, (a) in (d) sta pravilna.

4. Imamo dva različna algoritma za isto opravilo. Algoritem A ima povprečno časovno zahtevnost $O(\frac{1}{2} \cdot n^2)$, algoritem B pa $O(2 \cdot n \cdot \log_2 n)$. Kateri algoritem je bolj smiselno uporabljati?

- ✗ Algoritem A .
- ✗ Algoritem B .
- ✓ Pri majhnih n algoritem A , pri velikih n pa algoritem B .
- ✗ Pri majhnih n algoritem B , pri velikih n pa algoritem A .

5. Dvojiški številski sistem se v računalništvu za predstavitev podatkov uporablja:

- ✗ Ker je z elektriko (tok teče ali ne teče) ali magnetnim poljem (namagneteno v smeri urinega kazalca ali obratno) možno predstaviti točno dve vrednosti.
- ✓ Ker takšna zasnova omogoča visoko stopnjo zanesljivosti.
- ✗ Zaradi združljivosti s prejšnjimi generacijami računalnikov, ki so zaradi enostavnosti uporabljali dvojiški sistem.
- ✗ Ker v logiki potrebujemo le dve vrednosti, pravilno (TRUE) in napačno (FALSE).

6. Za računalništvo lahko rečemo, da je veda algoritmičnega reševanja problemov. Algoritmi so pomembni, ker:

- ✗ z njimi rešujemo večino matematičnih problemov.
- ✗ jih lahko vedno zapišemo v jeziku logike in tako uporabljamo le dve vrednosti (TRUE in FALSE) – to pa je na kožo pisano računalnikom, ki uporabljajo dvojiški številski sistem.
- ✓ če lahko sestavimo algoritem za reševanje nekega problema, lahko reševanje tega problema avtomatiziramo.
- ✗ če lahko sestavimo algoritem za reševanje nekega problema, lahko ta problem formalno zapišemo v izbranem programskem jeziku.

7. Vsak Booleov izraz lahko predstavimo z diagramom vezja (torej logičnim vezjem) in vsak izhod v diagramu vezja lahko zapišemo z Booleovim izrazom.

- ✓ Trditev je pravilna.
- ✗ Trditev ni pravilna.
- ✗ Trditev drži izključno pod pogojem, da diagram vezja uporablja samo vrata AND, OR in NOT.
- ✗ Trditev drži izključno pod pogojem, da diagram vezja uporablja samo vrata NOR in NAND.

8. Za Von Neumannov model računalnika je značilno:

- ✗ da je računalnik sestavljen iz štirih glavnih komponent: spomina, vhoda/izhoda, aritmetično-logične enote in kontrolne enote.
- ✗ da so programi shranjeni v pomnilniku računalnika.
- ✗ da se operacije izvajajo zaporedno.
- ✓ vse od zgoraj naštetega.
- ✗ drži samo (a) in (c).

9. Prekinitev (angl. *interrupt*):

- ✗ omogoča, da lahko uporabnik prekine izvajanje trenutnega programa kadarkoli želi.
- ✗ je pomembna, ker lahko z njeno pomočjo uporabnik prekine izvajanje trenutnega programa z drugim, ki je zanj ta hip bolj pomemben.
- ✓ je pomembna, ker na ta način vhodno-izhodni (V/I) krmilnik razbremeni procesor od večnega čakanja na dokončanje V/I operacije.
- ✗ je pomembna, ker omogoča ohlajanje procesorja.

10. Katera od spodnjih trditev glede predstavitve števil z dvojiškim komplementom je pravilna?

- ✓ V tem zapisu je samo ena ničla.
- ✗ V tem zapisu obstajata dve ničli, kar je dobro, ker lahko ločimo med +0 in -0.
- ✗ V tem zapisu obstajata dve ničli, kar pa je slabo, ker lahko povzroči zmedo.
- ✗ Ta zapis omogoča dvojno natančnost operacij v plavajoči vejici.

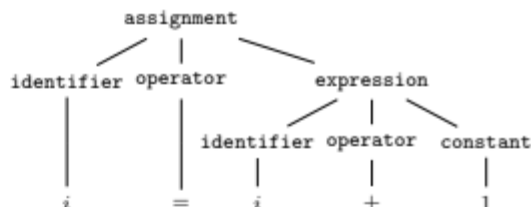
1. S pomočjo paritetnega bita (ki ni dodan spodnjim podatkom) lahko zaznamo napako pri prenosu:
 - × $00 \rightarrow 11$
 - × $000 \rightarrow 101$
 - ✓ $0000 \rightarrow 0010$
 - × Samo pri prenosu nizov sode dolžine.
 - × Odgovori (a), (b) in (c) so vsi pravilni.
2. Kratka notacija za Turingove stroje je v obliki (trenutno stanje, vhodni znak, izhodni znak, novo stanje, smer premika). Predpostavimo, da imamo stroj s samo naslednjim pravilom: $(1, 0, 1, 1, R)$. Naloga tega stroja je, da na traku vse ničle spremeni v enice, pri čemer je vhod na traku binarno število, npr. vhod 00101 spremeni v 11111 (lahko si npr. predstavljamo, da to predstavlja dolžino niza v eniškem sistemu). Katero pravilo manjka, da bi Turingov stroj deloval pravilno za to nalogo?
 - ✓ $(1, 1, 1, 1, R)$.
 - × $(1, b, b, 2, R)$.
 - × $(2, 1, 1, 2, R)$.
 - × Nobeno dodatno pravilo ni potrebno; Turingov stroj že deluje pravilno.
 - × Nič od naštetega.
3. Algoritem smo definirali kot "natančno urejeno zaporedje nedvoumnih in učinkovito izračunljivih operacij, ki vrne rezultat in se ustavi v končnem času".
 - × Algoritme je mogoče izvajati le na elektronskih računalnikih.
 - ✓ Pojem "algoritem" je veliko starejši od elektronskih računalnikov.
 - × Funkcijski programski jeziki niso primerni za zapis algoritmov.
 - × Nobena od zgornjih trditev ne drži.
4. Nek algoritem za računovodsko obdelavo podatkov ima povprečno časovno zahtevnost $O(1)$. Čez leto je bilo vsak mesec približno enako število zahtevkov za obdelavo, v decembru pa se je število zahtevkov podvojilo. Dani algoritem je prejšnje mesece potreboval 5 ur za obdelavo podatkov (za vsak mesec). Za mesec december bo potreboval:
 - ✓ 5 ur.
 - × 10 ur.
 - × 20 ur.
 - × Nimamo zadosti informacij, da bi to lahko napovedali.

