

## Assignment 1

### Due Date

The assignment is due at 3PM Wednesday March 29<sup>th</sup> 2017 and should be completed with a partner. You and your partner should work together on all of the design and programming. It should be done using the *pair-programming* methodology and not division of labour.

### Background

Group→	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
↓Period																		
1	1 H																	2 He
2	3 Li	4 Be											5 B	6 C	7 N	8 O	9 F	10 Ne
3	11 Na	12 Mg											13 Al	14 Si	15 P	16 S	17 Cl	18 Ar
4	19 K	20 Ca	21 Sc	22 Ti	23 V	24 Cr	25 Mn	26 Fe	27 Co	28 Ni	29 Cu	30 Zn	31 Ga	32 Ge	33 As	34 Se	35 Br	36 Kr
5	37 Rb	38 Sr	39 Y	40 Zr	41 Nb	42 Mo	43 Tc	44 Ru	45 Rh	46 Pd	47 Ag	48 Cd	49 In	50 Sn	51 Sb	52 Te	53 I	54 Xe
6	55 Cs	56 Ba	*	72 Hf	73 Ta	74 W	75 Re	76 Os	77 Ir	78 Pt	79 Au	80 Hg	81 Tl	82 Pb	83 Bi	84 Po	85 At	86 Rn
7	87 Fr	88 Ra	**	104 Rf	105 Db	106 Sg	107 Bh	108 Hs	109 Mt	110 Ds	111 Rg	112 Cn	113 Uut	114 Fl	115 Uup	116 Lv	117 Uus	118 Uuo
			*	57 La	58 Ce	59 Pr	60 Nd	61 Pm	62 Sm	63 Eu	64 Gd	65 Tb	66 Dy	67 Ho	68 Er	69 Tm	70 Yb	71 Lu
			**	89 Ac	90 Th	91 Pa	92 U	93 Np	94 Pu	95 Am	96 Cm	97 Bk	98 Cf	99 Es	100 Fm	101 Md	102 No	103 Lr

From [www.wikipedia.org](http://www.wikipedia.org): The periodic table is a tabular arrangement of the 118 chemical elements, organized on the basis of their atomic numbers, electron configurations (electron shell model), and recurring chemical properties. Elements are presented in order of increasing atomic

number (the number of protons in the nucleus). The standard form of the table consists of a grid of elements laid out in 18 columns and 7 rows, with a double row of elements below that. The rows of the table are called periods; the columns are called groups.

## Program specification

You must write a C program to print an excerpt of the periodic table of elements.

First you should obtain the following information from the user:

- whether the ‘Lanthanum’ group (elements 57–71) and ‘Actinium’ group (elements 89–103) should be printed — the default is no;
- the atomic number of the first element to display — the default is 1 and should be used whenever a value less than 1 or greater than 118 is given; and
- the atomic number of the last element to display — the default is 118 and should be used whenever a number smaller than the first number or larger than 118 is given.

Once these values have been entered the program should display a table with the elements within the specified range displayed in their appropriate group. Any element in that range from the ‘Lanthanum’ and ‘Actinium’ groups should only be displayed if the user has requested this; there is no need to indent these, they can simply be displayed after the table.

An example of the output is as follows:

Periodic Table Printer

Print the Lanthanum/Actinium groups if necessary [Y/N]: y

Enter number of first element to print: 1

Enter number of last element to print: 0

...118 assumed...

001 H																		002 He
003 Li	004 Be												005 B	006 C	007 N	008 O	009 F	010 Ne
011 Na	012 Mg												013 Al	014 Si	015 P	016 S	017 Cl	018 Ar
019 K	020 Ca	021 Sc	022 Ti	023 V	024 Cr	025 Mn	026 Fe	027 Co	028 Ni	029 Cu	030 Zn	031 Ga	032 Ge	033 As	034 Se	035 Br	036 Kr	
037 Rb	038 Sr	039 Y	040 Zr	041 Nb	042 Mo	043 Tc	044 Ru	045 Rh	046 Pd	047 Ag	048 Cd	049 In	050 Sn	051 Sb	052 Te	053 I	054 Xe	
055 Cs	056 Ba		072 Hf	073 Ta	074 W	075 Re	076 Os	077 Ir	078 Pt	079 Au	080 Hg	081 Tl	082 Pb	083 Bi	084 Po	085 At	086 Rn	
087 Fr	088 Ra		104 Rf	105 Db	106 Sg	107 Bh	108 Hs	109 Mt	110 Ds	111 Rg	112 Cn	113 Uut	114 Fl	115 Uup	116 Lv	117 Uus	118 Uuo	
057 La	058 Ce	059 Pr	060 Nd	061 Pm	062 Sm	063 Eu	064 Gd	065 Tb	066 Dy	067 Ho	068 Er	069 Tm	070 Yb	071 Lu				
089 Ac	090 Th	091 Pa	092 U	093 Np	094 Pu	095 Am	096 Cm	097 Bk	098 Cf	099 Es	100 Fm	101 Md	102 No	103 Lr				

A second example is:

### Periodic Table Printer

Print the Lanthanum/Actinium groups if necessary [Y/N]: X

...N assumed...

