**Programmeerimine II**

*Küsimused eksamiks*

1. Algoritmi mõiste ja esitamise viisid.
2. Arvusüsteemid.Arvude teisendamine ühest arvusüsteemist teise.
3. Keeled.Algoritmilise keele mõiste.
4. Translaatorid, nende tüübid ja käigud.
5. Süntaksdiagramid keele kirjeldamiseks.
6. Programmi projekteerimise strateegiad: ülevalt alla ja alt üles.
7. Programmi “elutsükkel” arvutis. Progammi silumine.
8. C-keele iseloomustus.
9. C-programmi üldstruktuur.
10. Preprotsessori direktiivid.
11. Staatiline mälujaotus. Deklaratsioonid.
12. Andmetüübid. Üleminek ühest tüübist teise.
13. Standard funktsioonid.
14. Tehted ja nende prioriteedid. Avaldise tüüp.
15. C-keele põhilaused.
16. Programmi hargnemise programmeerimine.
17. Iteratsioonide (tsüklite) programmeerimine.
18. Viidad ja aadressid.
19. Massiivide deklareerimine ja nendega opereerimine.
20. Kahedimensionaalse massiivi esitamine ühedimensionaalsena ja vastupidi.
21. Massiivi järjestamise algoritm.
22. Kirjete (struktuuride) deklareerimine ja nendega opereerimine.
23. Failide tüübid, nimed ja olekud. Faili puhver. Failidega opereerimine.
24. Faili järjestamise algoritm.
25. Funktsioonid kui alamprogrammid. Lokaalsed ja globaalsed muutujad, negatiivne kõrvalefekt.Prototüübid.
26. Funktsiooni deklareerimine ja tema poole pöördumine. Positiivne kõrvalefekt.
27. ***Void****-*funktsiooni deklareerimine ja tema poole pöördumine.
28. Rekursiooni mõiste. Otsene- ja kaudne rekursioon. Nõudmised rekursiivsele funktsioonile.
29. Dünaamiline mälujaotus. Mälu eraldamise ja -vabastamise funktsioonid.
30. Pinu- (stacki-) ja järjekorra struktuur.
31. Välismoodulid. Koos- ja eraldi kompileerimine.
32. Programmi silumine. Veatöötlus.

1622/5000

1. Definition and presentation of the algorithm.
2. Numerical systems. Convert numbers from one number system to another.
3. Languages. The concept of algorithmic language.
4. Translators, types and passages.
5. Syntax diagrams to describe the language.
6. Program design strategies: top to bottom and bottom to top.
7. The "life cycle" of the program on the computer. Program debugging.
8. Characterization of the C language.
9. General structure of the C program.
10. Preprocessor Directives.
11. Static memory allocation. Declarations.
12. Data types. Switching from one type to another.
13. Standard Functions.
14. Actions and their priorities. Expression type.
15. Basic phrases in C language.
16. Programming branchies.
17. Programming iterations (cycles).
18. References and addresses(pointers).
19. Declaration and operation of arrays.
20. Representing a two-dimensional array in a one-dimensional and vice versa.
21. Array Sorting Algorithm.
22. Declaring and operating records (structures).
23. File types, names, and statuses. File buffer. Managing files.
24. File sorting algorithm.
25. Functions as subroutines. Local and global variables, negative side effect. Prototypes.
26. Declaring and approaching a function. Positive side effect.
27. Declaring a Void Function and Addressing it.
28. The concept of recursion. Direct and indirect recursion. Requirements for recursive function.
29. Dynamic memory allocation. Memory allocation and release functions.
30. Stack and queue structure.
31. External modules. Compile and separate compilation.
32. Program debugging. Error handling.