5. Kako prevajalnikov razčlenjevalnik (angl. *parser*) ve kako sestaviti razčlenitveno drevo (angl. *parse tree*) za podani program?
- ✗ Obstajajo le trije tipi operacij, s katerimi lahko opišemo vsak algoritem in le-ti ustrezajo vozliščem v razčlenitvenem drevesu.
 - ✗ Vsako vozlišče v drevesu je povezano s specifičnim ukazom v zbirniku; preprosto preiskovanje je potrebno za izgradnjo ustreznega drevesa, ki je potem prevedeno v strojno kodo z zbirnikom.
 - ✓ Načrtovalci programskega jezika pripravijo gramatična pravila, ki jih prevajalnik uporabi za razčlenjevanje programov v tem programskem jeziku; preiskovanje pa poskrbi za preostalo.
 - ✗ Načrtovalci računalnika, na katerem bo program tekel, pripravijo gramatična pravila, ki jih prevajalnik uporabi za razčlenjevanje programov na tem računalniku; preiskovanje pa poskrbi za preostalo.
6. V računalniku shranjeno število dobi pomen šele, ko je uporabljeno (interpretirano) na določen način. Pomen je odvisen izključno od konteksta. Recimo, da je računalnik binarno število 10000011 interpretiral kot pravilno prenešen znak A. Kontekst je v tem primeru:
- ✗ Eniški komplement.
 - ✗ Dvojiški komplement.
 - ✗ ASCII zapis (črka A ima kodo 65) z dodanim zadnjim bitom (1) za preverjanje *sode* paritete.
 - ✓ ASCII zapis (črka A ima kodo 65) z dodanim zadnjim bitom (1) za preverjanje *lihe* paritete.
7. Po katerem velikem vojskovodji se imenuje tehnika šifriranja? (ključ: +5)
- ✗ MKTLOTF
 - ✗ ITKJGAJI
 - ✓ VASTL
 - ✗ MTŽDFJR
8. Osnovni gradnik strojne opreme (angl. "hardware device") mora izpolnjevati naslednji kriterij oz. kriterije:
- ✗ Imeti mora zmožnost razpoznavanja/merjenja stanja, brez da bi to stanje pri tem bilo uničeno.
 - ✗ Imeti mora zmožnost spremembe stanja z uporabo zadostne količine energije.
 - ✗ Med stanji mora imeti zadosti veliko energetske prepreko.
 - ✓ Vse od naštetega.
9. Princip lokalnosti (angl. *principle of locality*) je v računalništvu pomemben, ker:
- ✗ Olajša razvoj programov, ker je zaradi lokalno omejenega dostopa do metod in spremenljivk nekega razreda, tak program
- (a) lažje vzdrževati in spreminjati.
- ✓ Lajša probleme povezane s takoimenovanim Von Neumannovim ozkim grlom.
 - ✗ S svojimi protokoli omogoča lažji dostop do lokalnih zmogljivosti v omrežju.
 - ✗ Danes zaradi razvoja večjedrnih procesorjev princip lokalnosti nima več velikega pomena.
10. Gramatika, ki vsebuje samo naslednje BNF pravilo: $\langle s \rangle ::= a \mid b \mid a \langle s \rangle a \mid a a \langle s \rangle$
- ✗ Generira samo besede, ki vsebujejo liho število znakov a.
 - ✗ Je nedvoumna.
 - ✗ Generira samo besede lihe dolžine, a se nobena beseda ne more končati z znakom b.
 - ✗ Lahko generira le končno mnogo besed.
 - ✓ Nič od naštetega.

1. Kaj v računalniški komunikaciji označujemo z angleškim izrazom "collision"?
 - (a) Hkraten dostop do pomnilnika s strani dveh (ali več) aplikacij.
 - (b) Hkraten dostop do istih lokacij v pomnilniku s strani dveh (ali več) aplikacij.
 - ☒ (c) Hkratno pošiljanje sporočil po isti liniji s strani dveh ali več uporabnikov.
 - (d) Hkratna zahteva po dostopu do spletne strani s strani dveh ali več uporabnikov.
2. Church-Turingova teza trdi:
 - (a) Churchev in Turingov stroj sta ekvivalentna.
 - (b) Obstajajo problemi, ki so neizračunljivi in jih nikoli ne bomo mogli rešiti s pomočjo računalnikov.
 - (c) Vsak izračunljiv problem lahko rešimo ali s pomočjo Turingovega stroja ali s pomočjo Churchevega stroja.
 - ☒ (d) Vsak izračunljiv problem lahko rešimo s pomočjo Turingovega stroja.
3. Psevdokoda:
 - (a) Zahteva, da so vse vrstice obvezno oštevilčene.
 - (b) Uporablja le zanko while, ostale zanke (npr. for) zaradi enostavnejšega branja kode niso zaželeno.
 - (c) Uporablja enolično predpisan nabor stavčnih struktur.
 - ☒ (d) Nič od naštetega.
4. Mrežni protokoli so organizirani po nivojih (angl. *layers*). To je koristno, ker:
 - (a) Omogoča dober pregled s strani skrbnikov interneta oz. lokalne mreže.
 - (b) Na ta način se protokoli med seboj ne ovirajo (ne pridejo v konflikt).
 - (c) Če spremenimo protokol znotraj enega nivoja, to ne vpliva na ostale nivoje in lahko le-ti večinoma ostanejo takšni kot so bili (lažje vzdrževanje).
 - (d) Od tega ni danes nobene posebne koristi; tako je ostalo iz zgodovinskih razlogov.
5. Prekinitev (angl. *interrupt*):
 - (a) je pomembna, ker lahko z njeno pomočjo operacijski sistem prekine izvajanje programa, ki je naletel na napako med izvajanjem.
 - (b) je pomembna, ker lahko z njeno pomočjo uporabnik prekine izvajanje programa, ki je v mrtvi zanki, ki je operacijski sistem ni uspel preprečiti ali odpraviti.
 - ☒ (c) je pomemben koncept, ki je koristen zaradi dejstva, da različne strojne komponente računalnika delujejo z zelo različnimi hitrostmi.
 - (d) je pomembna, ker operacijskemu sistemu omogoča ohlajanje procesorja.

6. Po kateri fazi prevajanja zgleda izhod tako kot na sliki desno?

- (a) Leksikografski analizi.
- ☒ (b) Razčlenjevanju (angl. *parsing*).
- (c) Semantični analizi in generiranju kode.
- (d) Optimizaciji kode.



7. Nek algoritem za spletno obdelavo naročil ima povprečno prostorsko zahtevnost $O(1)$. V mesecu decembru je bilo veliko naročil (10.000), v januarju pa je število naročil močno upadlo (na samo 1.000 naročil). Dani algoritem bo za obdelavo vseh naročil v posameznem mesecu potreboval:

- (a) 10.000 bajtov za december in 1.000 bajtov za januar.
- (b) 10.000 pomnilniških celic za december in 1.000 pomnilniških celic za januar.
- ☒ (c) Približno enako prostora za oba posamezna meseca.
- (d) Nimamo zadosti informacij, da bi to lahko napovedali.

8. Katero izmed spodnjih števil nikakor ne pride v poštev kot javni ključ za RSA?

- (a) 11.
- (b) 23.
- (c) 1024.
- ☒ (d) Vsa naštetá števila pridejo v poštev, ker je ključ tako ali tako javen in ne privaten in zato ni posebnih omejitev.

9. Binarno število 10010101 računalniku pomeni:

- (a) 149.
- (b) -150.
- (c) Črko 'J' (1001010 = 74) v ASCII zapisu z dodanim zadnjim bitom (1) za preverjanje sode paritete.
- (d) Črko 'J' (1001010 = 74) v ASCII zapisu z dodanim zadnjim bitom (1) za preverjanje lihe paritete.
- ☒ (e) Shranjeno število dobi pomen šele, ko je uporabljeno (interpretirano) na določen način. Pomen je odvisen izključno od konteksta. Tako brez poznavanja konteksta ne moremo povedati kaj število pomeni.

10. Ena izmed nalog zbirnika je, da prevede simbolične naslove v binarne vrednosti. To naredi tako, da:

- (a) Poišče naslov za dani simbol v tabeli kod operacij (angl. *op code lookup table*), ki je podana za dani računalnik.
- (b) Poišče naslov za dani simbol v tabeli simbolov (angl. *symbol lookup table*), ki je podana za dani računalnik.
- ☒ (c) Poišče naslov za dani simbol v tabeli simbolov, ki jo ustvari zbirnik.
- (d) Nič od naštetega.

1. Imamo dva različna algoritma za isto opravilo. Algoritem A ima povprečno časovno zahtevnost $O(1)$, algoritem B pa $O(\log_2 n)$. Kateri algoritem je v splošnem bolj smiselno uporabljati?
 - ✓ Algoritem A .
 - ✗ Algoritem B .
 - ✗ Pri majhnih n algoritem A , pri velikih n pa algoritem B .
 - ✗ Oba algoritma sta hitra, zato je vseeno katerega izberemo.
 - ✗ Nimamo zadosti informacij, da bi se lahko odločili.
2. Naloga domenskega strežnika (angl. *Domain Name Server*; DNS) je:
 - ✗ Skrbeti za nemoteno komunikacijo med računalniki v eni domeni.
 - ✗ Skrbeti za nemoteno komunikacijo med računalniki v več različnih domenah.
 - ✗ Skrbeti za dinamično dodeljevanje IP naslovov v eni domeni.
 - ✓ Nič od naštetega.
3. Vse algoritme lahko opišemo s samo tremi tipi operacij.
 - ✗ Trditev ni resnična.
 - ✗ Trditev je pogojno resnična: pogoj je, da algoritmi ustrezajo Church-Turingovi tezi.
 - ✗ Trditev je pogojno resnična: pogoj je, da algoritme zapišemo s pomočjo psevdokode.
 - ✓ Trditev je resnična.
4. S pomočjo paritetnega bita (ki ni dodan spodnjim podatkom) lahko zaznamo napako pri prenosu:
 - ✗ $0000 \rightarrow 0001$
 - ✗ $0001 \rightarrow 0000$
 - ✗ $1101 \rightarrow 0011$
 - ✓ Vse zgoraj naštet: (a), (b) in (c).
 - ✗ Nič od naštetega.
5. Možno je narediti računalnik, katerega nabor ukazov bi vseboval le en ukaz in bi lahko izračunal vse kar lahko izračunajo RISC (reduced instruction set computer) ali CISC (complex instruction set computer) računalniki.
 - ✗ Trditev ni pravilna, pri čemer velja tudi, da CISC računalniki lahko izračunajo več kot RISC računalniki.
 - ✗ Trditev ni pravilna, pri čemer velja tudi, da RISC računalniki lahko izračunajo več kot CISC računalniki.
 - ✓ Trditev je pravilna.
 - ✗ Trditev je pogojno pravilna: teoretično je to mogoče, a takšen "vsemogočen" ukaz v praksi še ni bil najden.

