

# INSTRUCTIONS TO CANDIDATES

- All exam rules stated by the Tshwane University of Technology apply.
- Ensure a single final version of your source code is handed in as requested.
- If needed, state all necessary assumptions clearly in code commentary.

**MARKS:** 100%

PAGES: 15 (incl. cover)

**EXAMINER**:

Mr A.J. Smith

Prof J.A. Jordaan

**MODERATOR:** 

Mr D Engelbrecht

TIME:

90 Minutes

(30 minutes extra time)

### FACULTY OF ENGINEERING AND THE BUILT ENVIRONMENT

# DEPARTMENT OF ELECTRICAL ENGINEERING

ES216BB ENGINEERING SOFTWARE DESIGN B

**EVALUATION 1** 

**SEPTEMBER 2024** 

#### **EVALUATION INSTRUCTIONS**

- 1. Plagiarism: Submit only original work. We will use similarity software to verify the authenticity of all submissions.
- 2. Permitted Tools: You are allowed to use only CodeBlocks, and Google Chrome to access the evaluation, view the evaluation PDF and upload submission for this evaluation. Access to emails, other online resources, and memory sticks is strictly prohibited. Please be aware that computer activity will be remotely monitored. Breaches of TUT's official examination and module rules will result in a minimum penalty of zero for this evaluation, with the potential for further disciplinary action.
- 3. File Submission: Your source code file must be named according to this format: "<student number>.h" (e.g. 21011022.h). Do not add any other text (name, surname, etc.) to the file name (ONLY YOUR STUDENT NUMBER).
- 4. **Uploading Instructions:** Submit your ".cpp" file via the designated upload link. While multiple uploads are allowed, only the most recent submission will be retained on the system. If you make an error in your initial upload, simply re-upload your file, and the previous version will be overridden. Download your submission to confirm that the correct file has been uploaded.
- 5. Evaluation Scope: This assessment encompasses basic content from ES216AB and specifically ES216BB content defined in Unit1 to Unit3
- 6. Programming Language: Construct your program in C++ and adhere to structured programming principles.
- 7. Editing and Requirements: Your program must meet all specified requirements. Refer to the attached appendices for additional details.

### 8. Evaluation Requirements:

- a. Remember to save your work on the PC "D: Drive" and save regularly throughout the evaluation.
- b. Do not modify the given code in the ".cpp" file except for implementing the requested functions as required.
- c. Use the exact function names and parameters as used in the in the ".cpp" file function prototypes and main function.
- d. Complete the C++ functions below the main function in each comment block as shown.

#### C++ FILE CODE EXPLANATION

The provided C++ code is designed to manage a dynamically created array of structures, where each structure contains a string and a float value. The code starts by including necessary standard header files such as "iostream" and "fstream". The "using namespace std" directive is used to simplify the code by avoiding the need to prefix standard library entities with "std::".

A "struct" named "Record" is defined to represent each element of the dynamically allocated array. Each 'Record' contains a string ("sData") and a float ("fData"). The functions that manage this data will be implemented below the main function.

The "main" function initialises a pointer to the dynamically allocated array, "records", and sets it to "nullptr" or "NULL" (depending on the compiler settings), indicating an empty array. Variables such as "arrSize", "choice", "FileName", and "Average" are declared to store the array size, the user's menu choice, the name of the file to be read, and the average of the float values in the array, respectively.

The program then enters a do-while loop, which displays a menu to the user with options to read and populate the array from a file, calculate the average of the float values, display the string data, display the float data along with its deviation from the average, and delete the array and exit the program. The user is prompted to enter a choice, and based on the input, a switch-case construct is used to call the appropriate function.

For instance, if the user selects the first option, they are prompted to enter a filename, and the "ReadTextFile" function is called to populate the array. If the second option is chosen, the "AverageArray" function is called to compute the average of the float values, and the result is displayed. The third and fourth options call the two versions (overloaded) of the "DisplayRecordData" function to show the data in the array. The fifth option deletes the array using the "DeleteArray" function and exits the program. If the user enters an invalid choice, a message is displayed.

After each operation, the user is prompted to press any key to continue, and the screen is cleared using the "system("cls")" command. The loop continues until the user selects the fifth option to exit.

