Bine ați venit la cursul **Tehnici de programare - TP**



Cursul 1 – Introducere. Matrici Ş.l. dr. ing. Cătălin Iapă catalin.iapa@cs.upt.ro

Ne cunoaștem deja:

Ne cunoaștem

deja:

Curs:



Cătălin Iapă

Ne cunoaștem

deja:

Curs:



Cătălin Iapă

Laboratoare și proiecte:

Beatrice Toader
Alin Stanciu

Detalii administrative curs

- Semestrul 2: 14 săptămâni
 - 2 ore de curs/ săptămână
 - 2 ore de laborator/ săptămână
 - 2 ore de proiect/ 2 săptămâni
- Veți primi 2 note, media lor este nota finală la LSD
 - 1 notă la examen, în sesiune, după cele 14 săptămâni
 - 1 notă pentru activitatea de la laborator
- Pentru a promova materia e nevoie să luați cel puțin nota 5, atât la examen cât și la laborator
- Examenul se poate da de 3 ori



Ce facem la TP

Date în memoria calculatorului Matrici Matrici alocate dinamic Exerciții

Despre TP

Tehnici de Programare

- Ce vom face?
 - Programare în limbajul C

- De ce ai nevoie înainte să începi acest curs?
 - Noțiunile de programre asimilate la materia
 Programarea Calculatoarelor din Semestrul 1

Despre TP

- 1. Matrici, inclusiv alocare dinamica matrici
- 2. Eficienta algoritmilor. Functii de timp si numere aleatoare
- 3. Pointeri la functii. qsort, bsearch
- 4. Structuri. Structuri cu campuri pe biti
- 5. Functii cu numar variabil de argumente
- 6. Liste simplu inlantuite
- 7. Liste dublu inlantuite

Despre TP

- 8. Recursivitate
- 9. Divide et Impera
- 10. Greedy
- 11. Backtraking
- 12. Tehnici de cautare si sortare

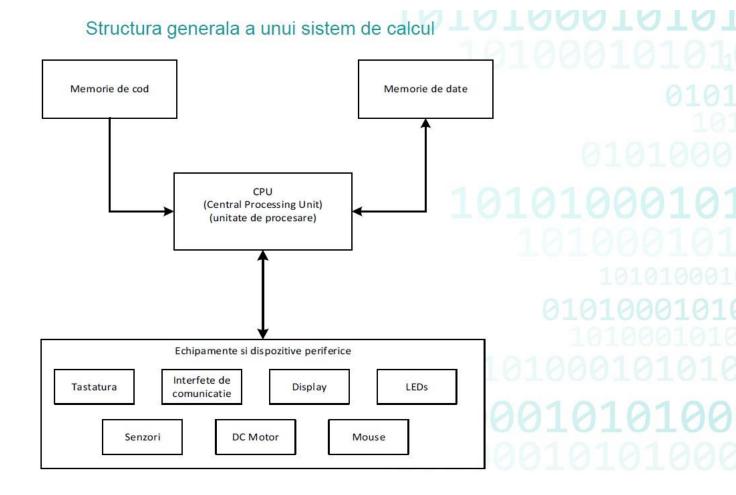


Ce facem la TP

Date în memoria calculatorului

Matrici Matrici alocate dinamic Exerciții

Date.



Date.

- Ce poate reține un calculator în memorie?
 - Instrucțiuni
 - Date

Ce fel de date poate stoca un calculator?

- Numere
- Litere (caractere)
- Culori
- Sunete

Date.

