

CHAPTER 1

INTRODUCTION

1.1 General Introduction

We know about the periodic table we studied in chemistry. The periodic table not easy to remember all symbol with atomic number. So i made a project about modern periodic table which is may helpful for student.

Student can easily find the name, symbol, atomic number, atomic weight, and some important properties in one click.

It is developed using the C programming language for the purpose of storing name, symbol, atomic number, atomic weight, and some important properties as well as to display them as per requirement of the user.

The source code for this project is complete and totally error free. The C source code is simple and understandable, and consists of around 550 lines of code.

In this project we can create a file using file pointer ,we can not only accessing the stored data in file but also modifying and remove the stored data from the file.

1.2 Problem Statement

In the periodic table Each element name is represented by an element symbol, it could be first letter of the name of the element in English, Latin, or German. For example

- C for carbon
- K for *kalium* (potassium in Latin)

The capitalized first letter followed by another lowercase letter from the name of the element in English or Latin. For example

- Si for silicon

a systematic element name and symbol — essentially a placeholder.

In the table there is two number are usually added to each cell.

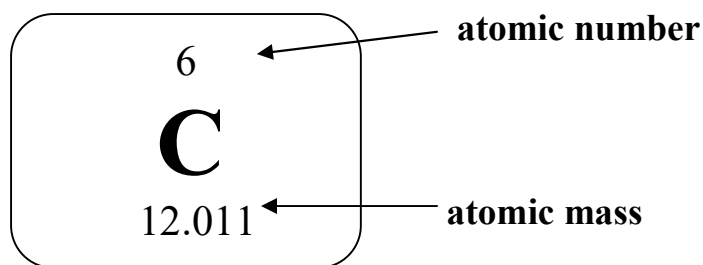


Fig 1.1 Atomic Structure

I this process the symbol and atomic mass are difficult to find and remember also.

1.3 Existing System

Modern periodic table is a innovative thought to simplify the how to search a certain data using name,symbol,atomic number etc.

In this present system the user first insert some data and after that that will be store in file, after that these data are fetch from file give output result in your screen.

Following details are maintain by the user

- Insert data with name,atomic number,atomic mass,symbol
- Explore data that means it will fetch from file

1.4 Proposed System With Methodology

In the previous system , details are store Manually. To share the details between the file and explore function.

Functional and advantage :

- Data is Centralized which has overcome the shearing problem
- As data is maintained electronically ,its's easy for update
- Maintenance is easy and performance is good
- Mainly the system has automated

All details are given by the user which is need name ,symbol,atomic number and atomic mass. All the details are store in database .

Whenever the user need to search the data just type its name ,symbol, number etc.

Name	Symbol	Atomic number	Atomic mass	Block	properties
Hydrogen	H	1	1.0079	1	gas
Helium	He	2	4.0026	18	gas
Lithium	Li	3	6.9411	1	solod
Beryllium	Be	4	9.0122	2	solid
Boron	B	5	10.811	13	solid
Carbon	C	6	12.0107	14	Solid
Nitrogen	N	7	14.0067	15	gas

Table 1.1 Element list Table

1.5 Objective of the Work

This project is to create to search the element periodic table easily. To search for the element by periodic name you have type **“Element Name”- “Atomic Number”**. Also, as mentioned you can search the elements by their atomic number. After you search the element it will show you the results by displaying their details.

You can search an element by using any of the following method:

1. By name of element
2. By symbol of element
3. By atomic number of element
4. By atomic weight of element

1.6 Feasibility Report

Preliminary investigation examine project feasibility, the likelihood the system usefull for the student .The main objective of the feasibility study is to test the technical operation and adding the new data module and debugging old paper periodic table.

CHAPTER 2

REVIEW OF LITERATURE

In 1864 a German chemist Julius Lothar Meyer organized the element by atomic mass and grouped them according to there chemical properties .

Latter that decade a Russian chemist Dimitri Mendeleev organized all the known elements according to similar properties. He left gaps in his table for what he thought were undiscovered elements, and he made some bold predictions regarding the properties of those undiscovered elements. When elements were later discovered whose properties closely matched Mendeleev's predictions .

Because certain properties of the elements repeat on a regular basis throughout the table (that is, they are periodic), it became known as the periodic table.

Mendeleev, who first published his periodic table in 1869, is usually credited with the origin of the modern periodic table.

Features of the Periodic Table

Elements that have similar chemical properties are grouped in columns called groups .

As well as being numbered, some of these groups have names for example, alkali metals (the first column of elements), alkaline earth metals (the second column of elements). Each row of elements on the periodic table is called a period. Periods have different lengths, the first period has only 2 elements (hydrogen and helium), while the second and third periods have 8 elements each. The fourth and fifth periods have 18 elements each, and later periods are so long that a segment from each is removed and placed beneath the main body of the table.