#### **HEADER FILE FUNCTIONS**

The required functions for the C++ program should be implemented in the appropriate comment blocks as given in the .cpp file. The function declarations and descriptions are as follows:

#### 1. Initialise Record Member Function

### void Record::InitialiseRecord(string sPara, float fPara);

The InitialiseRecord function is a member of the Record structure. It is used to initialise the sData and fData members of the Record structure with the provided parameters. The function receives two parameters: a string sPara and a float fPara, which are used to set the sData and fData members of the Record, respectively. This function populates a record structure element with data.

#### 2. Text File Line Count Function

### int TextFileLineCount(string FileName);

The **TextFileLineCount** function is designed to count the number of lines in the text file, which represent one data point entry. It has a file name parameter, string FileName, that contains the text file name and the file type extension. The function opens the text file and then traverses through the file by counting each line using the getline() function, which receives the file object and string variable as parameters. Thereafter, the file must be closed appropriately before the line count is returned to the function call.

### 3. Read File and Populate Function

### void ReadFileAndPopulate(string FileName, Record \*\*records, int \*arrSize);

The ReadFileAndPopulate function reads data from a text file and populates a dynamically allocated array of Record structures. It receives three parameters: the file name, string FileName, a double pointer, Record \*\*records, to the records array, and a pointer, int arrSize, that stores the size of the array.

- 1. The function first checks if any memory is allocated for the records array and deallocates it, if necessary, before continuing.
- 2. Counts the number of lines in the file using the **TextFileLineCount** function.
- 3. Allocates memory for the records array based on the count.
- 4. If memory allocation fails, an error message is displayed.
- 5. Reads the data from the file to populate each Record structure in the array using the structure member function InitialiseRecord.

### 4. Average Array Function

### float AverageArray(Record \*records, int arrSize);

The AverageArray function calculates the average of the fData values in an array of Record structures. It receives two parameters: a pointer to the records array and an integer representing the size of the array. The function sums up all the fData values in the array and divides the total by the number of elements in the array to calculate the average. If the array size is zero, the function returns 0.0 to avoid division by zero.

### 5. Display Record Data Function 1 - Overloaded

### void DisplayRecordData(Record \*records, int arrSize);

The **DisplayRecordData** function is used to display the contents of an array of Record structures in a formatted manner. It receives two parameters: a pointer to the records array and an integer representing the size of the array. The function outputs the sData and fData of each Record in the array in a table format, with appropriate column headings and separators. This function provides a basic display of the data without any additional calculations. The iomanip library function setw(), is used to effectively style the table output.

### 6. Display Record Data Function 2 - Overloaded

### void DisplayRecordData(Record \*records, int arrSize, float avg);

This version of the **DisplayRecordData** function is similar to function 4 above, but includes an additional parameter avg, representing the average of the fData values in the array. The function displays each Record's sData, fData, and the deviation of fData from the average. The deviation is calculated as "fData - avg" and is displayed in a third column. The output is formatted in a table with appropriate headings and separators. The iomanip library function setw(), is used to effectively style the table output.

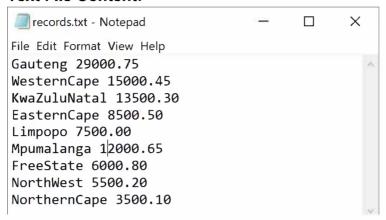
### 7. Delete Array Function

### void DeleteArray(Record \*\*records, int \*arrSize);

The **DeleteArray** function is responsible for deallocating the memory allocated for the records array and resetting its size to zero. It receives two parameters: a double pointer to the records array and a pointer to the size of the dynamic array. The function deletes the dynamically allocated array of Record structures, sets the records pointer to nullptr or NULL to avoid dangling pointers, and resets array size to zero, indicating that the array no longer holds any data.

### **PRINT SCREENS**

#### **Text File Content:**



### Main Menu:

```
Evaluation 1

    Read and Populate Array

2. Calculate Array Average
3. Display Data
4. Display Data And Deviation
5. Delete Array and Exit
Choice:
```

#### **Load Non-Existent File:**

```
Evaluation 1

    Read and Populate Array

2. Calculate Array Average
3. Display Data
4. Display Data And Deviation
5. Delete Array and Exit
Choice: 1
File Name: rec.txt
Error: Unable to open file rec.txt
Press any key to continue...
```

### **Load Existent File:**

## \_\_\_\_\_\_ Evaluation 1 1. Read and Populate Array 2. Calculate Array Average 3. Display Data 4. Display Data And Deviation 5. Delete Array and Exit Choice: 1 File Name: records.txt Press any key to continue...

