



MINI PROJECT ON C PROGRAMMING

THE PERIODIC TABLE

SUBMITTED BY:

ALOK KUMAR SAH (078BEL011)

ARUN GNAWALI (078BEL021)

AASHUTOSH PANDEY (078BEL004)

DEEPAK YADAV (078BEL036)

SUBMITTED TO:

DEPARTMENT OF
COMPUTER SCIENCE

CONTENT

- 1) Introduction
- 2) Objectives
- 3) Features
- 4) Area of Use
- 5) Problem Analysis
- 6) Algorithm
- 7) System Flow
- 8) Code
- 9) Output
- 10) Conclusion
- 11) Acknowledgement
- 12) References

INTRODUCTION

"The Periodic Table" is a computer program, written on C language using GNU GCC compiler on code block program. The array and string method is used to store the information of elements in a particular location. This location is used as a database for this project where we can perform the operations such as find and view the elements according to their atomic weight, atomic number and even their name or symbol. One of the best features of this project is it provide multiple information about the elements , like: atomic weight, atomic number, symbol, metallic property and its block. This program aims to be a part of the students facing difficulties in chemistry due to presence of more than 100 elements and their multiple properties.

Objectives

The main goals of our project are enlisted below:

- 1) To use different user defined function; to break a program into many simplified parts to deal with and to make it easier to understand the codes too.
- 2) To promote the use of array in general programming for simplified version of the program.
- 3) To apply the looping concepts in order to store and retrieve the element's data.
- 4) To use general concepts of C language to develop a simple program that as whole could be useful for students and learners in Chemistry.

Features

- 1) Compatible to search the element according to its atomic weight, atomic number, symbol or name.
- 2) Multiple information about the element searched.
- 3) Simple and easy interface.
- 4) Use of less complex codes to form an useful non-graphical interface.

AREA OF USE\SCOPE

- 1) Students facing problems in remembering the periodic table can use this simple platform to help themselves with the information of all elements in the periodic table.
- 2) Can be used to reduce time in calculating atomic weights of elements manually.
- 3) Can be useful to find and know about different elements which are not even in the syllabus (other than the first 20 or first 40 elements) of basic classes.
- 4) Can be helpful for teachers to teach periodic table in classroom with an effective and interesting way.
- 5) Can be a milestone for the programmers trying to learn the C language.

PROBLEM ANALYSIS

In order to display the different information about the elements we:

- 1) Compare the entered parameter by the user to the prewritten database in our program.
- 2) Search and match the entered parameter with the database and find the element wished for.
- 3) Display all the information about the element according to the data stored in the program.
- 4) For block and metallic property, we use logical operators with the support of real periodic table.

PROGRAM DESIGN

Algorithm:

Step 1: Start.

Step 2: Store atomic number, symbol, weight and name of all elements.

Step 3: Display "1 for atomic number,
2 for atomic symbol;
3 for atomic name;
4 for atomic weight. "

Step 3: Ask a number from the user according to the displayed menu.

Step 4: For 1 go to step 5

2 go to step 6

3 go to step 7

4 go to step 8

Other display "Invalid Input", go to step 12.

Step 5: Ask and Compare the atomic number entered to the pre-written atomic number and go to step 9

Step 6: Ask and Compare the atomic symbol entered to the pre-written atomic symbol and go to step 9

Step 7: Ask and Compare the atomic name entered to the pre-written atomic name and go to step 9

Step 8: Ask and Compare the atomic weight entered to the pre-written atomic weight and go to step 9