Group Period →	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
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	57 La	* 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	89 Ac	* 104 Rf	105 Db	106 Sg	107 Bh	108 Hs	109 Mt	110 Ds	111 Rg	112 Cn	113 Nh	114 Fl	115 Mc	116 Lv	117 Ts	118 Og
				* 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	
				* 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	

Fig 2.1 Simple Periodic Table

- The periodic table is an arrangement of the elements in order of increasing atomic number.
- Elements that exhibit similar chemistry appear in vertical columns called groups (numbered 1–18 from left to right); the seven horizontal rows are called periods.
- Some of the groups have widely-used common names, including the alkali metals (Group 1) and the alkaline earth metals (Group 2) on the far left, and the halogens (Group 17) and the noble gases (Group 18) on the far right.
- The elements can be broadly divided into metals, nonmetals, and metalloids.
- Semi metals exhibit properties intermediate between those of metals and nonmetals.
- Metals are located on the left of the periodic table, and nonmetals are located on the upper right. They are separated by a diagonal band of metalloids.

CHAPTER 3

SYSTEM CONFIGURATION

3.1 Hardware Requirements

For this project it need minimum system hardware requirement for run this program properly.

- Operating System: Windows 2000/NT/Xp/Vista/7
- RAM: 256 MB or more
- HARD DISK 40 GB or more
- Processor P3 or High
- Compiler Standard C Compiler

Developer++ compiler is a free full-featured Integrated Development Environment (IDE) distributed under the GNU General Public License for programming in c and C++. It is written in Delphi. It is bundled with, and uses, the MinGW or TDM-GCC 64bit port of the GCC as its compiler. Dev-C++ can also be used in combination with Cygwin or any other GCC-based compiler. Dev-C++ is generally considered a Windows-only program, but there are attempts to create a Linux version: header files and path delimiters are switchable between platforms.

3.2 Software Requirements

The code should be compiled in Code::Blocks or TurboC++ as the Integrated Development Environment(IDE) in which the project was developed is Code::Blocks using GCC compiler.

It batter to run on TurboC++ because it support all type of header file which is preinstall in TurboC++ but in Code::Block you need to install some IDE or may include some extra header file.

The project is done by Code::Block compiler, we can compile this with another compiler also as code::block is more comfortable so I used this compiler.

CHAPTER 4

MODULE DESCRIPTION

It has two important module

- Storage of Element Information:
- Exploration of element Information:

4.1 Storage of Element Information

In the project, you can add any new element with its name, symbol, atomic number, atomic weight and its some important properties. When new element information is to be added to this Modern Periodic Table, you have to enter 1 in the main menu and input information in given format. These information are stored in file created on the hard disk of computer by program itself.

After press 1 you see this type of user input screen where you have to input these information like Name, Symbol, Atomic No, Atomic Weight ,etc.

After successfully entered all information it will store in a file for further process.

In this below function user can add the data

- void add():
 - This function is used to input or add the information of new element to the program.
 - By this function user can add the name,symbol,atomic number etc.