6. Kratka notacija za Turingove stroje je v obliki (trenutno stanje, vhodni znak, izhodni znak, novo stanje, smer premika). Predpostavimo, da imamo stroj s samo naslednjim pravilom: (1, 0, b, 1, R). Naloga tega stroja je, da s traku zbriše neželene vodilne ničle pri čemer je vhod na traku binarno število, npr. vhod 000101 spremeni v samo 101 in vhod 000 v samo 0. Katero pravilo manjka, da bi Turingov stroj deloval pravilno za to nalogo?

- × (1, 1, 1, 1, R).
- × (1, b, 0, 2, R).
- × (2, 1, 1, 2, R).
- ✓ Nobeno dodatno pravilo ni potrebno; Turingov stroj že deluje pravilno.
- × Nič od naštetega.

7. Dvojiški številski sistem se v računalništvu za predstavitev podatkov uporablja:

- × Ker je z elektriko (tok teče ali ne teče) ali magnetnim poljem (namagneteno v smeri urinega kazalca ali obratno) možno predstaviti točno dve vrednosti.
- × Ker v logiki potrebujemo le dve vrednosti, pravilno (TRUE) in napačno (FALSE).
- ✓ Ker takšna zasnova omogoča širok "varovalni pas" med fizičnima vrednostima, ki predstavljata logično ničlo in enico.
- × Zaradi združljivosti s prejšnjimi generacijami računalnikov, ki so zaradi enostavnosti uporabljali dvojiški sistem.
- × Nič od naštetega.

8. Kako prevajalnikov razčlenjevalnik (angl. *parser*) ve kako sestaviti razčlenitveno drevo (angl. *parse tree*) za podani program?

- × Vsako vozlišče v drevesu je povezano s specifičnim strojnim ukazom, implementiranim s strojno opremo; preprosto preiskovanje je potrebno za izgradnjo ustreznega drevesa.
- × Vsako vozlišče v drevesu je povezano s specifičnim ukazom v zbirniku; preprosto preiskovanje je potrebno za izgradnjo ustreznega drevesa, ki je potem prevedeno v strojno kodo z zbirnikom.
- ✓ Načrtovalci programskega jezika pripravijo gramatična pravila, ki jih prevajalnik uporabi za razčlenjevanje programov v tem programskem jeziku; preiskovanje pa poskrbi za preostalo.
- × Načrtovalci računalnika, na katerem bo program tekel, pripravijo gramatična pravila, ki jih prevajalnik uporabi za razčlenjevanje programov na tem računalniku; preiskovanje pa poskrbi za preostalo.

9. Algoritem vsote produktov (angl. *Sum-of-Products*) za izgradnjo vezij:

- × Potrebuje samo vrata AND, ki predstavljajo "produkte" in vrata OR, ki predstavljajo "vsoto".
- × Zgradi neoptimalno vezje glede na število uporabljenih vrat, vendar se optimizacija danes več ne splača, ker so vrata zadosti majhna in poceni.
- × Zgradi neoptimalno vezje glede na število uporabljenih vrat; optimizacija pa je potrebna izključno iz finančnih razlogov.
- ✓ Zgradi neoptimalno vezje glede na število uporabljenih vrat; optimizacija pa je še danes koristna, ker med drugim prihrani prostor na vezju, energijo in zmanjša segrevanje vezja.

10. "BSIJUFLUVSJ ZFDJOF SBDVOBMOJLPZ KF RPŠPEJM JNF"? (ključ: -1)

- ✓ John von Neumann.
- × Alan Turing.
- × Gordon Moore.
- × George Boole.

1. Nek algoritem za urejanje števil ima povprečno časovno zahtevnost $O(n * n!)$. Tak algoritem bo za $n = 10$ v povprečju potreboval:
 - ✗ Manj kot 100 operacij.
 - ✗ Približno 1.000 operacij.
 - ✗ Približno 10.000 operacij.
 - ✓ Več kot 1.000.000 operacij.
2. S pomočjo paritetnega bita lahko pri prenosu podatkov:
 - ✗ Zaznamo le eno (na nivoju posameznega bita) samo napako.
 - ✓ Zaznamo liho število napak.
 - ✗ Zaznamo sodo število napak.
 - ✗ Napako popravimo, ker vemo kateri bit je napačen.
3. Kakšna je poglobljena prednost asimetričnih kriptirnih algoritmov (npr. RSA) pred simetričnimi?
 - ✓ Asimetrični algoritmi nimajo težav z varnim prenosom ključa med pošiljateljem in prejemnikom.
 - ✗ Asimetrični algoritmi so znatno hitrejši od simetričnih algoritmov.
 - ✗ Kriptiranje sloni na teoriji števil in prafaktorjih, slednjih pa je dokazano neskončno mnogo in zato ne more zmanjkati različnih ključev.
 - ✗ Asimetrični algoritmi nimajo nobene pomembne prednosti pred simetričnimi algoritmi.
4. Vsak zbirni jezik mora obvezno vsebovati operacijo CLEAR X (ukaz postavi pomnilniško celico X na vrednost 0).
 - ✗ Trditev drži, ker te operacije ni možno nadomestiti z nobeno drugo.
 - ✗ Trditev drži, ker sicer jezik ne bi bil Turing-popoln.
 - ✓ Trditev ne drži.
 - ✗ Trditve ne moremo preveriti, ker je odvisna od vsakega posameznega računalnika.
5. Algoritem smo definirali kot "natančno urejeno zaporedje nedvoumnih in učinkovito izračunljivih operacij, ki vrne rezultat in se ustavi v končnem času".
 - ✗ Algoritme je mogoče izvajati le na elektronskih računalnikih.
 - ✓ Pojem »algoritem« je veliko starejši od elektronskih računalnikov.
 - ✗ Funkcijski programski jeziki niso primerni za zapis algoritmov.
 - ✗ Nobena od zgornjih trditev ne drži.

6. Algoritem vsote produktov (angl. *Sum-of-Products*) za izgradnjo vezij:

- ✗ Vedno zgradi optimalno vezje glede na število uporabljenih vrat (čim manjše).
- ✗ Skoraj vedno (a ne vedno) zgradi optimalno vezje glede na število uporabljenih vrat.
- ✓ Potrebuje vrata AND, OR in NOT.
- ✗ Potrebuje samo vrata AND, ki predstavljajo "produkte" in vrata OR, ki predstavljajo "vsoto".

7. Katera trditev je pravilna?

- ✗ Programski jezik, katerega sintaksa je definirana s pomočjo BNF gramatike, je nedvoumen.
- ✓ Terminali gramatik predstavljajo končne simbole, ki se uporabljajo v jeziku (npr. if ali while).
- ✗ Programski jeziki, definirani s pomočjo dvoumne gramatike, so načeloma veliko uporabnejši zaradi možnosti optimizacije.
- ✗ Pravilna je trditev (b), a le za imperativno paradigmo programskih jezikov.