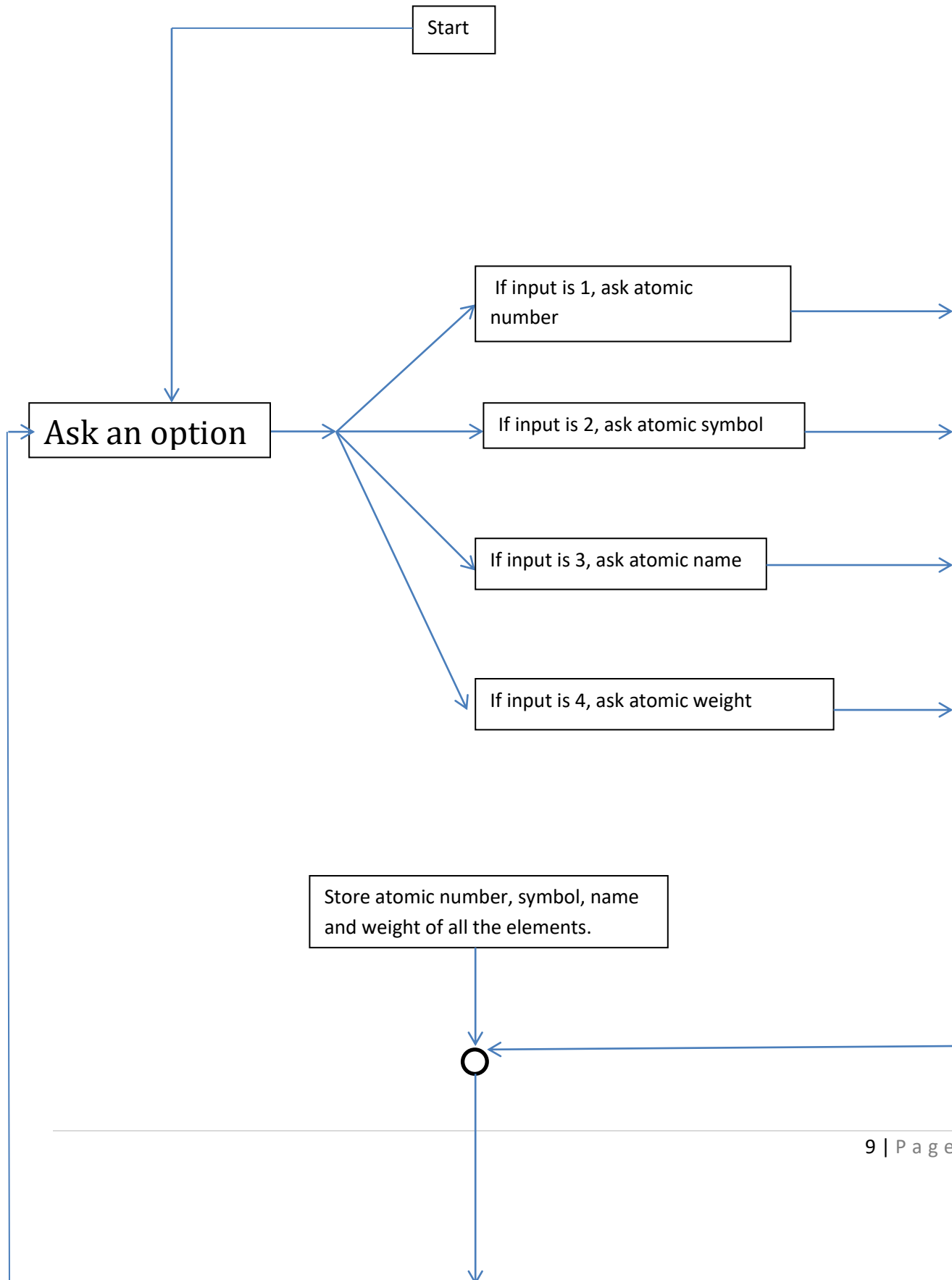
Step 9: Display all the information stored about the element in the program if the entered parameter matches with the stored parameter about the element OR display "Error!!!"