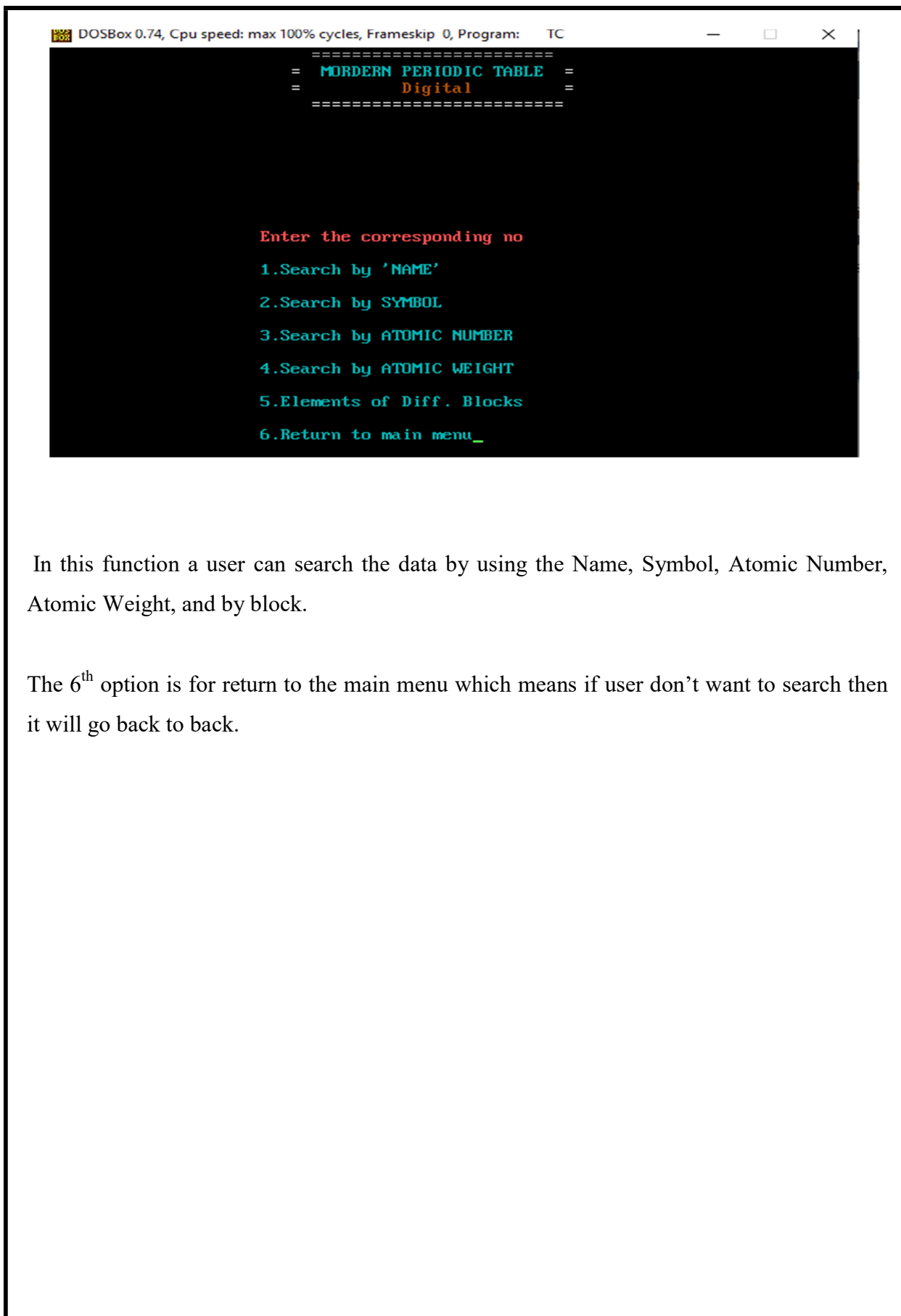
4.2 Exploration of Element Information

Another main function of this project is to explore or to display the stored information. You can search an element by using any of the following method:

1. By name of element
2. By symbol of element
3. By atomic number of element
4. By atomic weight of element

I am using below main function to designing my project

- void explor():
 - This function is used to explore the stored information in the file created.
 - In this function first a prompt will shown that is user can search the an item by there name, symbol, properties ,atomic number etc.
 - That is the stored information or data are display with there all properties.
- void mainscreen():
 - It is included in source code of project file in order to print the text style and to control its color.
 - It implement for design the project and for attractive.
- Void print():
 - This function is used to print the main screen or menu of the project.
- Int main():
 - In main function used for called the user define function and print error message if user select wrong choice.



In this function a user can search the data by using the Name, Symbol, Atomic Number, Atomic Weight, and by block.

The 6th option is for return to the main menu which means if user don't want to search then it will go back to back.

