

Poklicna Matura 2009 – Algoritmi in programski jeziki

Vprašanja za ustni del izpita

1. Kaj je to **algoritem** in kaj so njegove lastnosti?
2. Kaj je to **algoritem** in kako ga lahko zapišemo?
3. Nariši in razloži **gradnike** diagramov poteka.
4. Naštej in opiši generacije programskih jezikov. Za vsako generacijo navedi primer programskega jezika in primer ukaza v tem programskem jeziku.
5. Naštej in opiši generacije programskih jezikov. Kaj je glavna razlike med tako imenovanimi nižjimi in višjimi programskimi jeziki?
6. Kakšna je razlika med postopkovnimi in nepostopkovnimi programskimi jeziki? Navedi primer za vsako skupino.
7. Opiši razliko med **objektnim** in **strukturnim** programiranjem.
8. Kaj je **program**? S čim ga napišemo in kako ga pripravimo za izvajanje?
9. Opiši razlike med prevajanjem (kompiliranjem) in tolmačenjem (interpretiranjem) programov.
10. Skiciraj in opiši ustvarjanje izvršne kode v C++.
11. Kaj je to **semantika** in kaj **sintaksa** programskega jezika. Razloži pri katerih napakah (semantičnih ali sintaktičnih) lahko prevajalnik pomaga programerju? Katero orodje pa vsaj delno pomaga pri odpravljanju druge vrste napak?
12. Napiši strukturo programa v C++.
13. Razloži uporabo **predprocesorskih** direktiv v C++.
14. Kaj so **zaglavne** datoteke? Kako jih vključimo v program in zakaj jih sploh uporabljamo. Ali v C in C++ v programe vključujemo iste zaglavne datoteke? Razloži.
15. Naštej in razloži ter na primeru prikaži uporabo funkcij za **vpis** in **izpis** podatkov v C++.
16. Naštej in opiši **enostavne** podatkovne tipe v C++. Koliko prostora zasedajo v pomnilniku? Napiši deklaracijo spremenljivke in konstante za podatek tipa po lastni izbiri.
17. Naštej in opiši **enostavne** in **sestavljene** podatkovne tipe v C++. Opiši razliko med enostavnimi in sestavljenimi tipi.
18. Kakšna je razlika med **deklaracijo** in **definicijo** spremenljivke? Napiši primer za eno in drugo ter še primer za oboje hkrati. Pri zadnjem primeru razloži še pojem inicializacija spremenljivke.
19. Naštej in razloži **aritmetične** operatorje v C++.
20. Naštej in razloži **logične** operatorje v C++.
21. Naštej in razloži operatorje **prirejanja** v C++.
22. Naštej in razloži **primerjalne** operatorje v C++.
23. Opiši sintakso stavka **IF – ELSE** v C++. Navedi primer in nariši diagram poteka zanj.
24. Opiši sintakso stavka **SWITCH – CASE** v C++. Navedi primer in nariši diagram poteka zanj.
25. Opiši sintakso stavka **FOR** v C++. Navedi primer in nariši diagram poteka zanj.
26. Opiši sintakso stavka **WHILE** v C++. Navedi primer in nariši diagram poteka zanj.
27. Opiši sintakso stavka **DO – WHILE** v C++. Navedi primer in nariši diagram poteka zanj.
28. Katere **zanke** obstajajo v C++? Napiši njihovo sintakso in jih primerjaj med seboj na podlagi primera: izpisa števil od 1 do 5.
29. Opiši uporabo stavkov **BREAK** in **CONTINUE** v C++. Navedi primera in nariši diagrama poteka zanju.
30. Primerjaj stavek **RETURN** in funkcijo **EXIT()** v C++.
31. Kaj so to **funkcije**? Na lastnem primeru prikaži njihovo uporabo.
32. Kaj so to **prototipi** funkcij? Zakaj in kako jih uporabljamo? Prikaži na lastnem primeru. Navedi še, kako bi enako dosegli brez uporabe prototipov.
33. Opiši vrste **prenosa** parametrov pri uporabi funkcij.
34. Skiciraj in razloži, kaj se dogaja v pomnilniku pri klicu funkcije s parametri po **vrednosti**.
35. Skiciraj in razloži, kaj se dogaja v pomnilniku pri klicu funkcije s parametri po **referenci** z uporabo **kazalcev**.
36. Kaj je **življenjska doba** spremenljivke? Kakšna je v C++ pri avtomatičnih, pri statičnih in pri dinamičnih spremenljivkah? Kaj veš povedati o avtomatični inicializaciji zgornjih treh vrst spremenljivk?
37. Kaj so **lokalne** in kaj **globalne** spremenljivke? Kako deklariramo ene in kako druge? Kaj je z avtomatično inicializacijo, življenjsko dobo in dosegljivostjo enih in drugih?
38. Kaj so **polja**? Za kaj so uporabna? Skiciraj polje v pomnilniku. Napiši deklaracijo polja v C++. Napiši, kako dostopamo do posameznih elementov polja.