8. Prednost zbirnega jezika pred strojnim jezikom je:

- ✓ Uporaba simboličnih pomnilniških naslovov namesto binarnih.
- ✗ Prenosljivost programov med različnimi računalniki.
- ✗ Uporabnost tudi v ne-Von Neumannovih arhitekturah.
- ✗ Nič od naštetega.

9. Church-Turingova teza trdi naslednje:

- ✓ Poljuben izračunljiv problem lahko rešimo s pomočjo Turingovega stroja.
- ✗ Churchev in Turingov stroj sta povsem ekvivalentna.
- ✗ Poljuben izračunljiv problem lahko rešimo ali s pomočjo Turingovega stroja ali s pomočjo Churchevega stroja, stroja se torej dopolnjujeta.
- ✗ Obstajajo problemi, ki so neizračunljivi in jih nikoli ne bomo mogli rešiti s pomočjo računalnikov.

10. Naloga domenskega strežnika (angl. *Domain Name Server*; DNS) je:

- ✗ Skrbeti za varno povezovanje znotraj domene.
- ✗ Skrbeti za varno povezovanje med računalniki med domenami.
- ✓ Skrbeti za preslikavo simboličnih internetnih imen v IP naslove.
- ✗ Skrbeti za dinamično dodeljevanje IP naslovov.

1. Namen IP (Internet Protocol) številke je:

- ✗ Sporočanje namembnosti strežnika s to številko ostalim računalnikom v omrežju.
- ✓ Identifikacija računalnikov v internetnem omrežju.
- ✗ Identifikacija protokola po katerem bo potekala internetna komunikacija s tem strežnikom.
- ✗ Posredovanje javnega ključa strežnika s to številko.

2. TLS/SSL (Transport Layer Security/Secure Sockets Layer) protokol za varen prenos podatkov tipično uporablja:

- ✗ CRC kodo.
- ✗ DES šifriranje.
- ✗ RSA šifriranje.
- ✓ DES in RSA šifriranje.

3. Sortirni algoritem A ima povprečno časovno zahtevnost $O(0,5 \cdot n^2)$, sortirni algoritem B pa $O(2 \cdot n \cdot \log_2 n)$. Kateri algoritem se nam bolj splača uporabiti za sortiranje 16 števil?

- ✗ Algoritem A.
- ✗ Algoritem B.
- ✓ Oba sta enako dobra.
- ✗ Imamo premalo informacij, da bi lahko to napovedali.

4. Z vidika izračunljivosti uredi naslednje programske paradigme (simbol $>$ pomeni »lahko izračuna več« oz. »reši več problemov«):

- ✗ Imperativna $>$ logična $>$ funkcijska.
- ✗ Funkcijska $>$ logična $>$ imperativna.
- ✗ Imperativna $>$ funkcijska $>$ logična.
- ✓ Vse tri so enakovredne.

5. »Programa, ki bi za poljuben drug program in za poljuben vhod povedal ali bo le-ta vrnil točno določen (npr. pravi) rezultat, ni mogoče napisati.«

- ✓ Trditev je resnična.
- ✗ Trditev ni resnična.
- ✗ Trditev je resnična, a le za logično programsko paradigmo.
- ✗ Nimamo zadosti informacij, da bi lahko ocenili resničnost trditve.

6. Gramatika, ki vsebuje samo naslednje BNF pravilo: $\langle \text{bnum} \rangle ::= 0 \mid 1 \mid \langle \text{bnum} \rangle \langle \text{bnum} \rangle \mid \Lambda$

- ✗ Ne generira vseh binarnih števil.
- ✓ Je dvoumna.
- ✗ Ni dvoumna, a ne generira vseh binarnih števil.
- ✗ Ni dvoumna in generira vsa binarna števila.

7. S pomočjo samo vrat NOR (not OR):

- ✗ Ni možno zgraditi vrat OR.
- ✓ Je možno zgraditi računalnik, ki je Turing-popoln.
- ✗ Ni možno zgraditi računalnika, ki je Turing-popoln.
- ✗ Drži (a) in (c).

8. Kontrolne operacije kot so npr. *if*, *else* in *while*:

- ✓ Niso sestavni del vsakega programskega jezika.
- ✗ So sestavni del le psevdokode in skriptnih jezikov (kot npr. Python).
- ✗ Drži (a) in (b).
- ✗ Nič od naštetega.

9. S pomočjo paritetnega bita (ki ni dodan spodnjim podatkom) lahko zaznamo napako pri prenosu:

- ✗ $1111 \rightarrow 0000$
- ✓ $1011 \rightarrow 0000$
- ✗ $1001 \rightarrow 0000$
- ✗ Pri (a) in (c).
- ✗ Pri (a) in (b) in (c).

10. Algoritem vsote produktov (angl. *Sum-of-Products*) za izgradnjo vezij:

- ✗ Vedno zgradi optimalno vezje glede na število vrat (čim manjše).
- ✗ Skoraj vedno (a ne vedno) zgradi optimalno vezje glede na število vrat.
- ✗ Ne uporablja vrat NOT, ampak samo vrata AND in OR.
- ✓ Nič od naštetega.

1. Spletne aplikacije za prenos občutljivih podatkov tipično uporabljajo:
 - ✗ Asimetrične kriptirne algoritme, ker so veliko hitrejši od simetričnih.
 - ✗ Simetrične kriptirne algoritme, ker porabijo bistveno manj pomnilnika kot asimetrični.
 - ✓ Kombinacijo simetričnih in asimetričnih, ker se njihove prednosti dobro dopolnjujejo.
 - ✗ Kombinacijo simetričnih in asimetričnih, ker le tako zmorejo dešifrirati različne načine kriptiranja ključev uporabnikov.
2. Katera izjava je pravilna?
 - ✓ Možno je zgraditi računalnik, ki pozna samo en ukaz, ki je Turing-popoln.
 - ✗ Funkcijski jeziki niso Turing-popolni, ker so prilagojeni predvsem za matematične izračune.
 - ✗ V programskem jeziku C je možno rešiti več problemov kot v zbirnem jeziku.
 - ✗ Vseh višjenivojskih programskih jezikov ni možno prevesti v strojno kodo.
3. Kontrolne operacije kot so npr. *if*, *for* in *while*:
 - ✗ So sestavni del vsakega programskega jezika.
 - ✗ So nujno potrebne za hitrejše izvajanje sekvenčnih ukazov.
 - ✗ So sestavni del le psevdokode in skriptnih jezikov (kot npr. Python).
 - ✓ Se uporabljajo za spreminjanje vrstnega reda izvajanja ukazov.
4. S pomočjo samo vrat NAND (not AND):
 - ✗ Ni možno zgraditi vrat AND.
 - ✓ Je možno zgraditi računalnik, ki je Turing-popoln.
 - ✗ Ni možno zgraditi računalnika, ki je Turing-popoln.
 - ✗ Drži (a) in (c).
5. Zvok je tipično vzorčen s 44,1 kHz:
 - ✗ Zaradi kompatibilnosti s starejšimi audio napravami.
 - ✓ Ker takšna frekvenca vzorčenja omogoča zajem celotnega območja človeškega sluha.
 - ✗ Ker bi višja frekvenca bila slišati nenavadno (moteče) za ljudi.
 - ✗ Nič od prej naštetega ne drži.

6. Pri gramatikah, terminali:

- ✗ Se od neterminalov razlikujejo po tem, da se uporabljajo tudi na levi strani BNF pravila.
- ✓ Predstavljajo končne simbole, ki se uporabljajo v jeziku (npr. for ali enačaj).
- ✗ Se od neterminalov razlikujejo po tem, da se uporabljajo tudi na desni strani BNF pravila.
- ✗ Nič od prej naštetega ne drži.

7. Programske knjižnice (angl. *libraries*):

- ✗ Ni možno implementirati za zbirni jezik.
- ✗ Je možno implementirati le za višjenivojske jezike.
- ✗ Drži (a) in (b).
- ✓ Nič od prej naštetega ne drži.

8. ARQ algoritem, ki nadzoruje komunikacijo med omrežnima vozliščema A (pošiljatelj) in B (prejemnik):

- ✗ Deluje, ker A hrani kopijo posameznega dela sporočila.
- ✗ Deluje, ker B potrdi prejem vsakega dela sporočila.
- ✓ Drži (a) in (b).
- ✗ Nič od prej naštetega ne drži.