### **Display Average of Float Data:**

```
Evaluation 1
1. Read and Populate Array
2. Calculate Array Average
3. Display Data
4. Display Data And Deviation
5. Delete Array and Exit
Choice: 2
Average: 11167.1
Press any key to continue...
```

### Display records data:

```
Evaluation 1
1. Read and Populate Array
2. Calculate Array Average
3. Display Data
4. Display Data And Deviation
5. Delete Array and Exit
Choice: 3
String Data Float Data
Gauteng 29000.8
WesternCape 15000.5
KwaZuluNatal 13500.3
EasternCape 8500.5
Limpopo 7500
Mpumalanga 12000.7
FreeState 6000.8
                  7500
12000.7
6000.8
NorthWest
                       5500.2
NorthernCape 3500.1
Press any key to continue...
```

### **Display Records Data with Deviation:**

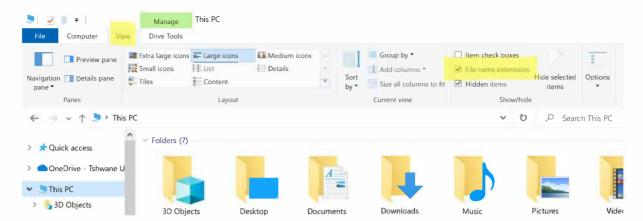
Evaluation	1		5
1. Read and Popula 2. Calculate Array 3. Display Data 4. Display Data An 5. Delete Array an Choice: 4	Average d Deviation		
String Data	Float Data	Deviation from Avg	
Gauteng WesternCape KwaZuluNatal EasternCape Limpopo Mpumalanga FreeState NorthWest NorthernCape	13500.3 8500.5 7500 12000.7 6000.8 5500.2 3500.1	17833.7 3833.37 2333.22 -2666.58 -3667.08 833.567 -5166.28 -5666.88 -7666.98	
rress any key es e	oneinae		,

### Exit:

```
Evaluation 1
1. Read and Populate Array
2. Calculate Array Average
3. Display Data
4. Display Data And Deviation
5. Delete Array and Exit
Choice: 5
Exit Program...
Process returned 0 (0x0) execution time : 126.855 s
Press any key to continue.
```

#### HOW TO RUN THE SHOWCASE FILE

1. Enable file extensions (see highlighted in yellow)



- 2. Change the name from "Showcase.old" to "Showcase.exe"
- 3. Run the "ShowcaseEV.exe" by double-clicking on the icon.
- 4. The following may be shown by Windows. Click on "More info"



5. Click on "Run anyway"



# ANNEXURE A – MARK ALLOCATION

Note: Score range is 0 - 4 which is: 0-none, 1-poor, 2-average, 3-good, 4-excellent

TEST RUBRIC	SCORE [0-4]	WEIGHT [%]
C++ CODE EVALUATION		55
Initialise Record Member Function     (Initialize structure members with default parameters)		5
2. Count File Lines Function (Insert new record into dynamic array)		7
3. Read and Populate Function 3.1 Deallocate Memory Before		2
3. Read and Populate Function 3.2 Retrieve Line Count		2
3. Read and Populate Function 3.3 Allocate Dynamic Memory and set Size		3
3. Read and Populate Function 3.4 Check Memory Allocation Failure		2
3. Read and Populate Function 3.5 Read Text File and Populate Using Initialise		3
4. Average Array Function (Calculate average of float data in array)		6
5. Display Record Data Function 1 - Overloaded (Display array data: string and float)		6
6. Display Record Data Function 2 - Overloaded (Display array data: string, float, and deviation)		6
7. Delete Array Function (Deallocate dynamic array memory)		5
9. Overall Impression (Neatness, Readability, Spacing, and Indentation)		4
8. No Compile or Runtime errors		4
TOTAL		50+5

<b>Graduate Attribute</b>	GA Number	GA Score [0-5]
Application of scientific and engineering knowledge	GA2	1, 2, 3
Engineering methods, skills, tools, including information technology	GA5	5,6,7
Impact of Engineering Activity	GA7	9
Engineering Professionalism	GA10	8