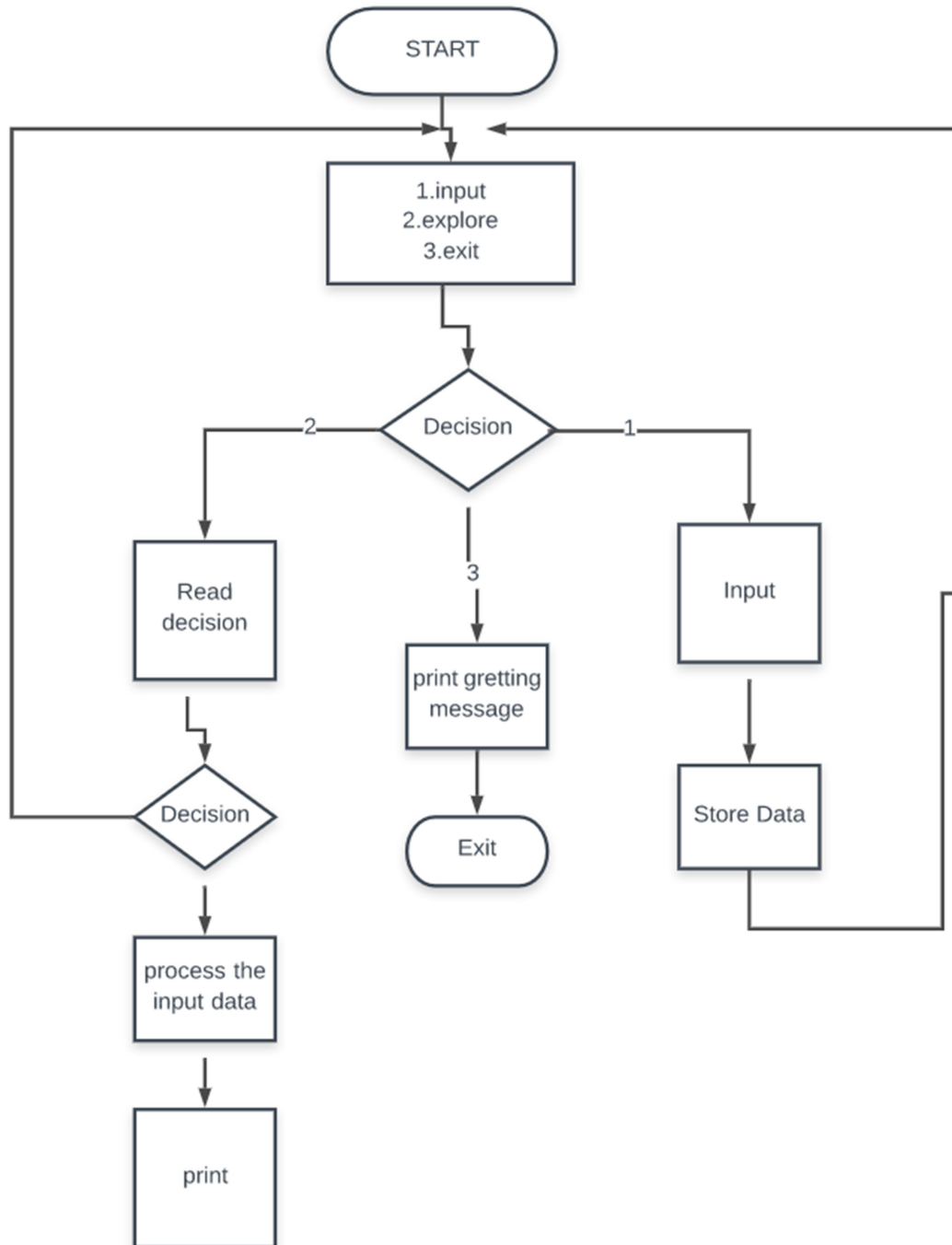
CHAPTER 5**SYSTEM DESIGN**

Fig 5.1 Data flow Diagram

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 Implementation

The project was implemented by c programming language using data structure.

In this project my aim is to store data in a file and fetch these data from file when needed.

Before implement the project we have to download Code::block software to develop the program.

6.1.1 Pre-Implementation

Download the code::block and install in your system after install the software you have to the IDE(Integrated Development Environment) and run a program to know IDE work properly or not.

6.1.2 Post-Implementation

In these project I am creating file name as PROJECT.C file to save these project. This is pure console basic program.

In this project I was using a user define not primitive data type which is structure, and create some member variable of structure.

These member variable are like name, symbol, atomic number, properties etc. I have three function in this project 1.add function to add new data into file 2.explore which is used to explore or view the stored data and another for exit function. These function are user define function. Inside the add function a console screen is open which is takes input from user to add data like name , symbol ,etc.

6.2 SOURCE CODE

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
#include <windows.h>
#include<iostream>
using namespace std;
struct element{
    char name[20];
    char sb[5];
    int atm;
    float atms;
    char block;
    char atc[20];
    char prop[250];
}p,q;
int rw,cl;
FILE*fp;
void add();
void explor();
void print();
void mainscreen();
void gotoxy(int,int);
void textcolor(int);
int main()
{
    int a,i,n,y;
    char c,d;
    char date1[15],date2[15],string1[20];
    unsigned int tsz;
```

```
mainscreen();
label1:
gotoxy(25,9);
textcolor(14);
cprintf("ENTER THE CORRESSPONDIND NO ->");
gotoxy(25,9);
textcolor(11);
cprintf("\n-----\n");

gotoxy(22,11);textcolor(10);
cprintf("1.Add new Element Information");

gotoxy(22,13);
cprintf("2.Explore");
gotoxy(22,15);
cprintf("3.Quit");
gotoxy(22,17);
fflush(stdin); /*Used to clear the buffer and accept next string */
d=getch();
switch(d)
{
    case '1': /*add item */
    {
        add();
        break;
    }
    case '2': /* Explor */
    {
        explor();
        break;
    }
    case '3': /* Quit */
```