- Cum stochează un calculator datele?
 - Sub formă de numere binare

Numere

123 0111 1011

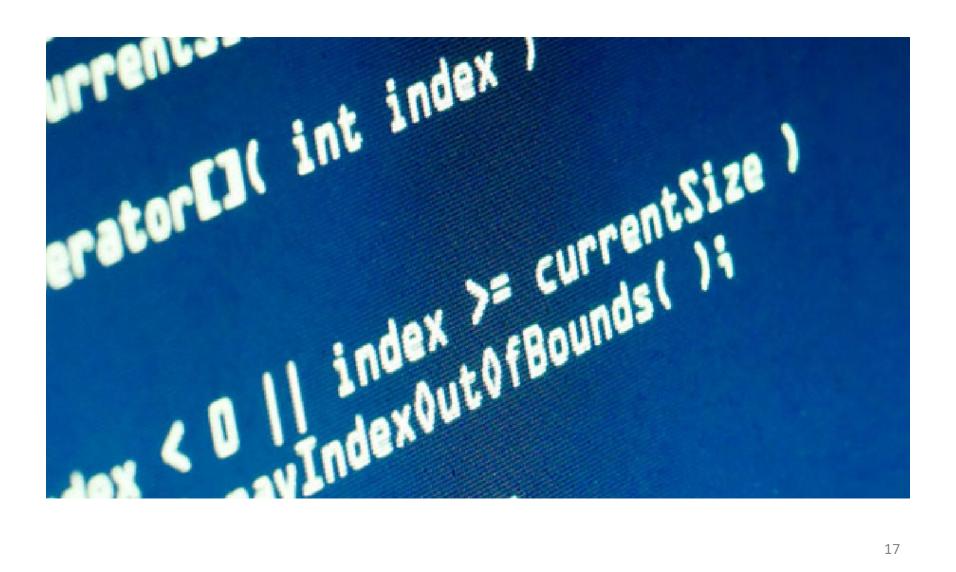
-7 1000 0111

Litere (caractere) – tabel ASCII

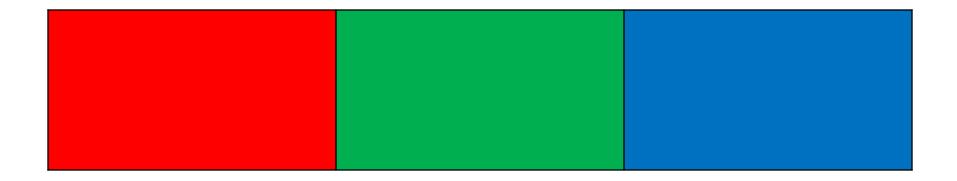
| 0 0 000 NUL (null) | Dec | Нх | Oct | Chai | r | Dec | Нх | Oct | Html | Chr | Dec | Нх | Oct | Html | Chr | Dec | : Hx | Oct | Html Cl | <u>nr</u> |
|--|-----|----|-----|------|--------------------------|-----|----|-----|----------------|-------|-----|----|-----|--------------|-----|-----|------|-----|----------------|-----------|
| 2 2 002 STX (start of text) 3 3 003 ETX (end of text) 3 62 043 & #345; # 4 004 EOT (end of transmission) 5 5 005 ENQ (enquiry) 6 6 006 ACK (acknowledge) 7 7 007 BEL (bell) 8 8 010 BS (backspace) 9 9 011 TAB (horizontal tab) 10 A 012 LF (NL line feed, new place) 11 B 013 VT (vertical tab) 12 C 014 FF (NP form feed, new page) 13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 11 022 DC2 (device control 1) 19 13 023 DC3 (device control 1) 19 13 023 DC3 (device control 1) 19 14 024 DC4 (device control 4) 20 15 NAK (negative acknowledge) 21 16 026 STM (synchronous idle) 22 16 026 STM (synchronous idle) 23 17 027 ETB (end of trans. block) 25 19 031 EM (end of medium) 25 10 035 GS (group separator) 26 16 036 RS (record separator) 27 10 035 GS (group separator) 28 1C 034 FS (file separator) 29 1D 035 GS (group separator) 20 1E 036 RS (record separator) 20 1E 040 A04 Q44 Q44 Q44 Q44 Q44 Q44 Q44 Q44 Q44 Q | 0 | 0 | 000 | NUL | (null) | 32 | 20 | 040 | @#32; | Space | 64 | 40 | 100 | a#64; | 0 | 96 | 60 | 140 | a#96; | 8 |
| 3 3 003 ETX (end of text) | 1 | 1 | 001 | SOH | (start of heading) | 33 | 21 | 041 | @#33; | 1 | 65 | 41 | 101 | A | A | 97 | 61 | 141 | & # 97; | a |
| 4 4 004 EOT (end of transmission) 36 24 044 4#36; \$ 5 005 ENQ (enquiry) 37 25 045 6#37; \$ 6 6 006 ACK (acknowledge) 38 26 046 6#36; \$ 7 7 007 BEL (bell) 39 27 047 6#39; \$ 7 7 007 BEL (bell) 39 27 047 6#39; \$ 8 8 010 BS (backspace) 40 28 050 6#40; (9 9 011 TAB (horizontal tab) 41 29 051 6#41;) 10 A 012 LF (MI line feed, new line) 11 B 013 VT (vertical tab) 42 2A 052 6#42; \$ 12 C 014 FF (NP form feed, new page) 13 D 015 CR (carriage return) 45 2D 055 6#45; - 13 D 015 CR (carriage return) 46 2E 056 6#46; . 14 E 016 SO (shift out) 46 2E 056 6#46; . 15 F 017 SI (shift in) 47 2F 057 6#47; / 16 10 020 DLE (data link escape) 48 30 060 6#40; 0 17 11 021 DC1 (device control 1) 49 31 061 6#49; 1 18 12 022 DC2 (device control 2) 50 32 062 6#50; 2 20 14 024 DC4 (device control 4) 52 34 064 6#52; 4 21 15 025 NAK (negative acknowledge) 53 35 065 6#55; 5 20 14 024 DC4 (device control 4) 52 34 064 6#55; 7 23 17 027 ETB (end of trans. block) 56 38 070 6#55; 8 24 18 030 CAN (cancel) 57 39 071 6#55; 7 25 19 031 EM (end of medium) 57 39 071 6#57; 9 26 18 032 SUB (substitute) 58 3A 072 6#56; 8 27 18 033 ESC (escape) 59 3B 073 6#59; 9 28 1C 034 FS (file separator) 66 3E 076 6#61; 9 37 25 045 6#37; 8 36 046 6#30; 6 37 25 045 6#37; 8 36 046 6#30; 6 37 047 6#30; 6 37 047 6#30; 6 37 047 6#30; 6 38 102 6#40; 1 39 27 047 6#30; 1 39 27 047 6#30; 1 39 27 047 6#30; 1 39 27 047 6#30; 1 39 27 047 6#30; 1 39 27 047 6#30; 1 39 27 047 6#30; 1 39 27 047 6#30; 1 39 27 047 6#30; 1 30 28 044 110 6#72; H 102 68 110 6#102; H 105 69 151 6#102 | 2 | 2 | 002 | STX | (start of text) | 34 | 22 | 042 | @#3 4 ; | rr | 66 | 42 | 102 | B | В | 98 | 62 | 142 | %#98; | b |