### ANNEXURE B – INFORMATION SHEET

Data types: void, char, short, int, float, double

Data Type modifiers: const, auto, static, unsigned, signed

Arithmetic operators: \* / % + -

Relational operators: < <= >= == !=

**Assignment operator:** = += -= \*= /= %= &= ^= |= <<= >>=

**Logic operators:** && || !

Bitwise logic operators: & | ^ ~ << >>

Pointer operators: Derefernce: \* Address: &

**Control Structures:** 

**IF** Selection: if (condition) { ... };

**IF ELSE** Selection: if (condition) { ... } else { ... };

**WHILE** Loop: while (condition) { ... };

**DO WHILE** loop: do { ... } while (condition);

**FOR** Loop: for (initial value of control variable; loop condition; increment

of control variable) { ... }

**SWITCH** Selection: switch (control variable) { case 'value': ...; break; default: ...;

break; }

Functions: return data\_type function\_name (parameters) { ... };

Common Library Functions: printf(), scanf(), rand(), srand(), time(), isalpha(),

isdigit() , getchar() , getch(), strcpy()

**Arrays:** 

One dimensional: data type variable name[size];

Two dimensional: data type variable name [x size][y size];

## ANNEXURE C – ASCII TABLE

2 2 002 STX (start of text) 3 4 22 042 "" 4 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 line) 11 B 013 VT (vertical tab) 12 C 014 FF (NP form feed, new page) 14 E 016 SO (shift out)  34 22 042 "" 66 42 102 B B 98 62 142 b b 67 43 103 C C 99 63 143 c c 68 44 104 D D 100 64 144 d d 68 44 104 D D 100 64 144 d d 68 44 104 D D 100 64 144 d d 69 45 105 E E 101 65 145 e e 70 46 106 F F 102 66 146 f f 71 47 107 G G 103 67 147 g g 72 48 110 H H 104 68 150 h h 73 49 111 I I 105 69 151 i i 74 4A 112 J J 106 6A 152 j j 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 110 6E 156 n m	Dec	<u>H</u>	(Oct	Chai	,	Dec	Нх	Oct	Html	Chr	Dec	Нх	Oct	Html	Chr	Dec	: Нх	Oct	Html C	hr_
2 2 002 STX (start of text)	0	0	000	NUL	(null)	32	20	040		Space	64	40	100	@	0	96	60	140	`	-
3 3 003 ETX (end of text) 4 4 004 EOT (end of transmission) 5 5 005 ENQ (enquiry) 6 6 6 006 ACK (acknowledge) 7 7 007 BEL (bell) 9 9 011 TAB (horizontal tab) 10 A 012 LF (NL line feed, new page) 11 B 013 VT (vertical tab) 12 C 014 FF (NP form feed, new page) 13 D 015 CR (carriage return) 14 E 016 S0 (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 10 10 12 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 D 03 DC3 (device control 3) 18 12 023 DC3 (device control 3) 18 12 033 DC3 (device control 4) 19 13 D 055 SWK (negative acknowledge) 19 13 D 055 SWK (negative acknowledge) 19 15 S 7 30 C3 SWB (substitute) 10 C 03 FS (group separator) 10 C 03 FS (group separator) 10 C 04 C 04 C 05 SW (unit separator) 10 C 05 SWS (unit separato	1	1	001	SOH	(start of heading)	33	21	041	<b>!</b> ;	Ĩ.	65	41	101	<b>&amp;</b> #65;	A	97	61	141	a	a
4 4 004 EOT (end of transmission) 5 5 005 ENQ (enquiry) 37 25 045 \$\pi37; \pi 69 45 105 \$\pi68; D 100 64 144 \$\pi#10; B 6 6 006 \$\pi KK (aknowledge) 38 26 046 \$\pi#38; \pi 69 45 105 \$\pi	2	2	002	STX	(start of text)	34	22	042	"	rr .	66	42	102	B	В	98	62	142	b	b