```
        {
            //clrscr();
            mainscreen();
            textcolor(14);
            gotoxy(30,20);
            cprintf("THANK U.....");
            gotoxy(30,22);
            cprintf("BYE.....");
            getch();
            exit(1);
            break;
        }
    default:
        {
            //clrscr();
            mainscreen();
            textcolor(12+128);
            gotoxy(29,12);
            cprintf("Wrong choice");
            gotoxy(29,13);textcolor(11);
            cprintf("Retype choice");
            goto label1;
        }
    }
    mainscreen();
    goto label1;
    return 0;
}

void mainscreen()
{
    int i,j;
    //clrscr();
    gotoxy(27,1);textcolor(7);
    cprintf("=====");
```

```

gotoxy(28,2);textcolor(3);
printf("MORDERN PERIODIC TABLE");

gotoxy(36,3);textcolor(6);
printf("Digital");

gotoxy(27,4); textcolor(15);
printf("=====");
gotoxy(25,3);
printf("=");gotoxy(25,2);
printf("=");gotoxy(25,1);

gotoxy(52,2);
printf("=");gotoxy(52,3);
printf("=");
}
void add()
{
    char ch;
    label1:
    mainscreen();

    gotoxy(15,8);textcolor(11);
    printf("Enter the Information of Elements:"); gotoxy(14,9);
    printf("=====");

    getch();
    gotoxy(15,10);textcolor(10);
    printf("Name:");

    gotoxy(15,12);
    printf("Symbol:");

    gotoxy(15,14);
    printf("Atomic No: ");
    gotoxy(15,16);
    printf("Atomic Wt: ");

    gotoxy(15,18);
    printf("Atomic Config:");

    gotoxy(15,20);
    printf("Block:");

    gotoxy(15,22);
    printf("Properties:");

    textcolor(2);

```



```
fflush(stdin);
gotoxy(20,10);
scanf("%s",p.name);

fflush(stdin);
gotoxy(23,12);
scanf("%c",&p.sb);

fflush(stdin);
gotoxy(25,14);
scanf("%d",&p.atm);

fflush(stdin);
gotoxy(25,16);
scanf("%f",&p.atms);

fflush(stdin);
gotoxy(29,18);
scanf("%s",p.atc);

fflush(stdin);
gotoxy(21,20);
scanf("%c",&p.block);

if(p.block!='S'&&p.block!='P'&&p.block!='D'&&p.block!='F')
p.block=' ';
fflush(stdin);

gotoxy(26,22);
scanf("%s",p.prop);

if((fp=fopen("data","ab+"))==NULL)
{
    printf("Cannot open the file");
    getch();
    exit(1);
}
fwrite(&p,sizeof(p),1,fp);
fclose(fp);
printf("\n\n\t\tEnter 'y' for next record(y/n):");
ch=getch();
if(ch=='y')
{
    goto label1;
}

}
void explor()
```

```
{
    char d,c;
    FILE *f;
    int given_atmic_no,a,i,tsz,n;
    float given_atmic_mass;
    int flag;
    char string[20];
    startofexplore:
    //clrscr();
    mainscreen();
    label6:
    gotoxy(22,12);textcolor(12);
    cprintf("Enter the corresponding no");
    gotoxy(22,14);textcolor(3);
    cprintf("1.Search by 'NAME'");
    gotoxy(22,16);
    cprintf("2.Search by SYMBOL");
    gotoxy(22,18);
    cprintf("3.Search by ATOMIC NUMBER");
    gotoxy(22,20);
    cprintf("4.Search by ATOMIC WEIGHT");
    gotoxy(22,22);
    cprintf("5.Elements of Diff. Blocks");
    gotoxy(22,24);
    cprintf("6.Return to main menu");
    gotoxy(25,26);
    fflush(stdin);
    d=getch();
    switch(d)
    {
        case '1':
        {
            //clrscr();
            mainscreen();

            gotoxy(15,20);
            textcolor(12);
            cprintf("Enter the Name of Element:");

            textcolor(3);
            fflush(stdin);
            scanf("%s",string);
            printf("%s",string);

            if((fp=fopen("data","rb+"))==NULL)
            {
                //clrscr();
```

```
        printf("\n cannot open the record file 1");
        getch();
        exit(1);
    }
    flag=1;
    while(fread(&p,sizeof(p),1,fp))
    {
        if(strcmp(p.name,string)==0)
        {
            print();

            flag=0;

            break;
        }
    }
    if(flag==1)
    {
        //clrscr();
        mainscreen();
        gotoxy(25,25);
        textcolor(12);
        cprintf("::No Element Available::");