| 5 5 005 ENQ (enquiry) 6 6 006 ACK (acknowledge) 7 7 007 BEL (bell) 8 8 010 BS (backspace) 9 9 011 TAB (horizontal tab) 10 A 012 LF (NL line feed, new line) 11 B 013 VT (vertical tab) 12 C 014 FF (NP form feed, new page) 13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 25 19 031 EM (end of medium) 26 10 035 GS (group separator) 27 19 10 035 GS (group separator) 28 25 045 % * 46 046 ! * 47 07 446 106 F F 102 66 145 e e 70 446 106 F F 102 66 145 f f 70 446 106 F F 102 66 145 f f 71 47 107 G G 103 67 147 g g 72 48 110 G G 103 67 147 g g 73 49 111 I I 105 69 151 i i 10 6 A 152 j j 74 44 112 J J 106 64 152 j j 75 44 112 J J 106 64 152 k k 75 4B 113 K K 107 6B 153 k k 76 4C 114 L L 108 6C 154 l L 77 4D 115 M M 109 6D 155 m m 78 4E 116 N N 10 6E 156 n n 79 4F 117 O O 111 6F 157 n n 79 4F 117 O O 111 6F 157 n n 79 4F 117 O O 111 6F 157 o O 80 50 120 P 112 70 160 p P 80 14 024 DC4 (device control 4) 81 51 121 Q O 113 71 161 q G 81 60 60 50 60 60 60 60 60 60 60 60 60 60 60 60 60 | 3 | 3 | 003 | ETX | (end of text) | 35 | 23 | 043 | # ; | # | 67 | 43 | 103 | <u>4#67;</u> | C | | | | | |
| 38 26 046 & & 70 46 106 F F 102 66 146 f f 7 7 007 BEL (bell) 39 27 047 ! 71 47 107 G G 103 67 147 g g 8 8 010 BS (backspace) 40 28 050 ((| 4 | 4 | 004 | EOT | (end of transmission) | 36 | 24 | 044 | @#36; | ş | ı | | | | | | | | | |
| 7 7 007 BEL (bell) 39 27 047 6#39; 71 47 107 6#71; 6 103 67 147 6#103; 9 8 8 010 BS (backspace) 40 28 050 6#40; 72 48 110 6#72; H 104 68 150 6#104; h 105 6#105; i 10 A 012 LF (NL line feed, new line) 42 2A 052 6#42; 7 74 4A 112 6#74; J 106 6A 152 6#106; j 11 B 013 VT (vertical tab) 43 2B 053 6#43; + 75 4B 113 6#75; K 107 6B 153 6#107; k 12 C 014 FF (NP form feed, new page) 44 2C 054 6#44; 75 4B 113 6#75; K 106 6A 152 6#106; j 13 D 015 CR (carriage return) 45 2D 055 6#46; - 76 4C 114 6#76; L 108 6C 154 6#108; L 108 6C 154 | 5 | 5 | 005 | ENQ | (enquiry) | | | | | | | | | | | | | | | |
| 8 8 010 BS (backspace) 9 9 011 TAB (horizontal tab) 10 A 012 LF (NL line feed, new line) 11 B 013 VT (vertical tab) 12 C 014 FF (NF form feed, new page) 13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 24 18 030 CAN (cancel) 25 19 031 EM (end of medium) 26 1A 032 SUB (substitute) 27 1B 033 ESC (escape) 28 1C 034 FS (file separator) 30 1E 036 RS (record separator) 40 28 050 «#40; (72 48 110 «#72; H 105 69 151 «#104; h 105 69 151 «#105; i 105 64; los 441;) 73 49 111 «#73; I 105 69 151 «#105; i 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 154 «#108; l 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 154 «#108; l 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 (a #73; I 106 6A 152 «#106; j 174 44 11 | 6 | 6 | 006 | ACK | (acknowledge) | | | | | | 70 | | | | | | | | | |
| 9 9 011 TAB (horizontal tab) 10 A 012 LF (NL line feed, new line) 11 B 013 VT (vertical tab) 12 C 014 FF (NP form feed, new page) 13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 23 17 027 ETB (end of medium) 25 19 031 EM (end of medium) 26 1A 032 SUB (substitute) 27 1B 033 ESC (escape) 28 1C 034 FS (file separator) 30 1E 036 RS (record separator) 30 1E 036 RS (record separator) 41 29 051 4#41;) 42 2A 052 4#42; * 43 2B 053 4#42; * 44 22 A 052 4#42; * 44 41: | 7 | | | | (bell) | | | | | | | | | | | | | | | |
| 10 A 012 LF (NL line feed, new line) 11 B 013 VT (vertical tab) 12 C 014 FF (NP form feed, new page) 13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 23 17 027 ETB (end of medium) 25 19 031 EM (end of medium) 26 10 035 GS (group separator) 27 10 035 GS (group separator) 28 10 05 6 446; 48 20 055 6445; 49 20 055 6445; 40 20 55 6446; 41 20 55 6446; 42 20 55 6446; 43 20 055 6445; 44 20 055 6446; 45 20 055 6446; 46 22 056 6446; 47 2F 057 647; 48 113 6477; M 109 6D 155 64109; m 100 6E 156 64110; n 100 6E 156 64111; n 100 6E 156 6411; n 100 6E 156 6411; n 100 6E 156 | 8 | | | | | | | | | | | | | | | | | | | |