5 5 005 ENQ (enquiry) 6 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) 19 13 023 DC3 (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) 26 1A 032 SUB (substitute) 27 1B 033 ESC (escape) 28 1C 034 FS (file separator) 28 1C 036 FS (record separator) 29 1D 035 GS (group separator) 20 14 024 DC3 (group separator) 20 16 036 RS (record separator) 20 17 037 US (unit separator) 20 18 038 FS (record separator) 20 19 036 RS (record separator) 20 19 036 RS (record separator) 20 19 036 RS (record separator) 20 19 037 US (unit separator) 20 19 038 RS (record separator) 20 10 036 RS (record separator) 20 10 036 RS (record separator) 20 10 037 US (unit separator) 20 10	3					35	23	043	<b>#</b> ;	#	67	43	103	C	C					
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 S0 (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 1 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 3) 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 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) 48 2 0 046 A#39; '7 47 107 A#71; G 103 67 147 4#103; E 7 107 107 A#71; G 103 67 147 4#104; E 7 107 4#11 107 107 107 6#11 108 15 107 17 4 110	4					36	24	044	<b>\$</b> ;	Ş	68	44	104	D	D	100	64	144	d	d
7 7 007 BEL (bell)	5					37	25	045	%	*										
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 1A 032 SUB (substitute) 27 1B 033 ESC (escape) 28 10 1F 037 US (unit separator) 30 1E 036 RS (record separator) 31 1F 037 US (unit separator) 40 28 050 6#40; (72 48 110 6#72; H 106 68 150 6#104; H 129 051 6#41; ) 73 49 111 6#73; I 105 69 151 6#105; I 106 6A 152 6#104; H 107 6B 153 6#107; I 106 6A 152 6#108; I 107 6B 153 6#107; I 106 6A 152 6#108; I 107 6B 153 6#107; I 106 6A 152 6#108; I 107 6B 153 6#107; I 108 6C 154 6#108; I 10	6	6	006	ACK	(acknowledge)															
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) 19 13 023 DC3 (device control 4) 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) 29 1D 035 GS (group separator) 30 1E 036 RS (record separator) 30 1F 037 US (unit separator) 41 29 051 ) ) 42 2A 052 * * 74 4A 112 J J 100 6A 152 j J 75 4B 113 M K 100 6B 153 k I 76 4D 111 I I 105 69 151 i I 106 6A 152 j J 77 4D 115 M M 109 6D 155 k M 77 4D 115 M M 109 6D 155 k M 78 4E 116 N N 110 6E 156 k M 79 4F 117 O O 111 6F 157 o D 110 6E 156 n M 79 4F 117 O O 111 6F 157 o M 110 6E 156 n M 110 6E 156 o M 110 6E 156 n M 111 6F 157 o M 110 6E 156 o M 110 6E 156 o M 111 6F 157 o M 110 6E 156 o M 110 6E 160 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	7				(bell)	39										1				
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 SVN (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) 29 1D 035 GS (group separator) 20 1E 036 RS (record separator) 30 1E 036 RS (record separator) 31 1F 037 US (unit separator) 42 2A 052 * * 44 A 112 J J 106 6A 152 j J 47 4A 112 J J 106 6A 152 j J 47 4A 112 J J 106 6A 152 k k 47 4A 112 J J 106 6A 152 k k 48 113 K K 107 6B 153 k k 49 2D 055 , - 76 4C 114 L L 108 6C 154 l p 77 4D 115 M M 109 6D 155 m p 78 4E 116 M M 109 6D 155 m p 78 4E 116 N N 109 6D 155 m p 78 4E 115 M M 109 6D 155 m p 78 4E 116 L L 108 6C 154 l p 78 4E 116 L L 108 6C 154 l p 78 4E 116 L L 108 6C 154 l p 78 4E 116 L L 108 6C 154 l p 78 4E 116 M M 109 6D 155 m p 78 4E 115 M M 109 6D 155 m p 78 4F 117 O O 111 ˆ n p 80 50 120 X P 112 70 160 o p 81 51 121 