9. Razdalja med tranzistorji v vezju:

- ✗ Nima pomembnega vpliva na hitrost računalnika, ker elektrika potuje s svetlobno hitrostjo in je zakasnitev med dvema tranzistorjema zanemarljivo majhna.
- ✗ Nima pomembnega vpliva na hitrost računalnika, ker je le-ta odvisna predvsem od nabora ukazov, ki jih pozna.
- ✗ Nima pomembnega vpliva na hitrost računalnika, je pa zelo pomembna za velikost njegovega pomnilnika.
- ✓ Nič od prej naštetega ne drži.

10. Nek algoritem ima asimptotično časovno zahtevnost $O(2^n)$. Izvajamo ga na računalniku z 8-jedrnim procesorjem, ki je sposoben optimalno izkoristiti vsa njegova jedra, torej je pohitritev dejansko 8-kratna. Kakšna je asimptotična časovna zahtevnost algoritma, ko se izvaja na tem računalniku?

- ✗ $O(2^{n/8})$
- ✗ $O(0,25^n)$
- ✓ $O(2^n)$
- ✗ Nič od prej naštetega ne drži.

1. Church-Turingova teza trdi:

- ✓ Vsak izračunljiv problem lahko rešimo s pomočjo Turingovega stroja.
- ✗ Churchev in Turingov stroj sta ekvivalentna.
- ✗ Vsak izračunljiv problem lahko rešimo ali s pomočjo Turingovega stroja ali s pomočjo Churchevega stroja.
- ✗ Obstajajo problemi, ki so neizračunljivi in jih nikoli ne bomo mogli rešiti s pomočjo računalnikov.

2. Program, ki bi za poljubna dva programa povedal ali sta ekvivalentna (bosta za poljuben vhod vrnila isti rezultat) ali ne:

- ✗ Je možno napisati za Turingov stroj, ker ima neskončen trak (pomnilnik).
- ✗ Je možno napisati v višjenivojskih jezikih, ne pa tudi v zbirnem jeziku.
- ✗ Ni možno napisati le v deklarativnih jezikih.
- ✓ Ni možno napisati.

3. Naloga domenskega strežnika (angl. *Domain Name Server*; DNS) je:

- ✗ Skrbeti za nemoteno komunikacijo med računalniki v eni domeni.
- ✗ Skrbeti za nemoteno komunikacijo med računalniki v več različnih domenah.
- ✓ Skrbeti za preslikavo simboličnih internetnih imen v IP naslove.
- ✗ Skrbeti za dinamično dodeljevanje IP naslovov v eni domeni.

4. S pomočjo paritetnega bita (ki ni dodan spodnjim podatkom) lahko zaznamo napako pri prenosu:

- ✗ $0010 \rightarrow 0000$
- ✗ $1101 \rightarrow 1110$
- ✗ $1101 \rightarrow 1010$
- ✓ Pri (a) in (c).
- ✗ Pri (a) in (b) in (c).

5. Ob pozabljenem geslu želimo, da nam sistem sporoči novo (začasno) geslo. Zakaj nam večina sistemov (aplikacij) ne sporoči kar našega originalnega gesla?

- ✗ Zato, da uporabnika prisili v menjavo gesla.
- ✓ Sistem originalnega gesla ne pozna in nima zadosti podatkov, da bi ga lahko rekonstruiral.
- ✗ Ker je rekonstrukcija originalnega gesla zamuden postopek in bi vzel preveč časa, še posebej pri internetnih aplikacijah, kjer je pričakovan hiter odziv.
- ✗ Večina sistemov pošlje originalno geslo.

6. Kakšna je glavna prednost asimetričnega kriptirnega algoritma (npr. RSA) pred simetričnimi?

- ✗ Kriptiranje sloni na teoriji števil in prafaktorjih, slednjih pa je dokazano neskončno mnogo in zato ne more zmanjkati različnih ključev.
- ✓ Asimetrični algoritem nima težav z varnim prenosom ključa med pošiljateljem in prejemnikom.
- ✗ Asimetrični algoritmi nimajo nobene realne prednosti pred simetričnimi algoritmi.
- ✗ Asimetrični algoritmi so precej hitrejši od simetričnih algoritmov.

7. Glavna prednost visokonivojskih jezikov pred zbirnim jezikom je:

- ✗ Podatki so v visokonivojskih jezikih lahko zapisani bistveno krajše, kar pomeni, da lahko prihranimo precej pomnilnika.
- ✗ Izvajanje programov napisanih v visokonivojskih jezikih je bistveno hitrejšo.
- ✓ Programiranje v visokonivojskih jezikih je bistveno hitrejšo.
- ✗ V visokonivojskih jezikih lahko rešimo več problemov kot v zbirnem jeziku.

8. Programski jezik je Turing-popoln (angl. *Turing complete*), če:

- ✗ Spada v vsaj dve od naslednjih paradigem programskih jezikov: imperativno, funkcijsko, deklarativno in objektno-orientirano.
- ✗ Zanj obstaja ustrezna BNF gramatika.
- ✓ Z njim lahko simuliramo izvajanje poljubnega Turingovega stroja z enim trakom.
- ✗ Lahko vsako popolno rekurzijo nadomesti z zanko.

9. Kadar lahko prevajalnik nek programski stavek s pomočjo gramatike izpelje na več različnih načinov (več različnih dreves izpeljav, angl. *parse trees*):

- ✓ je to slabo, ker ima lahko tak stavek več različnih pomenov.
- ✗ je to dobro, ker lahko kasneje prevajalnik izbere izpeljavo, ki vodi do bolj učinkovitega (prostorsko ali časovno) prevedenega programa (optimizacija).
- ✗ Je to dobro, ker se pohitri delovanje prevajalnika (hitreje najde vsaj eno izpeljavo, če jih je več možnih).
- ✗ nima to nobenega posebnega vpliva.

10. Leksikografska analiza izvornega programa:

- ✗ zgradi tabelo uporabljenih spremenljivk in jim dodeli pomnilniške naslove.
- ✓ poskrbi, da se izvorni program razbije na posamezne žetone, ki se jih obenem tudi klasificira v različne tipe (npr. število, spremenljivka, rezervirana beseda *while*, itd.).
- ✗ poskrbi za stavčno analizo stavka podobno kot pri naravnih jezikih.
- ✗ Prevede posamezne dele programa s pomočjo slovarja ukazov (leksikona).

1. Algoritem je:

- ☒ (a) natančno urejeno zaporedje nedvoumnih in učinkovito izračunljivih operacij, ki vrne rezultat in se ustavi v končnem času;
- (b) postopek, ki vedno vrne pravi rezultat;
- (c) drži (a) in (b);
- (d) nič od naštetega.

2. Pseudokoda:

- (a) zahteva, da so vse vrstice obvezno oštevilčene;
- (b) uporablja le zanko while, ostale zanke (npr. for) zaradi enostavnejšega branja kode niso zaželeni;
- (c) uporablja enolično predpisan nabor stavčnih struktur;
- ☒ (d) nič od naštetega.

3. Prednost strojnega jezika pred zbirnim jezikom je:

- (a) uporaba simboličnih pomnilniških naslovov namesto binarnih;
- (b) prenosljivost programov med različnimi računalniki;
- (c) uporabnost tudi v ne-Von Neumannovih arhitekturah;
- ☒ (d) nič od naštetega.

4. Glavna naloga V/I krmilnika (angl. I/O controller) je:

- ☒ (a) izboljšanje izkoristka procesorja preko premoščanja med zelo različnimi hitrostmi delovanja procesorja in V/I naprave s pomočjo prekinitiv;
- (b) prevajanje med ukazi v zbirnem jeziku in ukazi, ki jih razume V/I naprava;
- (c) nadzor nad smerjo komunikacije (vhod ali izhod);
- (d) nič od naštetega.

5. Princip lokalnosti (angl. *principle of locality*) pravi:

- (a) da je zelo verjetno, da bo program znova uporabil isti podatek ali ukaz;
- (b) da je zelo verjetno, da bo program uporabil ukaz ali podatek, ki je blizu v pomnilniku trenutno uporabljanemu podatku/izvajanemu ukazu;
- ☒ (c) da drži (a) in (b);
- (d) da operacijski sistem med izvajanjem programa iz varnostnih razlogov hrani lokalno kopijo programa in njegovih podatkov.