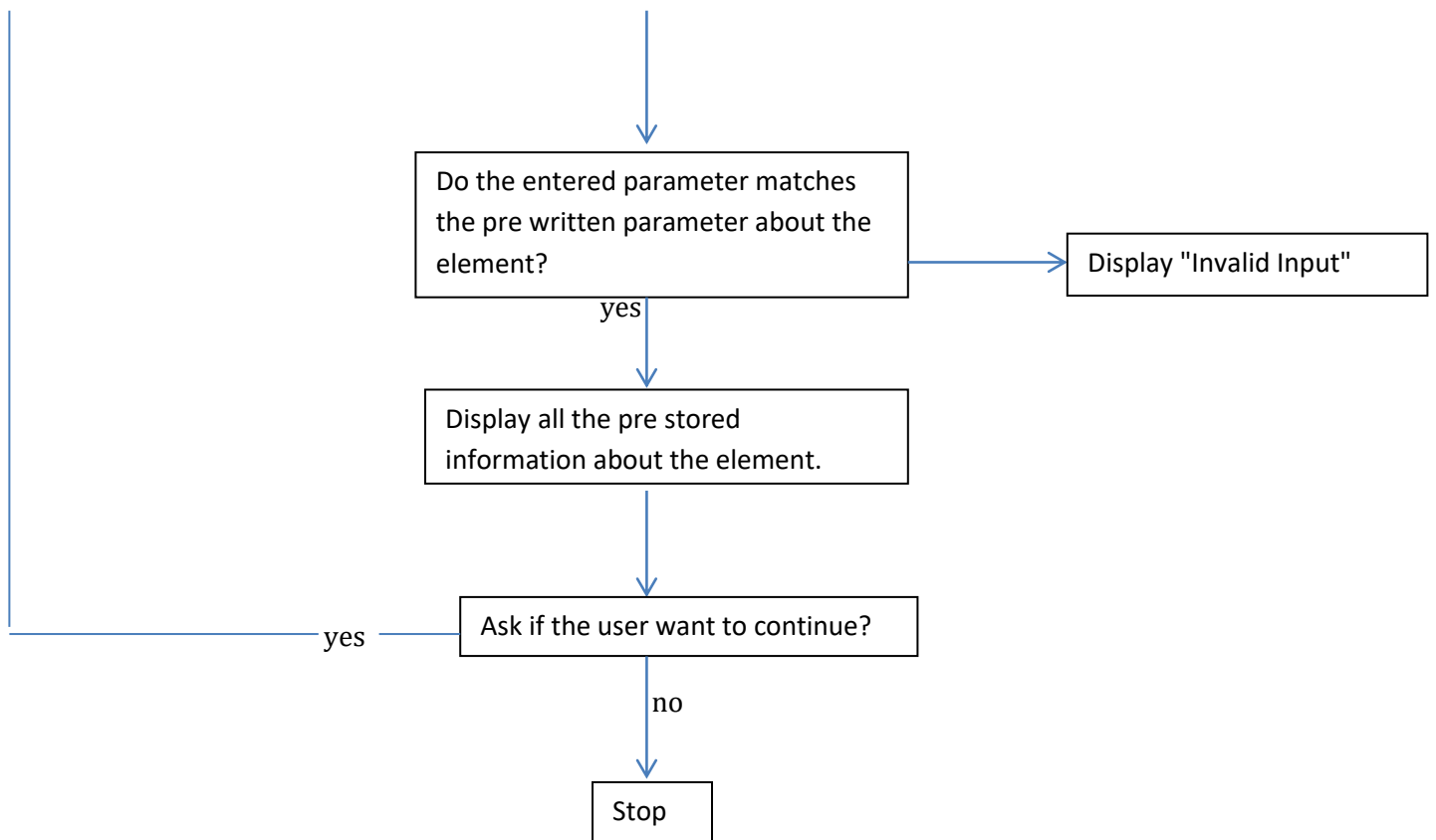
Step 10: Ask if the user wants to continue or not.

Step 11: If yes then go to step 3 else go to step 12.

Step 12: Stop.

SYSTEM FLOW





CODE

```
/*
```

```
* A digital periodic table that searches an element using search by atomic name, atomic number,  
atomic weight and atomic symbol.
```

```
Also displays an element's metallic property and the block in which the element is present.
```

```
#include<stdio.h>
```

```
#include<string.h>
```

```
#include<stdlib.h>
```

```
int get_option(void);
```

```
int atomic_number(void);
```

```
int atomic_symbol(void);
```

```
int atomic_name(void);
```

```
int atomic_weight(void);
```

```
char block(int);
```

```
char property(int);
```

```
void show(int,char,char);
```

```
/*Global Array Declarations and initialization.*/
```

```
char name[112][20]={ "Hydrogen", "Helium", "Lithium", "Beryllium", "Boron", "Carbon", "Nitrogen",  
"Oxygen", "Flourine", "Neon", "Sodium", "Magnesium", "Aluminium", "Silicon", "Phosphorus",  
"Sulphur", "Chlorine", "Argon", "Potassium", "Calcium", "Scandium", "Titanium", "Vanadium",  
"Chromium", "Manganese", "Iron", "Cobalt", "Nickel", "Copper", "Zinc", "Gallium", "Germanium",  
"Arsenic", "Selenium", "Bromine", "Krypton", "Rubidium", "Strontium", "Yttrium", "Zirconium",  
"Niobium", "Molybdenum", "Technetium", "Ruthenium", "Rhodium", "Palladium", "Silver", "Cadmium", "
```

```

Indium","Tin","Antimony","Tellurium","Iodine","Xenon","Cesium","Barium","Lanthanum","Cerium",
"Praseodymium","Neodymium","Promethium","Samarium","Europium","Gadolinium","Terbium","D
ysprosium","Holmium","Erbium","Thulium","Ytterbium","Lutetium","Hafnium","Tantalum",
"Tungsten","Rhenium","Osmium","Iridium","Platinum","Gold","Mercury","Thallium","Lead",
"Bismuth","Polonium","Astatine","Radon","Francium","Radium","Actinium","Thorium",
"Protactinium","Uranium",
"Neptunium","Plutonium","Americium","Curium","Berkelium","Californium","Einsteinium",
"Fermium","Mendelevium","Nobelium","Lawrencium","Rutherfordium","Dubnium","Seaborgium",
"Bohrium","Hassium","Meitnerium","Darmstadtium","Roentgenium","Copernicium"};