Q O 120 X P 112 70 160 o p 81 51 121 Q O 120 X P 112 70 160 o p 82 52 122 R R 114 72 162 r p 83 53 123 S S 115 73 163 s S 84 55 125 X U 117 75 165 o O 85 55 125 X U 117 75 165 o O 86 154 l D 87 50 120 X P 112 70 160 p D 80 50 120 X P 112 70 160 p D 81 51 121 Q O 120 X P 112 70 160 p D 81 51 121 Q O 120 X P 112 70 160 p D 81 51 121 Q O 120 X P 112 70 160 p D 82 52 122 R R 114 72 162 r D 83 53 123 S S 115 73 163 s S 84 55 125 X U 117 75 165 p D 84 52 120 X P	8	8	010	BS	(backspace)					•						1				
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) 31 1F 037 US (unit separator) 43 2B 053 + + 44 2C 054 , , 76 4C 114 K K 107 6B 153 k R 108 6C 154 l I 109 6D 155 m I 110 6F 157 o I 109 6D 155 m I 110 6F 157 n I 110 6F 157 o I 110 6F	9	_			· · · · · · · · · · · · · · · · · · ·					-										
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) 31 1F 037 US (unit separator) 44 2C 054 , , 76 4C 114 L L 108 6C 154 l L 77 4D 115 6#77; M 109 6D 155 m M 109 6D 155 &#	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) 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 1F 037 US (unit separator) 45 2D 055 - - 46 2E 056 . . 47 2F 057 / / 48 2E 056 . . 47 2F 057 / / 48 2E 056 . . 48 2E 056 . . 48 2E 056 . . 49 2F 057 / / 48 2E 056 . . 49 2F 057 / / 48 2E 056 . . 49 2F 057 / / 48 2E 056 . . 49 2F 057 / / 48 2E 056 . . 49 2F 057 / / 48 2E 056 . . 49 2F 057 / / 48 2F 057 / / 49 2F 057 / / 48 2F 057 / / 49 2F 057 / / 48 2F 056 . . 49 2F 056 . . 49 2F 057 / / 49 2F 057 0 D 110 6E 156 e D 111 6F 157  D 110 6E 156 e D 111 6F 157	11				(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 4) 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) 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 1F 037 US (unit separator) 46 2E 056 . .  78 4E 116 N N 110 6E 156 n n 79 4F 117 O 0 111 6F 157 o 0 111 70 160 p p 111 71 161 q 0 111 71 161 q 0 111 72 162 r p 113 71 161 q 0 114 72 162 r p 115 73 163 s s 117 75 165 u p 117 75 165 u p 118 72 164 T p 119 77 167 w p 110 05	12	_																		
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) 30 1E 036 RS (record separator) 31 1F 037 US (unit separator) 48 30 060 0 0 48 30 060 0 0 48 30 060 0 0 48 30 060 0 0 48 30 060 0 0 48 30 060 0 0 48 30 060 1 1 48 4 124 0 1 4 16 4 164 q 0 41 11 77 160 w 0 47 5 165 w 0 48 50 120  0 48 50 120	13																			
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) 31 1F 037 US (unit separator) 48 30 060 0 0 49 31 061 1 1 50 32 062 2 2 50 32 062 2 2 50 32 062 2 2 50 32 062 2 2 50 32 062 2 2 50 32 062 2 2 50 32 062 2 2 50 32 062 2 2 50 32 062 2 2 50 32 062 2 2 50 32 063 3 3 50 32 063 3 3 50 35 063 3 3 50 35 063 3 3 50 35 065 5 5 50 55 125 R R 114 72 162 r R 116 74 164 t R 117 75 165 u R 118 76 166 v R 119 77 167 w R 119 77 167 w R 110 77 167 w R 110 77 167 w R 110 77 167 w R 111 71 161 q R 112 70 160 p R 113 71 161 q R 113 71 161 q R 114 72 162 r R 115 73 163 s R 115 73 163 s R 116 74 164 t R 117 75 165 s R 118 75 165 u R 119 75 165 s R 110 75 165 r R 110 71 161 q R 111 71 161 q R 111 71 161 q R 112 70 160 p R 113 71 161 q R 114 72 162 r R 115 73 163 s R 116 74 164 t R 117 75 165 s R 117 75 165 u R 117 75 165 s	14	_														1				