39. Kako lahko polja v C++ **inicializiramo**? Prikaži na primeru.
40. Napiši **deklaracijo** dveh enako dolgih polj enakih tipov in kodo v C++, ki bo prvo polje kopirala v drugo.
41. Kaj so to **večdimenzionalna** polja? V C++ deklariraj poljubno dvodimenzionalno polje in inicializiraj njegove vrednosti. Napiši tabelico, s katero si lahko to polje predstavljamo in nariši skico tega pomnilnika z vrednostmi tega polja.
42. V C++ deklariraj poljubno **dvodimezionalno** polje in napiši kodo, ki bo omogočila, da uporabnik vpiše vrednosti v polje.
43. Razloži, kako prenašamo v funkcije **enodimenzionalna** polja in kako **večdimenzionalna**, tako da so funkcije čim bolj neodvisne od velikosti polja.
44. Razloži, kako **prenašamo** polja v funkcije. Kaj se zgodi z elementi polja, če jih v neki funkciji spremenimo: ali so spremembe vidne tudi izven funkcije ali ne? Za kakšno vrsto prenosa parametra v funkcijo torej gre?
45. V C++ ustvari **dinamično** polje z elementi po lastni izbiri. Dolžino polja naj določi uporabnik. Kaj je prednost dinamičnih polj pred statičnimi?
46. V C++ ustvari **dinamično** polje z elementi po lastni izbiri. Dolžino polja naj določi uporabnik. Pokaži, kako sprostimo pomnilnik, ki ga polje zaseda.
47. Kako so predstavljeni **nizi znakov** v C++? Napiši primer deklaracije in inicializacije niza ter skiciraj stanje v pomnilniku.
48. S primeri ilustriraj funkcije za **kopiranje, primerjanje in dolžino** niza, predstavljenega s poljem znakov v C++. Ne pozabi vključiti potrebne zaglavne datoteke.
49. Opiši in na primeru polja, ki bo vsebovalo imena dni v tednu ilustriraj uporabo **polj nizov**, predstavljenih s poljem znakov v C++ (deklaracija, inicializacija, vpis niza v polje). Razlago opremi s skico.
50. Razloži, kako prenašamo **nize** v funkcije. Za kakšno vrsto prenosa gre? Česa pri prenosu niza v funkcijo ne rabimo, medtem ko to pri prenosu drugih vrst polj rabimo? Razloži, zakaj pride do tega.
51. Kaj so **kazalci**? Kako jih v C++ deklariramo in vanje vpisujemo vrednosti?. Kako dostopamo do njih in do podatkov na katere kažejo? Napiši ustrezno kodo v C++ in skiciraj stanje v pomnilniku.
52. V C++ ustvari **spremenljivko** poljubnega enostavnega tipa, **kazalec** na to spremenljivko in **kazalec na kazalec**. Nato v spremenljivko, ki si jo ustvaril najprej, vpiši z uporabo kazalca in še z uporabo kazalca na kazalec. Skiciraj dogajanje v pomnilniku.
53. Razloži in prikaži uporabo operatorjev ***** in **&** v zvezi s kazalci in referencami v C++.
54. Razloži sintaktično razliko med **indeksnim** in **kazalčnim** delom s polji pri prenosu polj v funkcije in pri dostopanju do elementov polj ter njihovih naslovov.
55. Kaj so **strukture** v C++? Zakaj so koristne? Po čem se razlikujejo od polj? Napiši deklaracijo strukture in primerka (spremenljivke) te strukture v C++.
56. Napiši deklaracijo **strukture** in **primerka** (spremenljivke) te strukture v C++. Nato si izberi enega od elementov strukture in pokaži, kako lahko dostopaš do njega.
57. Razloži in na primeru prikaži razliko med operatorjema **.** in **->** za doseganje elementov struktur v C++.
58. V C++ deklariraj poljubno strukturo in ustvari njen primerek tako, da vrednosti elementov strukture inicializiraš že ob deklaraciji spremenljivke. Nato pokaži, kako lahko vrednost nekega elementa spremeniš.
59. V C++ ustvari strukturo, ki bo vsebovala drugo strukturo. To stori na dva načina: tako da sta obe strukturi deklarirani samostojno in je druga vključena v prvo in tako, da je druga struktura anonimna in gnezdena v prvi strukturi. Nato prikaži razliko v dostopanju do elementov strukture v strukturi.
60. V C++ ustvari strukturo, primerek strukture in kazalec na primerek strukture. Nato pokaži, na katere vse načine lahko dostopaš do elementov te strukture preko kazalca.
61. V C++ deklariraj strukturo in dinamično ustvari primerek te strukture. Nato pokaži, na katere vse načine lahko dostopaš do elementov te strukture preko kazalca. Kako sprostiš pomnilnik?
62. V C++ deklariraj poljubno strukturo in poljubno veliko polje primerkov te strukture. Nato prikaži, kako se skličeš na strukturo, ki je v polju na drugem mestu.
63. V C++ deklariraj strukturo, ki bo vsebovala tudi neko polje. Deklariraj še dinamični primerek te strukture in prikaži, kako se skličeš na drugi element polja.
64. Primerjaj polje struktur s strukturo, ki vsebuje polje.
65. Kaj so **unije** v C++? Kako jih deklariramo in kako uporabljamo? Razlago opremi s skico.
66. V C++ deklariraj strukturo, ki bo vsebovala niz znakov dolžine 10 in celo število. Nato deklariraj še unijo, ki bo imela enaka elementa. Razloži razliko.
67. Kaj so **bitna polja** v C++? Kako jih deklariramo in kako uporabljamo? Kaj je njihova prednost pred običajnimi strukturami?
68. Razloži razliko med **dinamičnim** in **statičnim** dodeljevanjem pomnilnika v C++. Razlago pospremi s skico pomnilnika.