| 11 B 013 VT (vertical tab) 12 C 014 FF (NP form feed, new page) 13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 24 18 030 CAN (cancel) 25 19 031 EM (end of medium) 26 1A 032 SUB (substitute) 27 1B 033 ESC (escape) 28 1C 034 FS (file separator) 30 1E 036 RS (record separator) 43 2B 053 «#43; + 44 2C 054 «#44; , 76 4C 114 «#76; L 108 6C 154 «#108; l 109 6D 155 «#109; m 110 6E 156 «#109 | 9 | | | | (horizontal tab) | | | | | | | | | | | | | | | |
| 12 C 014 FF (NP form feed, new page) 13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 23 17 027 ETB (end of medium) 24 18 030 CAN (cancel) 25 19 031 EM (end of medium) 26 1A 032 SUB (substitute) 27 1B 033 ESC (escape) 38 1C 034 FS (file separator) 44 2C 054 «#44; , | 10 | | | | | | | | | | | | | | | | | | | |
| 13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 23 17 027 ETB (end of medium) 24 18 030 CAN (cancel) 25 19 031 EM (end of medium) 26 1A 032 SUB (substitute) 27 1B 033 ESC (escape) 28 1C 034 FS (file separator) 30 1E 036 RS (record separator) 30 1E 036 RS (record separator) 45 2D 055 «#45; - | | | | | (vertical tab) | | | | | | | | | | | | | | | |
| 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 23 17 027 ETB (end of medium) 24 18 030 CAN (cancel) 25 19 031 EM (end of medium) 26 1A 032 SUB (substitute) 27 1B 033 ESC (escape) 28 1C 034 FS (file separator) 29 1D 035 GS (group separator) 30 1E 036 RS (record separator) 46 2E 056 . . 78 4E 116 N N 110 6E 156 n n 79 4F 117 O 0 111 6F 157 o 0 80 50 120 P P 112 70 160 p P 81 51 121 Q Q 82 52 122 R R 81 51 121 Q Q 83 53 123 S S 83 53 123 S S 83 53 123 S S 84 54 124 T T 85 54 126 t t 86 56 126 V V 87 57 127 W W 88 58 130 X X 89 59 131 Y Y 80 50 120 P P 80 50 120 P P 80 50 120 P P 81 112 70 160 p P 81 51 121 Q Q 82 52 122 R R 83 53 123 R U 84 54 124 T T 85 54 126 r t 86 56 126 V V 87 57 127 W W 88 58 130 X X 89 59 131 Y Y 80 50 120 P P 80 50 120 P P 81 112 70 160 p P 81 51 121 Q Q 82 52 122 R R 83 53 123 R U 83 53 123 S U 83 53 123 S U 83 53 123 S U 84 54 124 T T 85 54 126 V V 86 56 126 V V 87 57 127 W W 88 58 130 X X 89 59 131 Y Y 89 59 131 & | 12 | | | | (NP form feed, new page) | | | | | | | | | | | | | | | |
| 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 24 18 030 CAN (cancel) 25 19 031 EM (end of medium) 26 1A 032 SUB (substitute) 27 1B 033 ESC (escape) 28 1C 034 FS (file separator) 30 1E 036 RS (record separator) 47 2F 057 c#47; / 48 30 060 c#48; 0 49 31 061 c#49; 1 81 51 121 c#81; 0 113 71 161 c#113; 0 114 72 162 c#114; r 151 33 063 c#50; 2 82 52 122 c#82; R 114 72 162 c#114; r 116 74 164 c#116; t 88 55 125 c#85; U 117 75 165 c#117; u 88 55 126 c#86; V 118 76 166 c#118; v 89 59 131 c#89; V 119 77 167 c#119; w 89 59 131 c#89; V 120 78 170 c#120; x 120 78 170 c#121; V 121 79 171 c#121; V 122 77 18 033 ESC (escape) 123 071 c#57; 9 124 70 174 c#124; I 125 70 175 c#125; } 126 7E 176 c#126; ~ | | | | | | | | | | | | | | | | | | | | |