17 11 021 DC1 (device control 1)	15																			
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) 29 1D 035 GS (group separator) 30 1E 036 RS (record separator) 31 1F 037 US (unit separator) 32 062 2 2 38 070 & 2 2 38 070 & 2 2 38 070 & 2 3 38 070 & 2 3 38 070 & 2 4 38 070 & < 4 38 070 & < 4 39 071 & 9 9 39 071 & 9 9 39 071 & 9 9 39 071 & 9 9 39 071 & 9 9 39 071 & 9 9 39 071 & 9 9 39 071 & < 4 30 074 & < 4 30 075 & = 4 30 075 & = 4 30 075 & = 4 30 077 & ? 2 31 077 & ? 2 31 077 & ? 2 31 077 & ? 2 32 062 & 2 2 32 062 & 2 2 33 162 & R R 3114 72 162 & r r 31 14 72 162 & r r 31 16 34 115; s 31 18 73 163 & s s 31 17 027 ETB (avide control 4) 31 18 70 164 & t t 32 14 04 & T T 34 064 & 4 4 38 53 123 & S S 31 115 73 163 & s s 38 53 123 & S S 31 15 03 08 & = s 38 53 123 & S S 31 115 03 08 & = s 31 17 027 ETB (avide control 4) 32 12 02 4 6 & r r 33 063 & 3 3 34 064 & 4 4 34 064 & 4 4 38 53 123 & S S 31 115 73 163 & s s 38 53 123 & S S 31 115 73 163 & s s 38 53 123 & S S 31 115 73 163 & s s 38 53 123 & S S 31 115 73 163 & s s 38 53 123 & S S 31 115 73 163 & s s 38 53 123 & S S 31 115 73 163 & s s 38 53 123 & S S 31 15 73 163 & s s 38 53 123 & S S 31 15 73 163 & s s 38 53 123 & S S 31 115 73 163 & s s 38 53 123 & S S 31 115 73 163 & s s 38 53 123 & S S 31 115 73 163 & s s 38 53 123 & S S 31 16 74 04 04 & ¤ &  t 34 54 124 & T T 31 16 74 164 & t t 38 55 125																1				_
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) 31 1F 037 US (unit separator) 31 1F 037 US (unit separator) 32 34 064 4 4 34 064 4 4 34 064 4 4 34 064 4 4 38 53 123 S S 115 73 163 s S 38 65 126 V V 116 74 164 t U 38 55 125 U U 39 57 127 W W 39 77 127 W W 39 59 131 Y Y 39 50 135 Z Z 39 50 135 ] ] 30 16 36 RS (record separator) 31 1F 037 US (unit separator) 32 17 027 ETB (end of trans. block) 33 066 6 6 34 066 6 6 36 066 6 6 37 067 7 7 38 070 7 7 39 071 7 9 38 070 7 9 39 071 9 9 39 59 131 Y Y 39 59 131 Y					· · · · · · · · · · · · · · · · · · ·												. –			
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) 29 1D 035 GS (group separator) 30 1E 036 RS (record separator) 31 1F 037 US (unit separator) 32 4 064 4 4 34 064 4 4 35 065 5 5 35 065 5 5 36 066 6 6 37 067 7 7 38 076 7 7 38 070 8 8 38 070 8 8 38 070 8 8 39 071 9 9 39 071 9 9 39 071 9 9 30 071 9 9 31 075 < < 32 074 < < 33 075 = = 34 064 4 4 34 064 4 4 35 065 026 V U 36 066 6 6 37 067 7 7 38 076 7 7 38 076 7 7 39 071 9 9 30 071 9 9 30 072 : 10 034 Z 2 31 075 = = 32 076 < < 33 075 = = 34 064 4 4 35 076 ? 2 36 066 6 6 36 066 6 6 36 066 6 6 37 077 B 8 38 070 7 7 38 070 7 7 39 071 9 9 39 071 ʑ 9 30 071 ʑ 9 30 072 ʔ < 31 075 = = 32 076 > > 33 076 > > 34 064 4 4 35 076 A U 36 066 6 6 37 076 A U 38 076 V U 38 07																				
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) 29 1D 035 GS (group separator) 30 1E 036 RS (record separator) 31 1F 037 US (unit separator) 31 1F 037 US (unit separator) 35 35 065 5 5 36 066 6 6 36 066 6 6 36 066 6 6 36 066 6 6 36 066 6 6 36 066 6 6 36 066 6 6 36 067 7 7 37 067 7 7 38 070 7 7 38 070 7 9 38 070 7 9 38 070 7 9 38 070 7 9 39 071 9 9 39 071 ʑ 9 39 072 : 190 054 054 064 064 064 064 064 064 064 064 064 06																				
