



Academia de Studii Economice
Departamentul de Informatică și Cibernetică Economică

Calea Dorobanți, 15-17, Sector 1, București, 010552 (camera 2314)

Tel.: +40 21 319 19 00, ext. 319, 336, Fax: +40 21 311 20 66

www.dice.ase.ro

Contest Topics for Professor
Position 31, year 2024-2025, semester 2

Disciplines: Programming Techniques and Algorithms; Basics of Programming

Programming Techniques and Algorithms

1. External data organization as files
2. Text files. Sequential binary files.
3. Relative binary files.
4. Indexed binary files.
5. Algorithm complexity. Divide et Impera method.
6. Sorting algorithms: quick sort, Shell sort, counting sort, radix sort, bucket sort.
Randomizing quick sort.
7. Greedy method.
8. Searching in solution space. Backtracking method.
9. Graphs. Definitions, representations and traversing.
10. Connectivity. Paths.
11. Applications of graph traversing: edge classification, detecting cut nodes, topological sorting.
12. Weighted graphs. Shortest (lowest cost) paths.
13. Tree graphs. Minimum spanning trees: Kruskal and Prim algorithms.

Bibliography:

1. I.Gh. Roșca, B. Ghilic-Micu, C. Cocianu, M. Stoica, C. Uscatu, M. Mircea, Programarea calculatoarelor. Algoritmi în programare, ASE, București, 2007, România
2. C. Uscatu, M. Popa, L. Pocatilu (Bătăgan), C. Silvestru, Programarea calculatoarelor. Aplicații, ASE, București, 2012, România
3. Thomas H. Cormen, Charles E. Leiserson, Ronald R. Rivest, Introducere în algoritmi, Computer Libris Agora, 2000, România
4. D. Knuth, Arta programării calculatoarelor(vol. 1-3), Teora, 2001, România
5. C. Uscatu, C. Cocianu, B. Ghilic-Micu, M. Stoica, M. Mircea, Algoritmi și tehnici de programare, ASE, 2015, România
6. C. Uscatu, C. Cocianu, M. Mircea, L. Pocatilu, Algoritmi și tehnici de programare. Aplicații, ASE, 2015, România



Basics of Programming

1. Algorithms - The role and characteristics of algorithms in the problem-solving process; Iterative and recursive; Example.
2. Algorithms - representation of algorithms (logical, pseudocode, analytical and tree); Description of the fundamental structures: linear structure, alternative structures and repetitive structures; Examples.
3. Algorithms - practical methods of structuring algorithms; errors in algorithms; design of algorithms; verifying the correctness of the algorithms; analysis of algorithms; Examples.
4. Internal organization and representation of data - information, data, knowledge - specific concepts and approaches; data and data structure; static data structures; Examples.
5. Organization and internal representation of data - dynamic data structures (basic concepts of graphs, lists, stacks and queues); Internal representation of data. Examples.
6. Stages of problem solving with computer and C programming language - General characteristics of automated data processing problems (ADPP); organizing the process of solving ADPP; General characteristics of C language - Basic language constructions (identifiers, comments, instructions, functions, program, and pre-processing directives). Examples
7. Subprograms - Subprogram building and call; Data transfer between caller and called (parameter transfer, global variable transfer); Examples
8. Data types in C - Simple data types, constants, structured data types (arrays, structures); Examples.
9. Input / output operations with keyboard / monitor in C - Format descriptors; Writing and reading functions with format; Input and output functions without format; Examples.
10. Expressions in C - Operators and Operators - arithmetic operators; logical and relational operators; bit-level operators; comma operator; the explicit conversion operator; size operator; parent brackets; the conditional operator; other operators; evaluation of phrases. Examples.
11. Achievement of fundamental control structures in C language - types of instructions; simple instructions; the composite statement; structured instructions; unconditional jump instructions and forced exit from structures. Examples.
12. Dynamic data types - Pointers - declaring and initializing pointers; use of pointers; pointer operations, link between pointers and arrays, dynamic memory allocation, const modifier, command line parameter handling. Examples.
13. Subprograms in C language - statement and use of procedures and functions; standard subprograms; C language libraries; calling programs. Examples.



Academia de Studii Economice
Departamentul de Informatică și Cibernetică Economică

Calea Dorobanți, 15-17, Sector 1, București, 010552 (camera 2314)

Tel.: +40 21 319 19 00, ext. 319, 336, Fax: +40 21 311 20 66

www.dice.ase.ro

Bibliography:

1. B. Ghilic-Micu si colectiv, *Bazele programării calculatoarelor. Suport de seminar*, ASE, Bucuresti, 2013
2. B. Ghilic si colectiv, *Algoritmi si scheme logice cu exemplificare in C*, ASE, Bucuresti, 2017
3. Brian W. Kernigham, Dennis M. Ritchie, *The C Programming Language (ANSI C)*, Prentice Hall Software Series, Second Edition, ISBN 978-0131103627, Pearson,1988.
4. Roland Backhouse, *Algorithmic Problem Solving*, John Wiley, ISBN 978-0-470-68453-5, 2011.