```

```

char symbol[112][5]={ "H","He","Li","Be","B","C","N","O","F","Ne","Na","Mg","Al","Si","P","S",
"Cl","Ar","K","Ca","Sc","Ti","V","Cr","Mn","Fe","Co","Ni","Cu","Zn","Ga","Ge","As","Se","Br","Kr","Rb",
"Sr","Y","Zr","Nb","Mo","Tc","Ru","Rh","Pd","Ag","Cd","In","Sn","Sb","Te","I","Xe","Cs","Ba","La","Ce",
"Pr","Nd","Pm","Sm","Eu","Gd","Tb","Dy","Ho","Er","Tm","Yb","Lu","Hf","Ta","W","Re","Os","Ir","Pt",
"Au","Hg","Tl","Pb","Bi","Po","At","Rn","Fr","Ra","Ac","Th","Pa","U","Np","Pu","Am","Cm","Bk","Cf",
"Es","Fm","Md","No","Lr","Rf","Db","Sg","Bh","Hs","Mt","Ds","Rg","Cn"};

```

```

double weight[112]={1.00,4.00,6.94,9.01,10.81,12.01,14.00,16.00,20.00,20.18,23.00,24.31,26.99,
28.08,30.97,32.06,35.45,39.09,40.07,44.95,47.86,50.94,51.99,54.93,55.84,58.69,58.93,63.54,65.39,
69.7,72.64,74.92,78.96,79.90,83.80,85.46,87.62,88.90,91.22,92.90,95.94,98.00,101.07,102.90,
106.42,107.86,112.41,114.81,118.71,121.76,127.6,126.90,131.29,13.90,137.32,138.90,140.11,
140.90,144.24,145.00,150.36,151.96,157.25,158.92,162.50,164.93,167.5,168.93,173.04,174.96,
178.49,180.94,183.84,186.20,190.23,192.21,195.07,196.97,200.59,204.38,207.20,208.98,209.00,
210.00,222.00,223.00,226.00,227.00,231.03,232.03,237.00,238.03,243,244,247,247,251,252,257,2
58,259,261,262,264,266,268,272,277,276,281,280,285};

```

```

/*Main method*/

```

```

Int main(void)
{
    int i=999,option;

    do

    {

        option=get_option();

        char ch1,ch2;

```

```

switch (option)
{
    case 1: i=atomic_number();
        break;

    case 2: i=atomic_symbol();
        break;

    case 3: i=atomic_name();
        printf("%d",i);
        break;

    case 4: i=atomic_weight();
        break;

    case 5: return (0);
    default: system("cls");
        printf("\n\n\t\t\tWrong Choice!\n");
        printf("\n\t\t\tLet's try it again, Shall we ?\n\n");
}

if (i== -1)
{
    system("cls");
    printf("\t\t\t\tOops ! \n\t\t\t\tNo Such element present !\n");
    printf("\t\t\t\t Make a right choice this time !\n\n");
}

```

```

        else if (i==999)
            continue;
        else
        {
            system("cls");
            ch1=block(i);
            ch2=property(i);
            show(i,ch1,ch2);
        }
        i=999;
    }
    while (option!=5);
    return (0);
}

/*Gets the choice of search from the user*/

int get_option(void)
{
    int option;
    printf("\n\nEnter the corresponding number :\n\n");
    printf("1. Search by ATOMIC NUMBER\n");
    printf("2. Search by ATOMIC SYMBOL\n");
    printf("3. Search by ATOMIC NAME\n");
    printf("4. Search by ATOMIC WEIGHT\n");
    printf("5. QUIT\n");