| 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 24 18 030 CAN (cancel) 25 19 031 EM (end of medium) 26 1A 032 SUB (substitute) 27 1B 033 ESC (escape) 28 1C 034 FS (file separator) 30 1E 036 RS (record separator) 30 1E 036 RS (record separator) 30 1E 036 RS (record separator) 48 30 060 c#48; 0 49 31 061 c#49; 1 50 32 062 c#50; 2 49 31 061 c#49; 1 50 32 062 c#50; 2 49 31 061 c#49; 1 50 32 062 c#50; 2 49 31 061 c#49; 1 50 32 062 c#50; 2 49 31 061 c#49; 1 50 32 062 c#50; 2 49 31 061 c#49; 1 50 32 062 c#50; 2 48 2 52 122 c#82; R 114 72 162 c#114; L 115 73 163 c#115; S 116 74 164 c#116; L 117 75 165 c#111; U 117 75 165 c#112; U 117 75 166 c#112; U 117 75 166 c#112; U 118 70 160 c#112; D 119 17 161 c#112; D 110 04 04 05 c#10; C 110 c#10 c#10 c#10; C 111 07 160 c#112; D 112 70 160 c#112; D 113 71 161 c#113; Q 114 021 c#10 c#10 c#10; C 114 022 c#10 c#10; C 115 025 c#10; C 116 026 c#10; C 117 75 165 c#115; S 117 027 c#88; U 117 75 165 c#117; U 117 75 166 c#118; V 118 76 166 c#118; V 119 77 167 c#119; W 119 77 167 c#119; W 119 77 167 c#119; W 110 021 c#10; C 110 c#10; C 110 c#10; C 111 021 c#10; C 112 020 c#10; C 113 021 c#80; C 114 022 c#10; C 115 025 c#10; C 116 026 c#10; C 117 026 c#10; C 118 030 c#10; C 118 031 c#10; C 118 031 c#10; C 119 031 c#10; C 110 c#1 | 14 | | | | (shift out) | | | | | | | | | | | | | | | |
| 17 11 021 DC1 (device control 1) | | | | | | | | | | | | | | | | | | | | |
| 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 24 18 030 CAN (cancel) 25 19 031 EM (end of medium) 26 1A 032 SUB (substitute) 27 1B 033 ESC (escape) 28 1C 034 FS (file separator) 30 1E 036 RS (record separator) 30 1E 036 RS | | | | | | | | | | | | | | | | | | | | |
| 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 24 18 030 CAN (cancel) 25 19 031 EM (end of medium) 26 1A 032 SUB (substitute) 27 1B 033 ESC (escape) 28 1C 034 FS (file separator) 30 1E 036 RS (record separator) 30 1E 036 RS (| | | | | | | | | | | | | | | | | | | | |
| 20 14 024 DC4 (device control 4) 52 34 064 4 4 84 54 124 T T 116 74 164 t t 21 15 025 NAK (negative acknowledge) 53 35 065 5 5 85 55 125 U U 117 75 165 u u 22 16 026 SYN (synchronous idle) 54 36 066 6 6 86 56 126 V V 118 76 166 v V 23 17 027 ETB (end of trans. block) 55 37 067 7 7 87 57 127 W W 119 77 167 w W 24 18 030 CAN (cancel) 56 38 070 8 8 8 58 130 X X 120 78 170 x X 25 19 031 EM (end of medium) 57 39 071 9 9 89 59 131 Y Y 121 79 171 y Y 26 1A 032 SUB (substitute) 58 3A 072 : : 90 5A 132 Z Z 122 7A 172 z Z 27 1B 033 ESC (escape) 59 3B 073 ; : 91 5B 133 [[123 7B 173 { { 29 1D 035 GS (group separator) 61 3D 075 = = 93 5D 135]] 125 7D 175 } } 30 1E 036 RS (record separator) 62 3E 076 > > 94 5E 136 ^ ^ 126 7E 176 ~ ~ | | | | | | | | | | | | | | | | | | | | |
| 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 24 18 030 CAN (cancel) 25 19 031 EM (end of medium) 26 1A 032 SUB (substitute) 27 1B 033 ESC (escape) 28 1C 034 FS (file separator) 30 1E 036 RS (record separator) 31 35 065 5 5 35 065 5 5 36 066 6 6 36 56 126 V V 36 66 56 126 W W 37 77 167 w W 38 58 130 X X 38 070 8 8 38 070 8 8 38 070 8 8 38 070 8 8 38 58 130 X X 38 070 X X 38 070 X X 39 071 9 9 39 071 9 9 39 071 9 9 39 071 9 9 39 071 9 9 39 071 E 9 39 59 131 Y Y 30 12 036 RS (group separator) 30 18 036 RS (record separator) 31 075 = = 32 076 > 9 34 08 58 125 U U 36 117 75 165 u U 37 167 v V 38 58 125 X U 38 76 126 V V 38 77 127 W W 39 79 131 Y Y 39 121 79 171 y Y 39 121 79 171 y Y 39 122 7A 172 z Z 39 123 7B 173 { { 39 125 7D 175 } } 30 18 036 RS (record separator) 30 28 076 > > 31 075 = = 32 076 > > 34 08 58 125 U U 38 08 58 125 U U 38 08 58 125 U U 38 08 58 125 V V 38 08 58 126 V V 38 08 59 127 W W 39 09 120 X X 30 08 08 08 08 08 08 08 08 08 08 08 08 08 | | | | | | | | | | | | | | | | | | | | |