69. Kaj so prednosti **dinamičnega** dodeljevanja pomnilnika pred statičnim? Na kaj je treba biti pri dinamičnih spremenljivkah še posebej pozoren?
70. Kaj veš o **življenjski dobi** dinamičnih spremenljivk? Kako v C++ sprostim pomnilnik, če ga zaseda enostavna dinamična spremenljivka in kako, če ga zaseda dinamično polje?
71. Ustvari **dinamično** spremenljivko v C++. Skiciraj stanje v pomnilniku. Izpiši naslov dinamične spremenljivke in naslov kazalca nanjo. Ali se kazalec in sama spremenljivka nahajata v istem delu pomnilnika?
72. Kaj je to **rekurzija**? Na kaj je potrebno biti pri rekurzivnih funkcijah še posebej pozoren? Pokaži, kako v C++ ustvarimo rekurzivno funkcijo.
73. Razloži razliko med tekstovnimi in binarnimi datotekami, ter med datotekami z zaporednim in datotekami z naključnim dostopom.
74. Kako v C++ datoteko odpremo za branje, kako za pisanje in kako za oboje? Kako preverjamo uspešnost odpiranja datoteke in kako datoteko zapremo?
75. Pokaži, kako v C++ odpreš tekstovno datoteko z zaporednim dostopom za pisanje in vanjo nekaj vpišeš. Katere funkcije lahko uporabiš za vpisovanje?
76. Pokaži, kako v C++ odpreš tekstovno datoteko z zaporednim dostopom za branje in iz nje nekaj prebereš. Katere funkcije lahko uporabiš za branje?
77. Razloži, kaj je **razred** in kaj **objekt**. Na primeru pokaži, kako v C++ ustvarimo enega in drugega? Demonstriraj uporabo metod razreda v C++.
78. Kaj so **konstruktorji** in kaj **destruktorji**? Katere vrste konstruktorjev poznamo v C++? Kdaj se odločimo za lastne konstruktorje in kdaj za destruktor? Kaj pa, če ne napišemo nobenega lastnega konstruktorja?
79. V C++ napiši **razred**, ki bo imel več konstruktorjev. Vsaj eden naj ima privzete argumente.
80. Kaj so **statični** elementi razreda? Po čem se razlikujejo od nestatičnih? Kdaj uporabimo statične metode in kdaj statične spremenljivke in kako jih deklariramo? Pokaži, kako v C++ ustvarimo in uporabimo statično spremenljivko in statično metodo.
81. Opiši **dedovanje** v C++ in ga demonstriraj na primeru.
82. Opiši postopek razvrščanja z **izbiranjem**: predstavi osnovno idejo, napiši algoritem in prikaži korake sortiranja na primeru sedmih števil (števila si izmisli tako, da ne bodo urejena).
83. Opiši postopek razvrščanja z **vstavljanjem**: predstavi osnovno idejo, napiši algoritem in prikaži korake sortiranja na primeru sedmih števil (števila si izmisli tako, da ne bodo urejena).
84. Opiši postopek razvrščanja s **premenami (bubblesort)**: predstavi osnovno idejo, napiši algoritem in prikaži korake sortiranja na primeru sedmih števil (števila si izmisli tako, da ne bodo urejena).