```

```
    printf("\nEnter Your Choice : ");  
    scanf("%d",&option);  
    return option;  
}
```

```
/*Method to search element by atomic number*/
```

```
/*Returns the index of the element*/
```

```
int atomic_number(void)
```

```
{  
    int n;  
    printf("\nEnter the Atomic Number you want to search for : ");  
    scanf("%d",&n);  
    n--;  
    if (n>=0 && n<112)  
        return n;  
    else  
        return -1;  
}
```

```
/*Method to search element by atomic name*/
```

```
/*Returns the index of the element*/
```

```
int atomic_name(void)
```

```
{
```

```

char x[20];

int i,flag=0;

printf("\nEnter the Atomic Name you want to search for : ");

scanf("%s",x);

for (i=0;i<112;i++)
{
    if (strcmp(x,name[i])==0)
    {
        flag=1;
        break;
    }
}

if (flag==1)
return i;

else
return -1;
}

```

/*Method to search element by atomic symbol*/

/*Returns the index of the element*/

```

int atomic_symbol(void)
{
    char x[5];

    int i,flag=0;

    printf("\nEnter the Atomic Symbol you want to search for : ");

```



```

scanf("%s",x);
for (i=0;i<112;i++)
{
    if (strcmp(x,symbol[i])==0)
    {
        flag=1;
        break;
    }
}
if (flag==1)
return i;
else
return -1;
}

/*Method to search element by their corresponding atomic weight */
/*Returns the index of the element*/

int atomic_weight(void)
{
    double x;
    int i,flag=0;
    printf("\nEnter the Atomic Weight you want to search for : ");
    scanf("%lf",&x);
    for (i=0;i<112;i++)
    {

```

```

        if (weight[i]==x)
        {
            flag=1;

            break;
        }
    }
    if (flag==1)
        return i;
    else
        return -1;
}

/*Method to find the block in which the element falls*/

/*Returns the index of the element*/

char block(int i)
{
    char ch=' ';

    int n=i+1;

    if(n==1 || n==3 || n==4 || n==11 || n==12 || n==19 || n==20 || n==37 || n==38 || n==55 || n==56 ||
n==87 || n==88)

        ch = 's';

    else if(n==2 || (n>=5 && n<=10) || (n>=13 && n<=18) || (n>=31 && n<=36) || (n>=49 && n<=54)
|| (n>=81 && n<=86))

        ch = 'p';

    else if((n>=21 && n<=30) || (n>=39 && n<=48) || (n>=72 && n<=80) || (n>=104 && n<=112))

        ch = 'd';

```

```

else
    ch='f';
return ch;
}

/*Method to find the metallic property of the element*/
/*Returns the index of the element*/

char property(int i)
{
    int n=i+1;
    char ch=' ';
    if (n==1 || n==2 || (n>=6 && n<=10) || (n>=15 && n<=18) || (n>=34 && n<=36) || n==53 ||
n==54 ||n==86)
        ch='n';
    else if (n==5 || n==14 || n==32 || n==33 || n==51 ||n==52 ||n==85)
        ch='s';
    else
        ch='m';
    return ch;
}

/*Displays the search result*/
/*Displays the Atomic number, name, symbol, weight, metallic property and block*/

void show(int i,char ch1,char ch2)

```


OUTPUT

The Main User Interface:



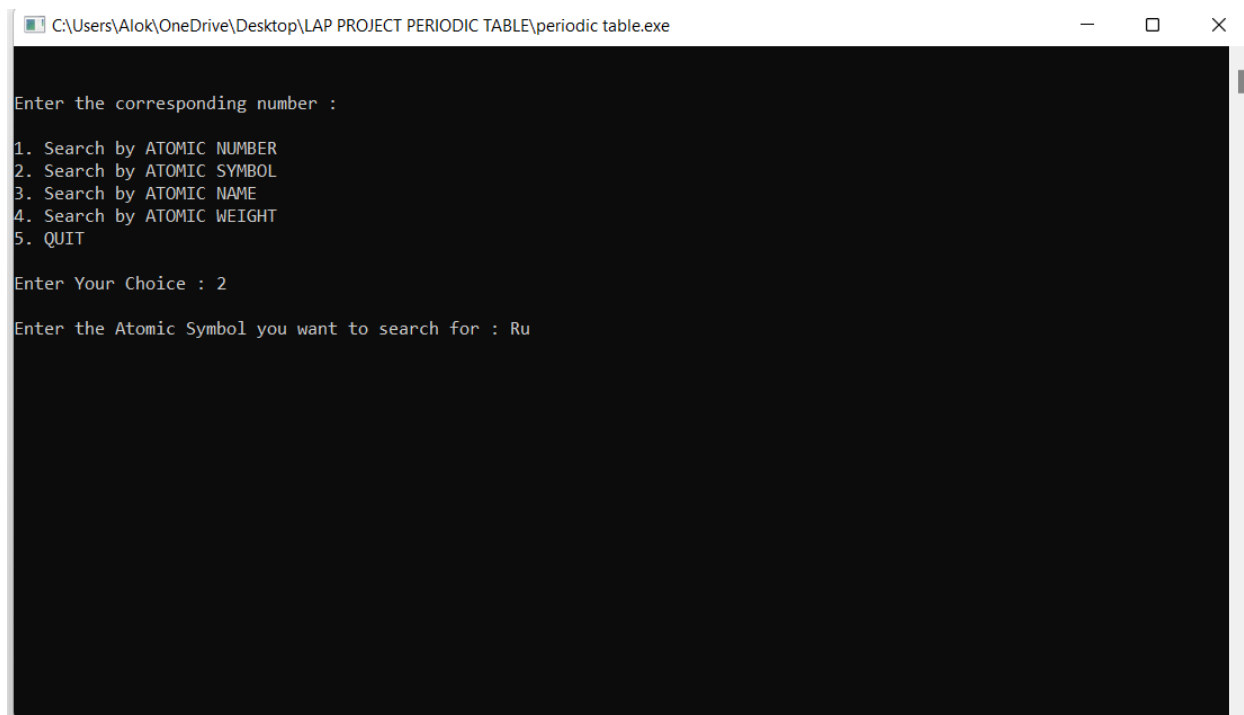
```
Select C:\Users\Alok\OneDrive\Desktop\LAP PROJECT PERIODIC TABLE\periodic table.exe

Enter the corresponding number :

1. Search by ATOMIC NUMBER
2. Search by ATOMIC SYMBOL
3. Search by ATOMIC NAME
4. Search by ATOMIC WEIGHT
5. QUIT

Enter Your Choice : _
```

Let's see an example of program searching the element with symbol Ru as input:



```
C:\Users\Alok\OneDrive\Desktop\LAP PROJECT PERIODIC TABLE\periodic table.exe

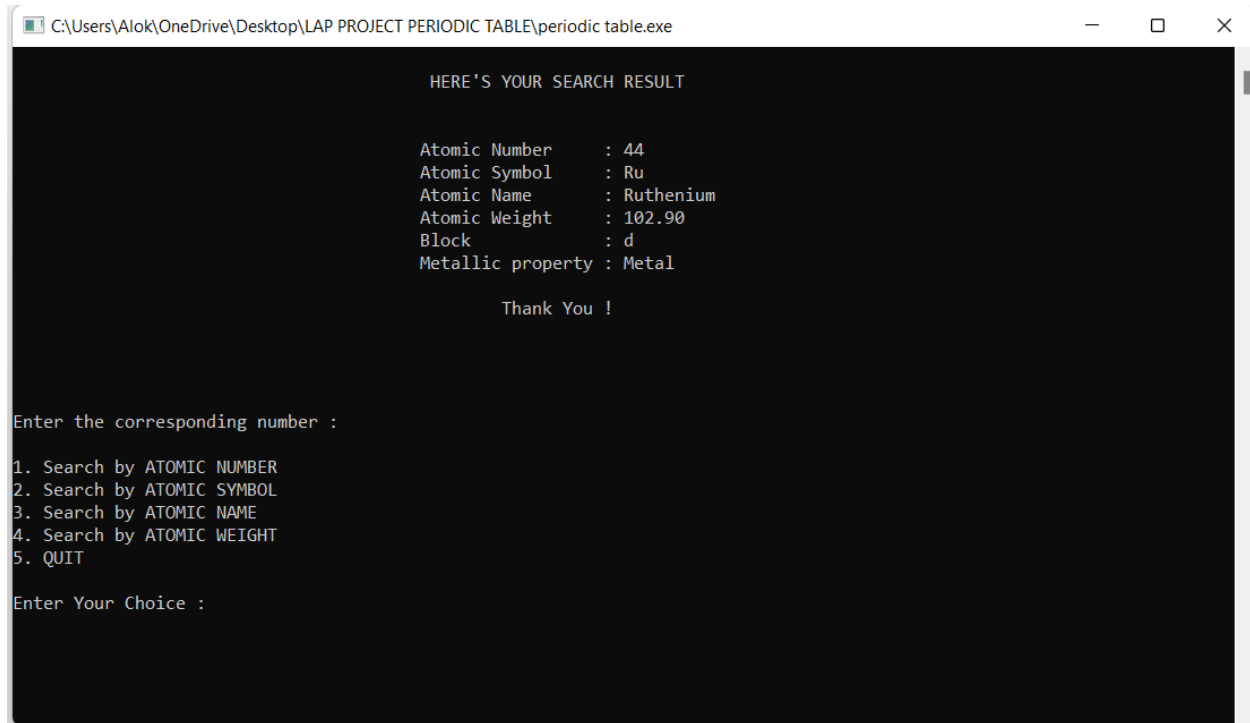
Enter the corresponding number :

1. Search by ATOMIC NUMBER
2. Search by ATOMIC SYMBOL
3. Search by ATOMIC NAME
4. Search by ATOMIC WEIGHT
5. QUIT

Enter Your Choice : 2

Enter the Atomic Symbol you want to search for : Ru
```