6. Dvojiški številski sistem se v računalništvu za predstavitev podatkov uporablja:

- (a) zaradi združljivosti s prejšnjimi generacijami računalnikov, ki so zaradi enostavnosti uporabljali dvojiški sistem;
- ☒ (b) zaradi zanesljivosti, ki jo omogoča takšen sistem;
- (c) ker je z elektriko (tok teče ali ne teče) ali magnetnim poljem (namagneteno v smeri urinega kazalca ali obratno) možno predstaviti le dve vrednosti.
- (d) nič od naštetega.

7. Pri analizi prostorske zahtevnosti algoritmov:

- (a) je algoritem s prostorsko zahtevnostjo $O(1000 \cdot \log n)$ asimptotično bolj potraten od algoritma s prostorsko zahtevnostjo $O(\frac{1}{2} \cdot n^2)$;
- ☒ (b) je algoritem s prostorsko zahtevnostjo $O(1000 \cdot \log n)$ asimptotično manj potraten od algoritma s prostorsko zahtevnostjo $O(\frac{1}{2} \cdot n^2)$;
- (c) je algoritem s prostorsko zahtevnostjo $O(1000 \cdot \log n)$ asimptotično približno enako potraten (razlika je zanemarljiva) od algoritma s prostorsko zahtevnostjo $O(\frac{1}{2} \cdot n^2)$;
- (d) brez dodatnih informacij ne moremo oceniti kateri izmed algoritmov je asimptotično bolj oz. manj potraten.

8. Zbirnik pri prevajanju izvirnega programa (angl. *source program*) v strojni jezik navadno:

- (a) naredi en prehod čez izvorno kodo;
- (b) naredi dva prehoda čez izvorno kodo;
- (c) naredi tri prehode čez izvorno kodo;
- (d) potrebuje samo tabelo simbolov (angl. *symbol table*).

9. Mrtva zanka (angl. *deadlock*):

- ☒ (a) nastopi, ko program *A* zaseda vir *V1*, program *B* zaseda vir *V2*, oba pa potrebujeta za nadaljevanje izvajanja oba vira, torej *V1* in *V2*;
- (b) se zgodi, ko operacijski sistem zaradi varnosti začasno zaklene uporabniški račun po več nepravilnih poskusih vnosa gesla;
- (c) je zanka v programu, ki se ne bo nikoli izvedla;
- (d) nastopi, ko program *A*, ki se normalno izvaja, zaseda vira *V1* in *V2*, program *B* pa za svoje nadaljevanje potrebuje vsaj enega od njih.

10. Moore-ov (Gordon E. Moore) zakon:

- (a) pravi, da se hitrost tranzistorjev na tiskanem vezju (angl. *printed circuit board*) podvoji cca. vsake dve leti;
- (b) ni matematični ali fizikalni zakon in predvideva se, da čez nekaj (deset) let ne bo več (povsem) veljal;
- ☒ (c) pravi, da se število tranzistorjev na tiskanem vezju (angl. *printed circuit board*) podvoji cca. vsake dve leti;
- (d) drži (a) in (b);
- (e) drži (b) in (c).

1. Za kaj skrbi povezavna plast v Internetni hierarhiji protokolov?
 - (a) Za uokvirjanje bitov v sporočila.
 - (b) Za nadzor nad dostopom do medija.
 - (c) Za detekcijo in odpravljanje napak pri prenosu z algoritmom ARQ.
 - ☒ (d) Za vse zgoraj naštet.
2. Kaj moramo v standardnem ciklu razvoja programske opreme opraviti pred samim kodiranjem oz. implementacijo?
 - (a) Nakup prostorov, nakup strojne in programske opreme.
 - ☒ (b) Študijo izvedljivosti, specifikacijo problema, načrt programa, izbor ali razvoj ter analizo algoritmov.
 - (c) Testiranje, verifikacijo, razhroščevanje.
 - (d) Nič od zgoraj naštetega.
3. Tri krat tri je (Cezarjeva šifra):
 - (a) tdeve
 - (b) davet
 - (c) aoscs
 - ☒ (d) oscsa
4. Kaj je sistemska programska oprema?
 - ☒ (a) Sistemska programska oprema zagotavlja uporabniku predstavitev pomembnih informacij na razumljiv način ter mu dovoli dostop do virov na razumljiv in varen način.
 - (b) Sistemska programska oprema je skupek aplikacij, ki omogočajo uporabniku učinkovito reševanje konkretnih problemov.
 - (c) Sistemska programska oprema skrbi za povezovanje med različnimi informacijskimi sistemi.
 - (d) Veljata (a) in (b).
5. Katera trditev je pravilna?
 - (a) Sintaksna analiza preverja sintaksno pravilnost semantičnega drevesa.
 - (b) Sintaksna analiza preverja semantično pravilnost semantičnega drevesa.
 - ☒ (c) Semantična analiza preverja semantično pravilnost sintaksnega drevesa.
 - (d) Semantična analiza preverja sintaksno pravilnost sintaksnega drevesa.

6. Ena izmed nalog zbirnika je, da prevede simbolične naslove v binarne vrednosti. To naredi tako, da:
- (a) Poišče naslov za dani simbol v tabeli kod operacij (*angl. op code lookup table*), ki je podana za dani računalnik.
 - (b) Poišče naslov za dani simbol v tabeli simbolov (*angl. symbol lookup table*), ki je podana za dani računalnik.
 - ☒ (c) Poišče naslov za dani simbol v tabeli simbolov, ki jo ustvari zbirnik.
 - (d) Nič od naštetega.
7. Kaj je posledica Paradoksa negotovosti oz. Collingridgeve dileme?
- (a) Zaradi negotovih zadetkov v predpomnilniku pride do Collingridgejevega ozkega grla oz. upočasnitve delovanja računalnika z večstopenjsko pomnilniško strukturo.
 - (b) Mrtve zanke je nemogoče detektirati, ker je negotovo kje se bodo pojavile.
 - (c) Pretirana začetna regulacija novih tehnologij preprečuje razvoj, prepozna regulacija pa je neučinkovita.
 - (d) Zaradi kvantne nedoločljivosti so kvantni računalniki, čeprav teoretično izjemno zmogljivi, v praksi še vedno neizvedljivi.
8. Katera trditev je pravilna?
- (a) Algoritem je urejeno zaporedje operacij, ki, če se ustavi, proizvede rezultat.
 - ☒ (b) Algoritem je popolnoma urejeno zaporedje nedvoumnih in učinkovito izračunljivih operacij, ki, ko se le-te izvedejo, proizvede rezultat in se ustavi v končnem času.
 - (c) Algoritem je popolnoma urejeno enosmerno zaporedje enostavnih operacij, ki, ko se le-te izvedejo, proizvede rezultat in se ustavi v končnem času.
 - (d) Algoritem je popolnoma urejeno zaporedje nedvoumnih in učinkovito izračunljivih operacij, ki proizvede pravilen rezultat in se ustavi v končnem času.
9. Pri digitalizaciji videa:
- (a) Vzorcimo signal po času; hranimo samo vrednosti na diskretnih intervalih 25-krat na sekundo.
 - (b) Vzročimo signal prostorsko; vrednosti barv shranjujemo na diskretnih intervalih v dveh dimenzijah.
 - (c) Kvantiziramo vrednosti amplitude signala; hranimo samo diskretne vrednosti barv.
 - ☒ (d) Drži vse zgoraj naštet.
10. Kaj pomeni, da je programska koda prenosljiva?
- ☒ (a) Razvita programska koda se lahko (po prevajanju ali interpretiranju) izvaja na različnih računalnikih.
 - (b) Programska koda je dovolj lahka, da jo prenašamo na USB ključu.
 - (c) Programska aplikacija je izdelana tako, da jo lahko uporabljajo različni uporabniki brez dodatne avtentikacije.
 - (d) Programska koda je šifrirana s simetričnim ključem; tako se lahko prenaša med različnimi računalniki na varen način.