Enter number of first element to print: 56

Enter number of last element to print: 76

056 Ba                      072 Hf    073 Ta    074 W      075 Re    076 Os

You will need `printf()` and `scanf()` from `<stdio.h>`, `atoi()` from `<stdlib.h>`, and `strcmp()` from `<string.h>`. You will also need to use `if` and `if-else` statements, and `for` and `while` loops.

To help you, a ‘table’ (i.e. an array of arrays) can be defined consisting of elements which are themselves arrays. Each element of the array can be a pair in which the first value (element 0) is the chemical symbol of the element and the second value (element 1) is the group to which it belongs. The index of the element in the array is one less than its atomic number. For example Helium is chemical element 2. It has symbol “He” and belongs to group “18”. It occurs at position 1 in the table. The ‘Lanthanum’ group appears with group numbers ranging from -11 to -25; the ‘Actinium’ group appears with group numbers ranging from -31 to -45.

Copy and paste the following into your program to define all chemical elements:

```
char *table[118][2]={{"H", "1"}, {"He", "18"}, {"Li", "1"}, {"Be", "2"}, {"B", "13"}, {"C", "14"}, {"N", "15"}, {"O", "16"}, {"F", "17"}, {"Ne", "18"}, {"Na", "1"}, {"Mg", "2"}, {"Al", "13"}, {"Si", "14"}, {"P", "15"}, {"S", "16"}, {"Cl", "17"}, {"Ar", "18"}, {"K", "1"}, {"Ca", "2"}, {"Sc", "3"}, {"Ti", "4"}, {"V", "5"}, {"Cr", "6"}, {"Mn", "7"}, {"Fe", "8"}, {"Co", "9"}, {"Ni", "10"}, {"Cu", "11"}, {"Zn", "12"}, {"Ga", "13"}, {"Ge", "14"}, {"As", "15"}, {"Se", "16"}, {"Br", "17"}, {"Kr", "18"}, {"Rb", "1"}, {"Sr", "2"}, {"Y", "3"}, {"Zr", "4"}, {"Nb", "5"}, {"Mo", "6"}, {"Tc", "7"}, {"Ru", "8"}, {"Rh", "9"}, {"Pd", "10"}, {"Ag", "11"}, {"Cd", "12"}, {"In", "13"}, {"Sn", "14"}, {"Sb", "15"}, {"Te", "16"}, {"I", "17"}, {"Xe", "18"}, {"Cs", "1"}, {"Ba", "2"}, {"La", "-11"}, {"Ce", "-12"}, {"Pr", "-13"}, {"Nd", "-14"}, {"Pm", "-15"}, {"Sm", "-16"}, {"Eu", "-17"}, {"Gd", "-18"}, {"Tb", "-19"}, {"Dy", "-20"}, {"Ho", "-21"}, {"Er", "-22"}, {"Tm", "-23"}, {"Yb", "-24"}, {"Lu", "-25"}, {"Hf", "4"}, {"Ta", "5"}, {"W", "6"}, {"Re", "7"}, {"Os", "8"}, {"Ir", "9"}, {"Pt", "10"}, {"Au", "11"}, {"Hg", "12"}, {"Tl", "13"}, {"Pb", "14"}, {"Bi", "15"}, {"Po", "16"}, {"At", "17"}, {"Rn", "18"}, {"Fr", "1"}, {"Ra", "2"}, {"Ac", "-31"}, {"Th", "-32"}, {"Pa", "-33"}, {"U", "-34"}, {"Np", "-35"}, {"Pu", "-36"}, {"Am", "-37"}, {"Cm", "-38"}, {"Bk", "-39"}, {"Cf", "-40"}, {"Es", "-41"}, {"Fm", "-42"}, {"Md", "-43"}, {"No", "-44"}, {"Lr", "-45"}, {"Rf", "4"}, {"Db", "5"}, {"Sg", "6"}, {"Bh", "7"}, {"Hs", "8"}, {"Mt", "9"}, {"Ds", "10"}, {"Rg", "11"}, {"Cn", "12"}, {"Uut", "13"}, {"Fl", "14"}, {"Uup", "15"}, {"Lv", "16"}, {"Uus", "17"}, {"Uuo", "18"}};
```

## Program Style

The program you write for this assignment must be in a single file called `assign_one117.c`. There should be at least three functions including the `main()` function within this file.