It's output:



```
C:\Users\Alok\OneDrive\Desktop\LAP PROJECT PERIODIC TABLE\periodic table.exe

                                     HERE'S YOUR SEARCH RESULT

Atomic Number      : 44
Atomic Symbol      : Ru
Atomic Name        : Ruthenium
Atomic Weight      : 102.90
Block              : d
Metallic property  : Metal

                               Thank You !

Enter the corresponding number :

1. Search by ATOMIC NUMBER
2. Search by ATOMIC SYMBOL
3. Search by ATOMIC NAME
4. Search by ATOMIC WEIGHT
5. QUIT

Enter Your Choice :
```

Now let's provide an invalid input:

```
Enter the corresponding number :
```

1. Search by ATOMIC NUMBER
2. Search by ATOMIC SYMBOL
3. Search by ATOMIC NAME
4. Search by ATOMIC WEIGHT
5. QUIT

```
Enter Your Choice : 6
```

It's output:

```
Wrong Choice!
```

```
Let's try it again, Shall we ?
```

```
Enter the corresponding number :
```

1. Search by ATOMIC NUMBER
2. Search by ATOMIC SYMBOL
3. Search by ATOMIC NAME
4. Search by ATOMIC WEIGHT
5. QUIT

```
Enter Your Choice :
```

CONCLUSION

Lastly this mini project was successfully accomplished with the constant effort of our team members and various references used. After completion of this project, we are now able to tackle most of the problems in our syllabus and are much more clear about all the basic concepts of C Programming like looping, array and string, formatted I\O, operators etc. Hopefully, this mini program can help students and its users to get the precise, accurate and quick knowledge of all the elements enlisted in the periodic table. We heartily welcome any kind of suggestions that would be appropriate for the project.

ACKNOWLEDGEMENT

Developing this project was challenging and we spent a considerable amount of time, brainstorming for the proper functioning of the program. The project wouldn't have all the accurate information about the elements if we hadn't used the externally available periodic table. Completing this project on time would have been a prick in the flesh without the collective effort and teamwork of our fellow teammates. We would also like to thank Department of Computer Science for letting us have this wonderful opportunity to showcase our knowledge on C programming and our team working capability to successfully tie up this wonderful output as "The Periodic Table". We cannot stop thanking our faculties who taught us about the concepts of C programming both in theory and practical. Our deepest gratitude goes to the makers of the periodic table online on google chrome, writers of the Basic Chemistry-11, websites like Tutorialspoint and Programiz. The successful execution of this program would have been impossible without these resources. Finally, we will like to provide our sincere gratitude to all the helping hands who suggested and helped with various ideas and helps.

REFERENCES

- 1) <https://www.tutorialspoint.com/>
- 2) <https://www.programiz.com/c-programming>
- 3) https://www.angelo.edu/faculty/kboudrea/periodic/structure_numbers.htm
- 4) Basic Chemistry-11
- 5) Our friends of BEL-078(AB)