    }

    fclose(fp);
    getch();
    break;
}
case '2':
{
    //clrscr();
    mainscreen();

    gotoxy(22,15);
    textcolor(12);
    cprintf("Enter the symbol:");

    textcolor(3);
    fflush(stdin);
    scanf("%s",string);
    printf("%s",string);

    if((fp=fopen("data","rb+"))==NULL)
    {
        //clrscr();
```

```
printf("\n cannot open the record file 1");
getch();
exit(1);
}
flag=1;
while(fread(&p,sizeof(p),1,fp))
{
    if(strcmp(p.sb,string)==0)
    {
        print();
        flag=0;
        break;
    }
}
if(flag==1)
{
    //clrscr();
    mainscreen();
    gotoxy(25,25);
    textcolor(12);
    cprintf("::No Element Available::");
}

fclose(fp);
getch();
break;
}
case '6':
{
    return;
}
case '3':
{
    //clrscr();
    mainscreen();
    gotoxy(15,20);
    textcolor(12);
    cprintf("Enter the Atomic No. Element:");
    textcolor(3);
    fflush(stdin);
    scanf("%d",&given_atmic_no);
    if((fp=fopen("data","rb+"))==NULL)
    {
        printf("\n cannot open the record file 1");
        getch();
    }
}
```

```
        exit(1);
    }
    flag=1;
    while(fread(&p,sizeof(p),1,fp))
    {
        if(p.atm==given_atmic_no)
        {
            print();
            flag=0;

            break;
        }
    }
    if(flag==1)
    {
        //clrscr();
        mainscreen();
        gotoxy(25,25);
        textcolor(12);
        cprintf("::No Element Available::");
    }

    fclose(fp);
    getch();
    break;
}
case '4':
{
    //clrscr();
    mainscreen();
    gotoxy(15,22);
    textcolor(12);
    cprintf("Enter the Atomic mass of Element:");
    textcolor(3);
    fflush(stdin);
    scanf("%f",&given_atmic_mass);
    if((fp=fopen("data","rb+"))==NULL)
    {
        //clrscr();

        printf("\n cannot open the record file 1");
        getch();
        exit(1);
    }
    flag=1;
    while(fread(&p,sizeof(p),1,fp))
```

```
        {
            if(p.atms==given_atmic_mass)
            {
                print();
                flag=0;

                break;
            }
        }
    if(flag==1)
    {
        //clrscr();
        mainscreen();
        gotoxy(25,25);
        textcolor(12);
        cprintf("::No Element Available::");

    }

    fclose(fp);
    getch();
    break;
}
case '5':
{

    clrscr();
    mainscreen();
    gotoxy(15,25);
    textcolor(12);
    cprintf("Enter the Block:");
    textcolor(3);
    fflush(stdin);
    scanf("%c",&c);
    if((f=fopen("temp","wb+"))==NULL)
    {
        clrscr();
        printf("\n cannot open the temp file 1");
        getch();
        exit(1);
    }

    if((fp=fopen("data","rb+"))==NULL)
    {
        clrscr();

        printf("\n cannot open the record file 1");
```

```

        getch();
        exit(1);
    }
    flag=1;
    while(fread(&p,sizeof(p),1,fp))
    {
        if(p.block==c)
        {
            fwrite(&p,sizeof(p),1,f);
        }
    }
    fclose(f);
    fclose(fp);

    if((f=fopen("temp","rb+"))==NULL)
    {
        printf("Cannot open the file");
        getch();
        exit(1);
    }
    fseek(f,0,SEEK_END);
    tsz=ftell(f);
    n=(int)(tsz/sizeof(p));
    for(i=0;i<(n-1);i++)
    {
        for(a=i+1;a<n;a++)
        {
            fseek(f,i*sizeof(p),SEEK_SET);
            fread(&p,sizeof(p),1,f);
            fseek(f,a*sizeof(p),SEEK_SET);
            fread(&q,sizeof(p),1,f);
            if((p.atm-q.atm)>0)
            {
                fseek(f,i*sizeof(p),SEEK_SET);
                fwrite(&q,sizeof(p),1,f);
                fseek(f,a*sizeof(p),SEEK_SET);

                fwrite(&p,sizeof(p),1,fp);
            }
        }
    }
    rewind(f);
    while(fread(&p,sizeof(p),1,f))
    {

        print();
        getch();
    }

```

```

        }
        clrscr();
        mainscreen();
        gotoxy(25,25);
        textcolor(12);
        cprintf("::No Element Available::");

        fclose(f);
        getch();
        break;
    }
    default:
    {
        clrscr();
        mainscreen();
        textcolor(12+128);gotoxy(22,11);
        cprintf("Wrong choice");gotoxy(22,13);textcolor(15);
        cprintf("Retype choice");
        goto label6;
    }
}
goto startofexplore;
}

void print()
{
    clrscr();
    mainscreen();
    gotoxy(15,12);
    cprintf("Name:");

    gotoxy(15,14);
    cprintf("Symbol:");

    gotoxy(15,16);
    cprintf("Atomic No: ");

    gotoxy(15,18);
    cprintf("Atomic Wt: ");