| 22 16 026 SYN (synchronous idle) 54 36 066 6 6 86 56 126 V V 118 76 166 v V 23 17 027 ETB (end of trans. block) 55 37 067 7 7 87 57 127 W W 119 77 167 w W 24 18 030 CAN (cancel) 56 38 070 8 8 88 58 130 X X 120 78 170 x X 25 19 031 EM (end of medium) 57 39 071 9 9 89 59 131 Y Y 121 79 171 y Y 26 1A 032 SUB (substitute) 58 3A 072 : : 90 5A 132 Z Z 122 7A 172 z Z 27 1B 033 ESC (escape) 59 3B 073 ; ; 91 5B 133 [[123 7B 173 { { 28 1C 034 FS (file separator) 60 3C 074 < < 92 5C 134 \ \ 124 7C 174 29 1D 035 GS (group separator) 61 3D 075 = = 93 5D 135]] 125 7D 175 } } 30 1E 036 RS (record separator) 62 3E 076 > > 94 5E 136 ^ ^ 126 7E 176 ~ ~ | | | | | | | | | | | | | | | | | | | | |
| 23 17 027 ETB (end of trans. block) 55 37 067 7 7 87 57 127 W W 119 77 167 w W 24 18 030 CAN (cancel) 56 38 070 8 8 88 58 130 X X 120 78 170 x X 25 19 031 EM (end of medium) 57 39 071 9 9 89 59 131 Y Y 121 79 171 y Y 26 1A 032 SUB (substitute) 58 3A 072 : : 90 5A 132 Z Z 122 7A 172 z Z 27 1B 033 ESC (escape) 59 3B 073 ; ; 91 5B 133 [[123 7B 173 { { 28 1C 034 FS (file separator) 60 3C 074 < < 92 5C 134 \ \ 124 7C 174 29 1D 035 GS (group separator) 61 3D 075 = = 93 5D 135]] 125 7D 175 } } 30 1E 036 RS (record separator) 62 3E 076 > > 94 5E 136 ^ ^ 126 7E 176 ~ ~ | | | | | | | | | | | | | | | | | | | | |
| 24 18 030 CAN (cancel) 56 38 070 8 8 88 58 130 X X 120 78 170 x X 25 19 031 EM (end of medium) 57 39 071 9 9 89 59 131 Y Y 121 79 171 y Y 26 1A 032 SUB (substitute) 58 3A 072 : : 90 5A 132 Z Z 122 7A 172 z Z 27 1B 033 ESC (escape) 59 3B 073 ; ; 91 5B 133 [[123 7B 173 { { 28 1C 034 FS (file separator) 60 3C 074 < < 92 5C 134 \ \ 124 7C 174 29 1D 035 GS (group separator) 61 3D 075 = = 93 5D 135]] 125 7D 175 } } 30 1E 036 RS (record separator) 62 3E 076 > > 94 5E 136 ^ ^ 126 7E 176 ~ ~ | | | | | | | | | | | | | | | | | | | | |
| 25 19 031 EM (end of medium) 57 39 071 9 9 89 59 131 Y Y 121 79 171 y Y 26 1A 032 SUB (substitute) 58 3A 072 : 90 5A 132 Z Z 122 7A 172 z Z 27 1B 033 ESC (escape) 59 3B 073 ; 91 5B 133 [[123 7B 173 { { 28 1C 034 FS (file separator) 60 3C 074 < < 92 5C 134 \ \ 124 7C 174 29 1D 035 GS (group separator) 61 3D 075 = = 93 5D 135]] 125 7D 175 } } 30 1E 036 RS (record separator) 62 3E 076 > > 94 5E 136 ^ ^ 126 7E 176 ~ ~ | | | | | | | | | | | | | | | | | | | | |
| 26 1A 032 SUB (substitute) 58 3A 072 :: 90 5A 132 Z Z 122 7A 172 z Z 27 1B 033 ESC (escape) 59 3B 073 ;; 91 5B 133 [[123 7B 173 { { 28 1C 034 FS (file separator) 60 3C 074 < < 92 5C 134 \ \ 124 7C 174 29 1D 035 GS (group separator) 61 3D 075 = = 93 5D 135]] 125 7D 175 } } 30 1E 036 RS (record separator) 62 3E 076 > > 94 5E 136 ^ ^ 126 7E 176 ~ ~ | | | | | (cancel) | | | | | | | | | | | | | | | |
| 27 1B 033 ESC (escape) | | | | | | | | | | | | | | | | | | | | |
| 28 1C 034 FS (file separator) 60 3C 074 < < 92 5C 134 \ \ 124 7C 174 \ 29 1D 035 GS (group separator) 61 3D 075 = = 93 5D 135]] 125 7D 175 } \ 30 1E 036 RS (record separator) 62 3E 076 > > 94 5E 136 ^ ^ 126 7E 176 ~ ~ | | | | | | | | | | | | | | | | | | | | |
| 29 1D 035 GS (group separator) 61 3D 075 = = 93 5D 135]] 125 7D 175 } } 30 1E 036 RS (record separator) 62 3E 076 > > 94 5E 136 ^ ^ 126 7E 176 ~ ~ | | | | | (escape) | | | | | | ı | | | | | | | | | |
| 30 1E 036 RS (record separator) 62 3E 076 > > 94 5E 136 ^ ^ 126 7E 176 ~ ~ | | | | | (file separator) | | | | | | | | | | | | | | | |
| | | | | | (group separator) | | | | | | ı | | | | | | | | | |
| 31 1F 037 HS (unit separator) 63 3F 077 A#63: 2 95 5F 137 A#95: 127 7F 177 A#127: DEL | | | | | | | | | | | | | | | | | | | | |
| 51 11 55, 50 (white proparation) 55 51 50, 6850, _ 127 71 177 68127, 452 | 31 | 1F | 037 | US | (unit separator) | 63 | ЗF | 077 | ۵#63; | 2 | 95 | 5F | 137 | a#95; | _ | 127 | 7F | 177 | | PEL |