1. Katera trditev drži:

- ☒ (a) Računalništvo je znanost učinkovitega algoritmičnega reševanja problemov.
- (b) S pravim algoritmom lahko rešimo katerikoli problem.
- (c) Algoritmi se lahko izvajajo samo na Von Neumanovi arhitekturi.
- (d) Veljata (a) in (b).

2. Pri digitalizaciji videa:

- (a) Vzorčimo signal po času; hranimo samo vrednosti na diskretnih intervalih 25-krat na sekundo.
- (b) Vzročimo signal prostorsko; vrednosti barv shranjujemo na diskretnih intervalih v dveh dimenzijah.
- (c) Kvantiziramo vrednosti amplitude signala; hranimo samo diskretne vrednosti barv.
- ☒ (d) Drži vse zgoraj naštet.

3. Tri krat štiri je (Cezarjeva šifra):

- (a) tsjanavd
- (b) whpžpuz
- (c) whpžbpuz
- ☒ (d) whpžpbuz

4. Prednost strojnega jezika pred zbirnim jezikom je:

- (a) uporaba simboličnih pomnilniških naslovov namesto binarnih;
- (b) prenosljivost programov med različnimi računalniki;
- (c) abstrakcija dela z mikroukazi;
- ☒ (d) nič od naštetega.

5. Za Von Neumannov model računalnika je značilno:

- (a) da je računalnik sestavljen iz štirih glavnih komponent: pomnilnika, centralno-procesne enote, vhodno/izhodnih naprav in aritmetično-logične enote.
- (b) da je računalnik sestavljen iz štirih glavnih komponent: pomnilnika, vhodno/izhodnih naprav, aritmetično-logične enote in kontrolne enote.
- (c) da so programi shranjeni v pomnilniku računalnika.
- ☒ (d) drži (b) in (c).

6. Kaj testiramo na Turingovem testu?

- (a) Ali se bo Turingov stroj ustavil.
- (b) Ali lahko prevedemo pravila za Turingov stroj v algoritem brez zank.
- (c) Ali je problem izračunljiv.
- ☒ (d) Inteligenco stroja.

7. Kateri spodaj omenjeni protokoli delujejo na aplikacijski plasti v Internetni hierarhiji protokolov?

- (a) TCP, IP.
- (b) SMTP, POP3, IMAP.
- (c) HTTP, FTP.
- ☒ (d) Veljata (b) in (c).

8. Kateri izmed spodnjih delov kode je napisan v programskem jeziku za logično programiranje?

- (a) `SELECT * FROM Students WHERE Year=1`
- (b) `(define negate(X) (-X))`
- ☒ (c) `toTheLeft(X, Y) :- left(X, Z), toTheLeft(Z, Y).`
- (d) `for i=1:10, print(i), end;`

9. Katera trditev je pravilna?

- (a) Sintaksna analiza analizira znake in jih združuje v lekseme.
- ☒ (b) Sintaksna analiza gradi sintaksno drevo.
- (c) Sintaksna analiza analizira semantično pravilnost sintaksnega drevesa.
- (d) Sintaksna analiza analizira sintaksno pravilnost semantičnega drevesa.

10. Koliko tranzistorjev je potrebnih za implementacijo vrat za negacijo?

- ☒ (a) 1
- (b) 2
- (c) 3
- (d) 4

1. Psevdokoda:

- (a) uporablja le zanko while, ostale zanke (npr. for) zaradi enostavnejšega branja kode niso zaželeni;
- (b) uporablja enolično predpisan nabor stavčnih struktur;
- (c) zahteva, da so vse vrstice obvezno oštevilčene;
- ☒ (d) nič od naštetega.

2. Pri digitalizaciji zvoka:

- (a) vzorčimo signal po času; hranimo samo vrednosti na diskretnih intervalih;
- (b) kvantiziramo vrednosti amplitude signala; hranimo samo diskretne vrednosti amplitude;
- ☒ (c) držita (a) in (b);
- (d) ne drži nič od zgoraj naštetega.

3. Mrtva zanka (angl. deadlock):

- (a) se zgodi, ko operacijski sistem zaradi varnosti začasno zaklene uporabniški račun po več nepravilnih poskusih vnosa;
- ☒ (b) nastopi, ko program A zaseda vir V1, program B zaseda vir V2, oba pa potrebujeta za nadaljevanje izvajanja oba vira, torej V1 in V2;
- (c) je zanka v programu, ki se ne bo nikoli izvedla;
- (d) nastopi, ko program A, ki se normalno izvaja, zaseda vira V1 in V2, program B pa za svoje nadaljevanje potrebuje vsaj enega od njih.

4. Izbirnik oz. multiplekser

- (a) je vezje, ki pošlje signal na en izhod izbran z vrednostjo na vhodnih linijah;
- ☒ (b) je vezje, ki z N izbirnimi linijami izberega enega od 2^N vhodov;
- (c) je prevajalnik iz zbirnega v strojni jezik;
- (d) je V/I krmilnik, ki izbere pravo V/I napravo.

5. Kako poteka pisanje v pomnilniško lokacijo?

- (a) Naloži naslov v MDR, naloži vrednost v MAR, dekodiraj naslov v MDR, shrani vsebino iz MAR na pomnilniško lokacijo.
- (b) Naloži naslov v MAR, dekodiraj naslov v MAR, shrani vsebino iz registra R1 na pomnilniško lokacijo.
- (c) Naloži naslov v PC, prenesi vsebino registra R1 na naslov na katerega kaže PC.
- ☒ (d) Naloži naslov v MAR, naloži vrednost v MDR, dekodiraj naslov v MAR, shrani vsebino iz MDR na pomnilniško lokacijo.

6. Internetna hierarhija protokolov je sestavljena iz naslednjih plasti:

- (a) fizična, mrežna, aplikaijska;
- (b) fizična, mrežna, prenosna, aplikacijska;
- ☒ (c) fizična, povezavna, mrežna, prenosna, aplikacijska;
- (d) fizična, povezovalna, mrežna, programska, aplikacijska.

7. Kadar lahko prevajalnik nek programski stavek s pomočjo gramatike izpelje na več različnih načinov (več različnih dreves izpeljav, angl. parse trees):

- (a) Je to dobro, ker se pohitri delovanje prevajalnika (hitreje najde vsaj eno izpeljavo, če jih je več možnih).
- (b) je to dobro, ker lahko kasneje prevajalnik izbere izpeljavo, ki vodi do bolj učinkovitega (prostorsko ali časovno) prevedenega programa (optimizacija).
- ☒ (c) je to slabo, ker ima lahko tak stavek več različnih pomenov.
- (d) je to slabo, ker so takšne izpeljave bolj zamudne.

8. ARQ algoritem, ki nadzoruje komunikacijo med omrežnima vozliščema A (pošiljatelj) in B (prejemnik):

- (a) Deluje, ker B potrdi prejem vsakega sporočila.
- (b) Deluje, ker A hrani kopijo posameznega sporočila.
- ☒ (c) Držita (a) in (b).
- (d) Nič od prej naštetega ne drži.

9. BSIJUFLUVSJ ZFDJOF SBDVOBMOJLP Z KF RPŠPEJM JNF? (ključ: -1)

- (a) George Boole.
- ☒ (b) John von Neumann.
- (c) Alan Turing.
- (d) Gordon Moore.

10. Programa, ki bi za poljuben drug program in za poljuben vhod povedal ali bo le-ta vrnil točno določen (npr. pravilen) rezultat, ni mogoče napisati.

- (a) Trditev ni resnična.
- ☒ (b) Trditev je resnična.
- (c) Trditev je resnična, a le za logično programsko paradigmo.
- (d) Nimamo zadosti informacij, da bi lahko ocenili resničnost trditve.

1. Katera definicija najbolj opredeli računalništvo?

- (a) Računalništvo je veda, ki preučuje računalnike.
- (b) Računalništvo je veda, ki preučuje kako pisati računalniške programe.
- (c) Računalništvo je veda, ki preučuje uporabo in aplikacije računalnikov in programske opreme.
- ☒ (d) Računalništvo je veda o načrtovanju in razvoju algoritmov za reševanje množice pomembnih problemov, vključno z načrtovanjem algoritmov, njihovi strojni in jezikovni realizaciji ter aplikaciji.