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) 29 1D 035 GS (group separator) 30 1E 036 RS (record separator) 31 1F 037 US (unit separator) 32 17 027 ETB (end of trans. block) 34 066 6 454; 6 35 36 066 6 454; 6 36 56 126 6 486; V 37 118 76 166 6 4118; V 38 57 127 6 487; W 39 57 127 6 487; W 31 19 77 167 6 4119; W 38 58 130 6 488; X 31 070 6 455; 9 38 58 130 6 488; X 31 070 6 455; 9 38 58 130 6 488; X 31 070 6 455; 9 38 58 130 6 488; X 31 070 6 455; 9 39 59 131 6 489; Y 31 18 76 166 6 4118; V 31 19 77 167 6 4112; V 32 120 78 170 6 4120; V 38 58 130 6 488; X 31 18 70 167 6 455; 9 39 59 131 6 489; Y 31 18 76 166 6 4118; V 31 19 77 167 6 4112; V 32 120 78 170 6 4120; V 33 17 027 6 455; 9 39 59 131 6 489; Y 30 12 78 170 6 4121; V 30 12 7																				
23 17 027 ETB (end of trans. block) 55 37 067 & #55; 7 24 18 030 CAN (cancel) 56 38 070 & #56; 8 8 58 130 & #88; X 120 78 170 & #120; X 120 78 170 & #121; Y 121 79 171 & #121; Y 122 74 18 033 ESC (escape) 59 3B 073 & #59; 28 1C 034 FS (file separator) 60 3C 074 & #60; 4 92 5C 134 & #92; 1 125 7D 175 & #125; 3 1 1F 037 US (unit separator) 63 3F 077 & #63; 2 95 5F 137 & #95; 1 127 7F 177 & #127; D																				
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) 31 1F 037 US (unit separator) 35 38 070 7 8 38 070 8 8 38 070 8 8 38 070 8 8 38 070 8 8 38 070 8 8 38 58 130 X X 120 78 170 x X 121 79 171 y Y 122 7A 172 z Z 123 7B 173 { { 92 5C 134 \ \ 124 7C 174   \ 93 5D 135 ] \ 125 7D 175 } \ 126 7E 176 ~ ~ 95 5F 137 _ \ 127 7F 177  D																				
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 : 1 90 5A 132 Z Z 122 7A 172 z Z 27 1B 033 ESC (escape) 59 3B 073 ; 191 5B 133 [ [ 123 7B 173 { { 28 1C 034 FS (file separator) 60 3C 074 < <																				
26 1A 032 SUB (substitute)  58 3A 072 ::  90 5A 132 Z Z  122 7A 172 z Z  91 5B 133 [ [  123 7B 173 { {  124 7C 174   I  125 7D 175 } }  126 7E 176 ~ ~  127 7F 177  D																				
27 1B 033 ESC (escape)  59 3B 073 ; 2  60 3C 074 < 4  92 5C 134 \ \ 124 7C 174   \ 93 5D 135 ] ]  125 7D 175 } \ 93 1 1F 037 US (unit separator)  63 3F 077 ? 2  91 5B 133 [ [ 123 7B 173 { { 92 5C 134 \ \ 124 7C 174   \ 93 5D 135 ] ] 125 7D 175 } \ 94 5E 136 ^ \ 126 7E 176 ~ \ 95 5F 137 _ _ 127 7F 177  D																				_
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 ^ \ 20 126 7E 176 ~ \ 20 3F 077 ? \ 20 95 5F 137 _ \ 20 127 7F 177  \ 20 127 7F 177  \ 20 128 129 129 129 129 129 129 129 129 129 129					·															
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 ~ ~ 95 5F 137 _ _ 127 7F 177  D																				
30 1E 036 RS (record separator) 62 3E 076 > > 94 5E 136 ^ ^ 126 7E 176 ~ ~ 31 1F 037 US (unit separator) 63 3F 077 ? 95 5F 137 _ _ 127 7F 177  D					-															
31 1F 037 US (unit separator) 63 3F 077 ? 2 95 5F 137 _ _   127 7F 177  D															-					
																	. —			
Source: www.LookupTables.c	31	1F	037	US	(unit separator)	63	3F	077	<b>&amp;</b> #63;	?	95	5F	137	&#<b>95</b>;</td><td>_</td><td>127</td><td>7F</td><td>177</td><td></td><td>DE.</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>S</td><td>DUFC</td><td>e: 4</td><td>VIVIY.</td><td>.Look</td><td>upTable</td><td>5 .00</td></tr></tbody></table>						

Page 15 of 15