Your program should follow the following coding conventions:

- `const` variable identifiers should be used as much as possible, should be written all in upper case and should be declared before all other variables
- `#defined` symbols should be used for constant values when `const` is inappropriate — the length of arrays (for example)
- variable identifiers should start with a lower case letter
- every `if` and `if-else` statement should have a block of code (i.e. collections of lines surrounded by `{` and `}`) for both the `if` part and the `else` part (if used)
- every `do`, `for`, and `while` loop should have a block of code (i.e. `{}`s)
- the keyword `continue` should not be used
- the keyword `break` should only be used as part of a `switch` statement
- opening and closing braces of a block should be aligned
- all code within a block should be aligned and indented 1 tab stop (or 4 spaces) from the braces marking this block
- global variables should be used sparingly with parameter lists used to pass information in and out of functions.
- commenting:
  - There should be a block of header comment which includes at least
    - file name
    - student names
    - student identity numbers
    - a statement of the purpose of the program
    - date
    - the percentage of the work completed by the authors — 50:50 is expected, reasons should be given if it is more/less than this
  - Each variable declaration should be on a single line and should be commented
  - There should be a comment identifying groups of statements that do various parts of the task
  - Comments should describe the strategy of the code and should not simply translate the C into English

## What and how to submit

### What to submit

#### *Paper submission*

- A signed cover page (blanks can be collected from the ICT discipline office or from the [ICT discipline web site](#))
- A print-out of the source code file for the program. *Your assignment will not be fully marked unless this is present.*

#### *Electronic submission*

- The source code file which *must* be named `assig_one117.c`

### How to submit

#### *Paper submission*

- Firmly staple together all of the required documents (with the signed cover page on top) and place them in the appropriate submissions box near the ICT Help Desk.

#### *Electronic submission*

- Log in to MyLO and navigate to the Assignments tool under the Assessments icon in the top tool bar.
- Select Assignment 1 from the list of available drop-boxes.
- Click on Add a File and follow the instructions to attach your source code file (`assig_one117.c`) and then click Add. Then click Submit.
- If you want to resubmit, rename your source code as `assign_one117V2.c` and repeat the submission process.
- You may submit an entire *Visual Studio* project which contains the `assign_one117.c` source file if you wish, or just the source file.

*Please note: only one paper and electronic submission is required for the pair.*

## Marking scheme

Task/Topic	Maximum mark
<i>Program submitted in form required</i>	
Program submitted correctly	1
Program compiles without error and runs to completion with test data	1
<i>Program operates as specified</i>	
Prompts user for required information (in correct order)	2
Defaults for required information used as appropriate	2
Produces rows for valid range of elements	2
Produces columns for valid range of elements	3
Produces rows for 'Lanthanum' and 'Actinium' groups as appropriate	2
Formatting is correct (decimal places and justification)	2
Program contains at least three functions	3
<i>Program Style</i>	
Uses constants where appropriate	1
Uses symbols where necessary	1
Does not unnecessarily repeat tests or have other redundant/confusing code	2
Uses pre-existing methods (printf/scanf/atoi/strcmp)	1
<i>Coding style and internal documentation</i>	
<i>NOTE: To obtain any of the marks for this section you must submit a copy of your source code</i>	
Uses correctly the C naming conventions	2
Alignment of code and use of white space makes code readable	2
Always uses blocks in branch and loop constructs	2
Meaningful identifiers	2
Header comments for the program (author, date etc)	2
Each variable declaration is commented	2
Comments within the code indicate the purpose of sections of code (but DO NOT just duplicate what the code says)	1

## **Plagiarism and Cheating:**

Practical assignments are used in the ICT discipline for students to both reinforce and demonstrate their understanding of material which has been presented in class. They have a role both for assessment and for learning. It is a requirement that work you hand in for assessment is substantially your own.

### **Working with others**

One effective way to grasp principles and concepts is to discuss the issues with your peers and/or friends. You are encouraged to do this. We also encourage you to discuss aspects of practical assignments with others. However, once you have clarified the principles, you must express them in writing or electronically entirely by yourself in your pair. In other words you and your partner must develop the algorithm to solve the problem and write the program which implements this algorithm yourselves. You can discuss the question, but not the solution.

### **Cheating**

- Cheating occurs if you claim work as your own when it is substantially the work of someone else.
- Cheating is an offence under the [Ordinance of Student Discipline](#) within the University. Furthermore, the ICT profession has ethical standards in which cheating has no place.
- Cheating involves two or more parties.
  - If you allow written work, computer listings, or electronic version of your code to be borrowed or copied by another student you are an equal partner in the act of cheating.
  - You should be careful to ensure that your work is not left in a situation where it may be stolen by others.
- Where there is a reasonable cause to believe that a case of cheating has occurred, this will be brought to the attention of the unit lecturer. If the lecturer considers that there is evidence of cheating, then no marks will be given to any of the students involved. The case will be referred to the Head of School for consideration of further action.

Julian Dermoudy, March 6<sup>th</sup> 2017.