2. Kaj je algoritem?

- (a) Algoritem je urejeno zaporedje operacij, ki, če se ustavi, proizvede rezultat.
- (b) Algoritem je popolnoma urejeno enosmerno zaporedje enostavnih operacij, ki, ko se le-te izvedejo, proizvede rezultat in se ustavi v končnem času.
- ☒ (c) Algoritem je popolnoma urejeno zaporedje nedvoumnih in učinkovito izračunljivih operacij, ki, ko se le-te izvedejo, proizvede rezultat in se ustavi v končnem času.
- (d) Algoritem je popolnoma urejeno zaporedje nedvoumnih in učinkovito izračunljivih operacij, ki proizvede pravilen rezultat in se ustavi v končnem času.

3. Prvi programabilni stroj je bil:

- (a) Pascalina Francoza Blaisea Pascala.
- ☒ (b) Avtomatske statve Francoza Josepha Jacquarda.
- (c) Diferenčni stroj Angleža Charlesa Babbagea.
- (d) Colossus Angleža Alana Turinga.

4. Katere trditve o analognih podatkih držijo:

- (a) Analogni podatki lahko zavzamejo samo diskretne vrednosti.
- (b) Analogni podatki imajo zvezno zalogo vrednosti.
- (c) Analogni podatki lahko zavzamejo neskončno število vrednosti.
- ☒ (d) Držita (b) in (c).

5. Algoritem vsote produktov (angl. Sum-of-Products) za izgradnjo vezij:

- (a) Potrebuje samo vrata AND, ki predstavljajo "produkte" in vrata OR, ki predstavljajo "vsoto".
- (b) Zgradi optimalno vezje, ki implementira izbrano booleov izraz.
- ☒ (c) Zgradi neoptimalno vezje glede na število uporabljenih vrat, ki implementira izbrani booleov izraz.
- (d) Nobena izmed zgornjih trditev ne drži.

6. Imamo tri različna algoritme za isto opravilo. Algoritem A ima povprečno časovno zahtevnost $O(\frac{1}{2} \cdot n^2)$, algoritem B ima časovno zahtevnost $O(2^n)$, algoritem C pa $O(10 \cdot n \cdot \log_2 n)$. Kateri algoritem je najbolj smiselno uporabljati?
- ☒ (a) Algoritem A .
 - (b) Algoritem B .
 - (c) Algoritem C .
 - (d) Vseeno je kateri algoritem uporabimo.
7. Princip lokalnosti (angl. *principle of locality*):
- (a) Pravi, da je zelo verjetno, da bo program znova uporabil isti podatek ali ukaz.
 - (b) Pravi, da je zelo verjetno, da bo program uporabil ukaz ali podatek, ki je blizu v pomnilniku trenutno uporabljanemu podatku /izvajanemu ukazu.
 - (c) Je osnova za učinkovito delovanje predpomnilnika.
 - ☒ (d) Drži vse zgoraj naštetega.
8. Za Von Neumannov model računalnika je značilno:
- (a) da je računalnik sestavljen iz štirih glavnih komponent: pomnilnika, vhoda/izhoda, aritmetično-logične enote in kontrolne enote.
 - (b) da so programi shranjeni v pomnilniku računalnika.
 - (c) da se operacije izvajajo zaporedno.
 - ☒ (d) vse od zgoraj naštetega.
 - (e) drži samo (a) in (c).
9. Pri načrtovanju binarnega računalnika moramo zagotoviti:
- (a) dve stabilni energijski stanji.
 - (b) dve energijski stanji in tretje stanje za robustnost.
 - (c) stikala za preklapljanje med stanji.
 - ☒ (d) (a) in (c)
 - (e) (b) in (c)
10. Katera od spodnjih trditev glede predstavitve števil z dvojiškim komplementom je pravilna?
- ☒ (a) V tem zapisu je samo ena ničla.
 - (b) V tem zapisu obstajata dve ničli, kar je dobro, ker lahko ločimo med $+0$ in -0 .
 - (c) V tem zapisu obstajata dve ničli, kar pa je slabo, ker lahko povzroči zmedo.
 - (d) Ta zapis omogoča dvojno natančnost operacij v plavajoči vejici.

1. Naloga domenskega strežnika (angl. Domain Name Server; DNS) je:
 - (a) Skrbeti za nemoteno komunikacijo med računalniki v eni domeni.
 - (b) Skrbeti za nemoteno komunikacijo med računalniki v več različnih domenah.
 - ☒ (c) Skrbeti za preslikavo simboličnih internetnih imen v IP naslove.
 - (d) Skrbeti za dinamično dodeljevanje IP naslovov v eni domeni.
2. Operacijski sistem ve kateri aplikaciji je namenjeno prejeta sporočila po:
 - (a) IP številki vsebovani v sporočilu.
 - (b) DNS zapisu vsebovanem v sporočilu.
 - ☒ (c) navedeni številki vrat (angl. port).
 - (d) nizu z imenom ciljne aplikacije vsebovanem v sporočilu.
3. Church-Turingova teza trdi:
 - (a) Obstajajo problemi, ki so neizračunljivi in jih nikoli ne bomo mogli rešiti s pomočjo računalnikov.
 - (b) Churchev in Turingov stroj sta ekvivalentna.
 - (c) Vsak problem lahko rešimo ali s pomočjo Turingovega stroja ali s pomočjo Churchevega stroja.
 - ☒ (d) Če obstaja algoritem, ki reši neko nalogo, potem obstaja tudi Turingov stroj, ki reši isto nalogo.
4. Ob pozabljenem geslu želimo, da nam sistem sporoči novo (začasno) geslo. Zakaj nam večina sistemov (aplikacij) ne sporoči kar našega originalnega gesla?
 - (a) Zato, da uporabnika prisili v menjavo gesla.
 - (b) Ker bi lahko originalno geslo kdo prestregel.
 - ☒ (c) Ker sistem originalnega gesla ne pozna, saj ga nima shranjenega v taki obliki.
 - (d) Večina sistemov pošlje originalno geslo.
5. Glavna prednost visokonivojskih jezikov pred zbirnim jezikom je:
 - (a) Izvajanje programov napisanih v visokonivojskih jezikih je bistveno hitrejše.
 - (b) Podatki so v visokonivojskih jezikih lahko zapisani bistveno krajše, kar pomeni, da lahko prihranimo precej pomnilnika.
 - (c) V visokonivojskih jezikih lahko rešimo več problemov kot v zbirnem jeziku.
 - ☒ (d) Programiranje v visokonivojskih jezikih je bistveno hitrejše in enostavnejše.

6. Leksikalna analiza izvornega programa:

- (a) prevede posamezne dele programa s pomočjo slovarja ukazov (leksikona).
- (b) zgradi tabelo uporabljenih spremenljivk in jim dodeli pomnilniške naslove.
- ☒ (c) izvorni program razbije na posamezne lekseme in jih obenem tudi klasificira v različne tipe (npr. število, spremenljivka, rezervirana beseda while, itd.).
- (d) poskrbi za stavčno analizo stavka podobno kot pri naravnih jezikih.

7. HMBZOP NFŠUP ŠMPZFOJKF KF (Cezarjeva šifra, ključ +1):

- (a) Rim
- (b) London
- (c) Dunaj
- ☒ (d) Ljubljana

8. Glavne naloge operacijskih sistemov so:

- (a) uporabiški vmesnik in komunikacija z uporabnikom
- (b) varnost in zaščita sistema
- (c) učinkovito razporejanje in varna uporaba virov
- ☒ (d) vse zgoraj naštet.

9. Tabela simbolov se zgradi

- ☒ (a) pri prevajanju programa v zbirnem jeziku v strojni jezik.
- (b) pri optimizaciji kode.
- (c) pri interpretaciji programa v interpreterskem jeziku.
- (d) pri leksikalni analizi.

10. Turingov stroj ima samo naslednji dve pravili: $(1,1,1,1,R)$ in $(1,0,1,1,R)$. Kakšen bo izhod po zaustavitvi tega Turingovega stroja?

- (a) Biti bodo invertirani glede na vhod.
- (b) Vseboval bo samo ničle.
- ☒ (c) Vseboval bo samo enice.
- (d) Nič od zgoraj naštetega.