./

Report – Periodic Table

Course Code: <CODE>



Version Number:

Team Members :

Team No:

Module: Model Based System Engineering

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **Approved By** | **Remarks/Revision Details** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Document History**

Problem Statement

It is usually difficult for a user to find the essential details regarding the different aspects of the elements. This program will provide an easy fix for this problem

Description

Everyone may not know all the information about each element, sometimes you need someone or something that you can refer to make sure that you are handling the right information, like someone once said “Half knowledge is worse than ignorance”. So it will be super nice if we were to know someone who can provide you the answers to your questions regarding those ever evasive values or information that is vital.

This project is for all those chemistry lovers out there who might have a hard time to find some of the essential details of each element. The Periodic Table Assistant is the right tool for you, it helps you with all the details of the known periodic table elements. It also provides you with the provision of adding new elements into the periodic table upon there discovery.

Requirements

CodeBlocks IDE and C Programming Language.

Header Files used :

stdio.h - Used to perform input and output operations in C

stdlib.h - Perform standard utility

string.h - Perform string manipulation like strcmp, strlen etc

elements.h - contains all the information about the elements in the periodic table, it is programmer made header file that is added to the program

Main Features

The main feature is to easily find the details about the element that one wants to find. The elements can be searched using their

* Name
* Symbol
* Atomic Number or
* Atomic Weight

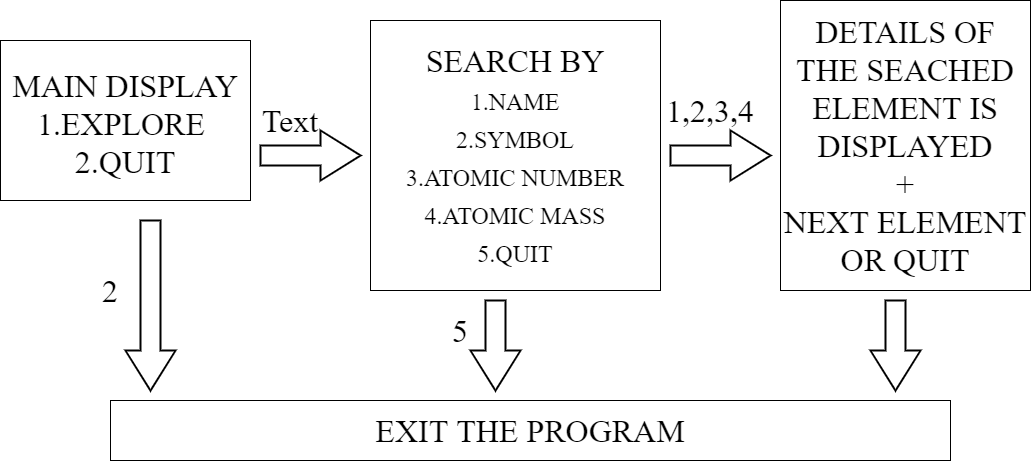
The additional that the program will provide the user are

* Crystal Structure
* Atomic Weight
* Atomic Radius
* Principal Oxidation Numbers
* Ionic Radii
* Ionisation Energies
* Electron Affinities
* Electronegativities
* Density
* Melting Point
* Boiling Point
* and probably some random facts about chemistry or elements

Design

This gives a basic idea about how the program executes. It is designed to have a main function window that provides the user with the option to explore the elements of periodic table or to quit based on the choice of the user.

When the user selects the explore option the program asks the mode of search to the user ie, whether they want to search using the name, symbol, atomic number or atomic mass.When the corresponding value is entered by the user the program prints the details regarding that element



The different functions used in these program are

* newScreen() - this function clears the screen each time it is called and tops the window with the heading “Periodic Table”.
* quit() - this function when called will ask for confirmation from the user to exit from the program
* again() - this function is called after the result of a search for an element, this asks the user whether they want to search another element
* info1() - this function is called when the user selects the option to search the element using its name, it prints the information related with the corresponding element
* info2() - this function is called when the user selects the option to search the element using its symbol, it prints the information related with the corresponding element
* info3() - this function is called when the user selects the option to search the element using its atomic number, it prints the information related with the corresponding element
* info4() - this function is called when the user selects the option to search the element using its atomic mass, it prints the information related with the corresponding element

The programmer also creates a header that contains the all the information regarding all the elements in the periodic table. When the infox() are called the details are found from this header file ‘elements.h’

Test Plan

The test plan is designed to prescribe the scope, approach, resources, and schedule of all the testing activities in the program. The plan is to identify the items to be tested, features to be tested,types of testing to be performed and other factors.

All the features present in the program should be tested. There are three types of testing that should be conducted

Integeration testing - individual software modules are tested are combined and tested as a group

System testing - conducted on a complete, integrated system to evaluate the system’s compliance with its specified requirements

API testing - test all the API’s create for the software under tested

The test objective is to verify the functionality of the program.

Suspension Criterion - If there is 40% of the test cases are failed then the testing is suspended until the programmer fixes the mistakes

Exit Criterion - Specifies a criteria that denotes a successful completion of the test phase

The test plan for this program is to check whether the right information of the element, is printed based on the search option provided by the user.The user should make sure to select the right option for the element search, it is quite easy for the user to enter the atomic number as the mass number or vice-versa.

Test Cases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl No | Test Description | Test Steps | Test Data | Test Output |
| 1 | whether user wants to explore | User selects option to explore present in the screen | Input = 2 | Output =  Goes to the next screen |
| 2 | If user decides to enter name of element | User selects option in the explore screen | Input = 1,name of element | Output = The details of the corresponding element is displayed |
| 3 | If user decides to enter the symbol of the element | User selects option in the explore screen | Input = 2,symbol of element | Output = The details of the corresponding element is displayed |
| 4 | If user decides to enter the atomic no. of the element | User selects option in the explore screen | Input = 3,atomic number | Output = The details of the corresponding element is displayed |
| 5 | If the user decides to enter the atomic mas of the element | User selects option in the explore screen | Input = 4, mass number | Output = The details of the corresponding element is displayed |
| 6 | If the user decides to quit | User selects the option corresponding to quit | Input = 2(in first screen) | Output = The user gets exited from the program |

Expected Outcome

The outcome of this program is such that the user finds the correct information based on his search without much overhead or difficulty. It should be very easy for the user, since it is clearly provided in the program whether the element search is based on either the name or symbol or stomach no. or mass. Due to this it will be easy for the user to find the other essential details regarding the searched element.

Conclusion

This program is developed for all the chemistry lovers, whom often find it difficult to find that small bit of information. This is the solution to that problem. The program has a easy interface and provides one with correct information so that they don’t need go digging in the Clark’s Tables or other reference materials. The easy and user friendly design makes it the perfect companion for those in need.