Source: www.LookupTables.com









Structura unui algoritm





Ce facem la TP
Date în memoria calculatorului
Matrici

Matrici alocate dinamic Exerciții

Matricile au aplicații numeroase. Multe probleme din teoria grafurilor, rețele neuronale, modelare sau grafică 3D au la bază matrici.

La Logică și Structuri Discrete - LSD am văzut cum putem reprezenta relații sau grafuri cu matrici:

Reprezentarea unei relații

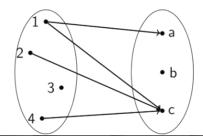
3. Ca matrice booleană/binară, dacă A, B finite,

- linii indexate după A, și coloanele după B

 $m_{xy} = 1 \operatorname{daca}(x, y) \in R$,

 $m_{xy} = 0 \, dac \, a \, (x, y) \notin R$

În practică putem folosi acest tip de reprezentare dacă A și B nu sunt foarte mari



| | a | b | С |
|---|---|---|---|
| 1 | 1 | 0 | 1 |
| 2 | 0 | 0 | 1 |
| 3 | 0 | 0 | 0 |
| 4 | 0 | 0 | 1 |

Reprezentarea grafurilor

Dacă identificăm nodurile prin numere (consecutive), putem reprezenta graful ca *matrice de adiacență* pătratică

M[i,j] = 1 dacă există muchie de la i la j

M[i,j] = 0 dacă *nu există* muchie de la i la j

sau M[i,j] poate conține lungimea/costul muchiei (graf ponderat)

În limbajul C se pot crea vectori de vectori.

Poartă denumirea de vectori multidimensionali.

Un vector bidimensional poartă denumirea de matrice.

int a[3][4];

La fel ca la vectori, în general, elementele unei matrici sunt variabile, în schimb, în C, matricea ca întreg nu poate fi tratată ca o variabilă.

Nu putem, de exemplu, să atribuim unei matrici o altă matrice cu operatorul de atribuire, =.

Putem în schimb să copiem element cu element în bucle repetitive (for etc.).

```
int a[3][4];
3 rânduri (de la 0 la 2),
4 coloane (de la 0 la 3),
  12 elemente (3x4)
```

Inițializarea elementelor unei matrici

int
$$a[2][3] = \{\{1, 3, 4\}, \{-3, 8, 12\}\};$$

| | 0 | 1 | 2 |
|---|----|---|----|
| 0 | 1 | 3 | 4 |
| 1 | -3 | 8 | 12 |

Accesarea elementelor unei matrici

```
int a[2][3];
```

```
a[0][0] = 7;
a[0][1] = 2;
a[0][2] = -3;
a[1][0] = 3;
a[1][1] = 8;
a[1][2] = 12;
```

| 0 | 1 | 2 |
|---|---|----|
| 7 | 2 | -3 |
| 3 | 8 | 12 |

```
printf("%d", a[1][1]);
scanf("%d", &a[1][0]);
```

Accesarea elementelor unei matrici

```
#define N 2
#define M 3
int a[N][M];
for(i=0; i<N;i++)
     for(i=0; i<M; j++)
                 printf("a[%d][%d] =");
                 scanf("%d", &a[i][j]);
```

Afișarea elementelor unei matrici

```
void afisare(int a[][M], int n, int m)
for(int i=0; i<n;i++)
            for(int j=0; j<m; j++)
                  printf("%3d ",a[i][j]);
            printf("\n");
```

Transmiterea ca parametrii

De regulă, în C, majoritatea variabilelor sunt transmise ca valoare la apelurile ca parametrii în funcții.

Matricile, la fel ca vectorii, nu respectă această regulă, ele sunt transmise ca parametrii prin referință, nu prin valoare.

Funcția care va prelucra ca parametru o matrice va primi matricea originală, nu o copie a acesteia.

Transmiterea ca parametrii

```
#include <stdio.h>
                                     Initial: a= 1, m[0][0]=1
                                     Final : a= 1, m[0][0]=100
void set_int(int a) {
 a=100;
void set_matrix (int m[2][3]) {
 m[0][0]=100;
int main() {
 int a=1;
 int m[2][3]={{1,2,3},{4,5,6}};
 printf("Initial: a= %d, m[0][0]=%d\n", a, m[0][0]);
 set_int(a);
 set matrix(m);
 printf("Final: a= %d, m[0][0]=%d", a, m[0][0]);
 return 0;
```

Vectori și pointeri în contextul matricilor

O matrice este un vector de vectori. Compilatorul va amplasa în memorie elementele matricii în ordinea liniilor, fără niciun spațiu între linii:

| a[0][0] | a[0][1] | *** | a[0][N-1] | a[1][0] | a[1][1] | *** | a[1][N-1] | 2889 | a[M- 2][0] | a[M- 2][1] | *** | a[M- 2][N-1] | a[M- 1][0] | a[M- 1][1] | | a[M- 1][N-1] |
|---------|---------|-----|-----------|---------|---------|-----|-----------|------|---------------|---------------|-----|-----------------|---------------|---------------|--|-----------------|
| linia 0 | | | | linia 1 | | | | | linia | M-2 | | | linia | M-1 | | |

Adresa de memorie a elementului a[i][j]:

$$&a[i][j] = &a[0][0] + i*N + j$$

- &a[0][0] este adresa de început a matricii,
- i*N este nr. de elem. conținut în liniile anterioare
- j este nr. de elem. până la indexul coloanei curente