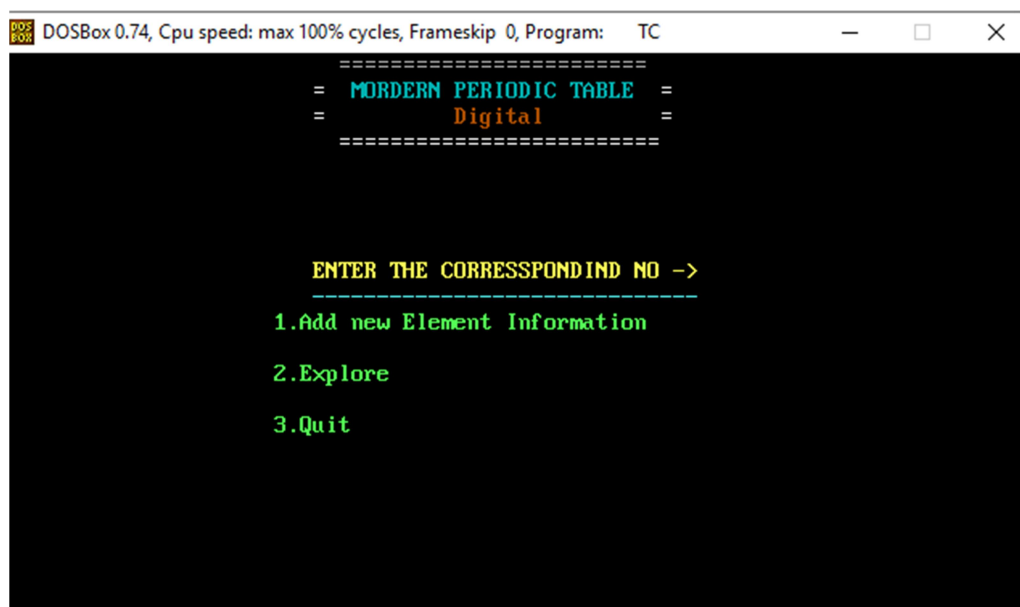
    gotoxy(15,20);
    cprintf("Atomic Config:");
    fflush(stdin);
    gotoxy(15,22);
    cprintf("Block:");

```

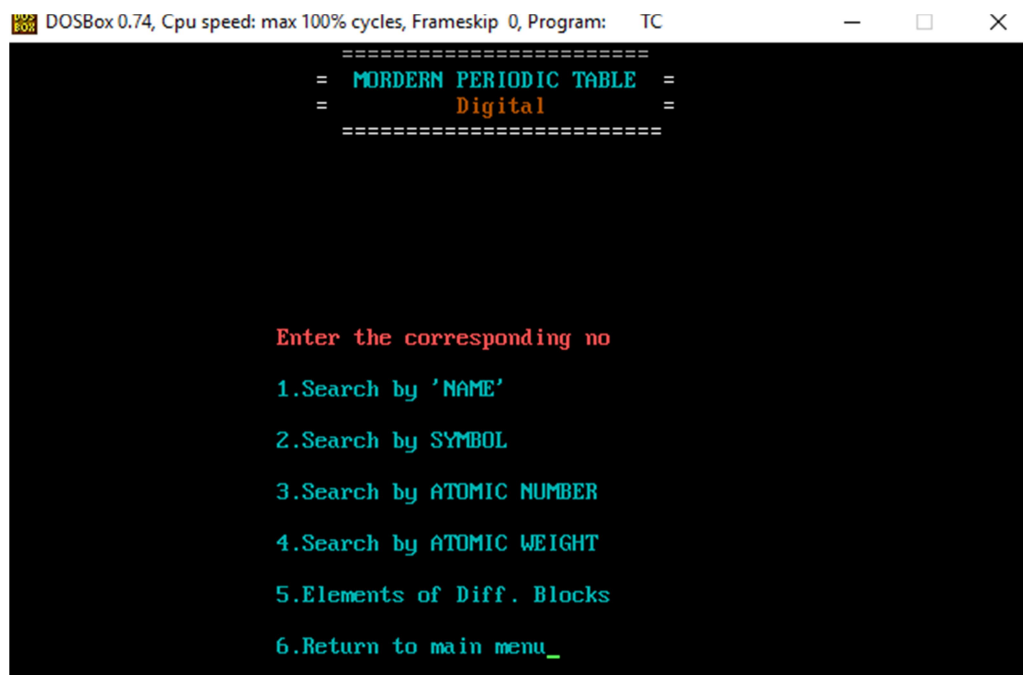