Vectori și pointeri în contextul matricilor

| a[0][0] | a[0][1] | *** | a[0][N-1] | a[1][0] | a[1][1] | 6.55.5° | a[1][N-1] | 2000 | a[M- 2][0] | a[M- 2][1] | 6153 | a[M- 2][N-1] | a[M- 1][0] | a[M- 1][1] | | a[M- 1][N-1] | |
|---------|---------|-----|-----------|---------|---------|---------|-----------|------|---------------|---------------|------|-----------------|---------------|-------------------------|--|-----------------|--|
| | linia 0 | | | | linia | a 1 | | | | linia M-2 | | | | linia <mark>M</mark> -1 | | | |

Matricea se poate parcurge cu un pointer cu un singur for, folosindu-ne de modul în care aceasta este reținută în memorie:

```
int *p;
for(p=&a[0][0]; p<&a[m][0]; ++p)
*p=0;
```

Vectori și pointeri în contextul matricilor

```
În cele mai multe cazuri, când declarăm o matrice static, se va folosi doar o porțiune din matrice.
int main() {
  int a[10][10], nr_linii, nr_coloane;

  scanf("%d",&nr_linii);
  scanf("%d",&nr_coloane);
  // daca se citesc nr_linii=2 si nr_coloane=3

  for(int i=0; i<nr_linii; i++)
      for(int j=0;j<nr_coloane;j++)
      alillii-i+i:
```

```
a[i][j]=i+j;

afisare(a, nr_linii, nr_coloane);

printf("\n");

afisare(a, 10, 10);

return 0;
```



Ce facem la TP
Date în memoria calculatorului
Matrici

Matrici alocate dinamic

Exerciții

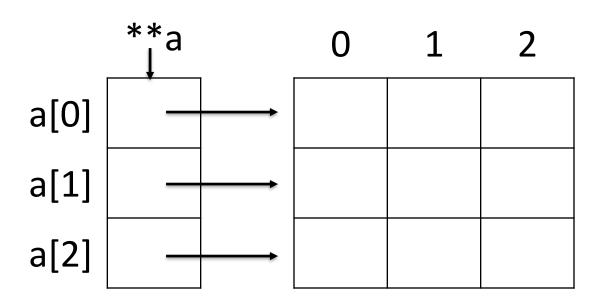
La fel ca vectorii, și matricile se pot aloca dinamic.

Avantajul folosirii matricilor alocate dinamic este că se pot redimensiona în funcție de necesități pe parcursul executării programului.

Putem să folosim un vector de pointeri, fiecare pointer pointând la câte o linie din matrice.

În acest caz liniile matricii vor fi alocate separat, ca blocuri de memorie independente.

Matricea va fi definită ca int **a
a[i] este pointerul (int*) de la indexul i din a, adică
adresa memoriei care conține acea linie



```
O
                                                                2
int m,n,i,j;
int **a;
                                 a[0]
                                 a[1]
printf("m=");scanf("%d",&m);
printf("n=");scanf("%d",&n);
                                 a[2]
a=(int**)malloc(m*sizeof(int*));
if(a==NULL)
       printf("memorie insuficienta\n");
       exit(EXIT_FAILURE);
```

```
// alocare linii din matrice
                                                    O
                                                                2
                                 a[0]
for(i=0;i<m;i++)
                                 a[1]
a[i]=(int*)malloc(n*sizeof(int));
                                 a[2]
 if(a[i]==NULL)
        for(i--;i>=0;i--)
          free(a[i]);
                     // elibereaza liniile alocate anterior
        free(a);
                          // elibereaza vectorul de pointeri
         printf("memorie insuficienta\n");
         exit(EXIT FAILURE);
```

Eliberarea memoriei unei matrici alocate dinamic

Având în vedere că am alocat memorie dinamică, e obligatoriu ca la finalul programului să o eliberăm.

Înainte de a elibera cu free(a) e nevoie să eliberăm memoria ocupată de vectorii care reprezintă liniile matricii (free(a[i])):

```
for(i=0;i<m;i++)
    free(a[i]);  // elibereaza fiecare linie
free(a);  // elibereaza vectorul de linii</pre>
```

Alocarea dinamică a matricilor

Alocarea și eliberarea memoriei sunt mai complexe, pentru că fiecare linie se alocă separat.

În schimb, accesarea elementelor matricii se face exact ca în cazul în care ea ar fi fost declarată static (int a[M][N]), ceea ce simplifică toate operațiile cu matricea.



Ce facem la TP
Date în memoria calculatorului
Matrici
Matrici alocate dinamic
Exerciții

Exerciții

1. Se citesc de la tastatură *m* și *n*, fiecare mai mici decat 10. Să se creeze o matrice în care la fiecare poziție să fie maximul indecșilor acelei poziții și să se afișeze matricea.

Exerciții

2. Se citește un număr n<=10 de orașe și apoi pentru fiecare 2 orașe se citește distanța directă dintre ele. Dacă distanța este 0, înseamnă că între acele orașe nu există drum direct.

Să se afișeze perechea de orașe cele mai apropiate între ele în mod direct.

Exerciții

3. Se citește de la tastatură o matrice cu n<=20 linii și coloane.

Să se memoreze într-un vector suma tuturor elementelor de pe fiecare linie și într-un alt vector, suma tuturor elementelor de pe fiecare coloană. Să se afișeze cei doi vectori.

Se vor utiliza pointeri atât pentru vectori cât și pentru matrici.



Vă mulțumesc!