```
    gotoxy(15,24);
    cprintf("Properties:");
    textcolor(6);
    gotoxy(20,12);
    cprintf("%s",p.name);
    gotoxy(23,14);
    cprintf("%s",p.sb);
    fflush(stdin);
    gotoxy(25,16);
    cprintf("%d",p.atm);
    fflush(stdin);
    gotoxy(25,18);
    cprintf("%f",p.atms);
    fflush(stdin);
    gotoxy(29,20);
    cprintf("%s",p.atc);
    gotoxy(21,22);
    cprintf("%c",p.block);
    gotoxy(26,24);
    cprintf("%s",p.prop);
}
```

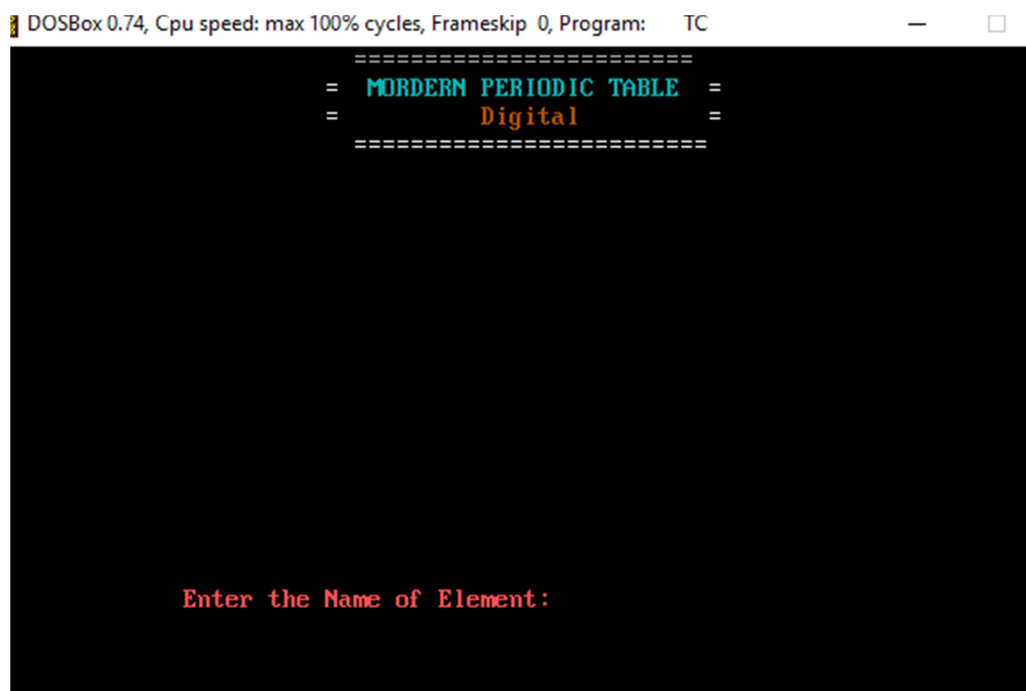
6.3 Screenshot



In the above screen shot it shows the main screen when you run the program. It will ask some condition statement that are for add the data into the file and second one is the explore the data that is describe below in another screen shot . And we have last option that is quite which will break this program and return to the program.



The fig 6.3.2 is about explore function .The explore function ask some option to input data from the user and search by what by you want to search like if you want to search by name then write the symbol name after that you get output as per the that symbol.



```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
=====
= MORDERN PERIODIC TABLE =
= Digital =
=====

Enter the Name of Element:
```

In above diagram shown that to Enter the name of Element Which is entered by the user. Whenever user Want to search an element by the name this function will run background and it will show as a output.



```
Modern Periodic Table
Digital
-----

Name :Hydrogen
Symbol: H
Atomic No:1
Atomic Wt:1.008000
Atomic Config:s Chemically active
```

CHAPTER 7

SYSTEM TESTING

System testing is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.

7.1 White Box Testing

This is the first test case which is found some error. Focuses on the functional requirements of the software .White-box testing is done at the early stages where as black-box is done at the later stages of software development

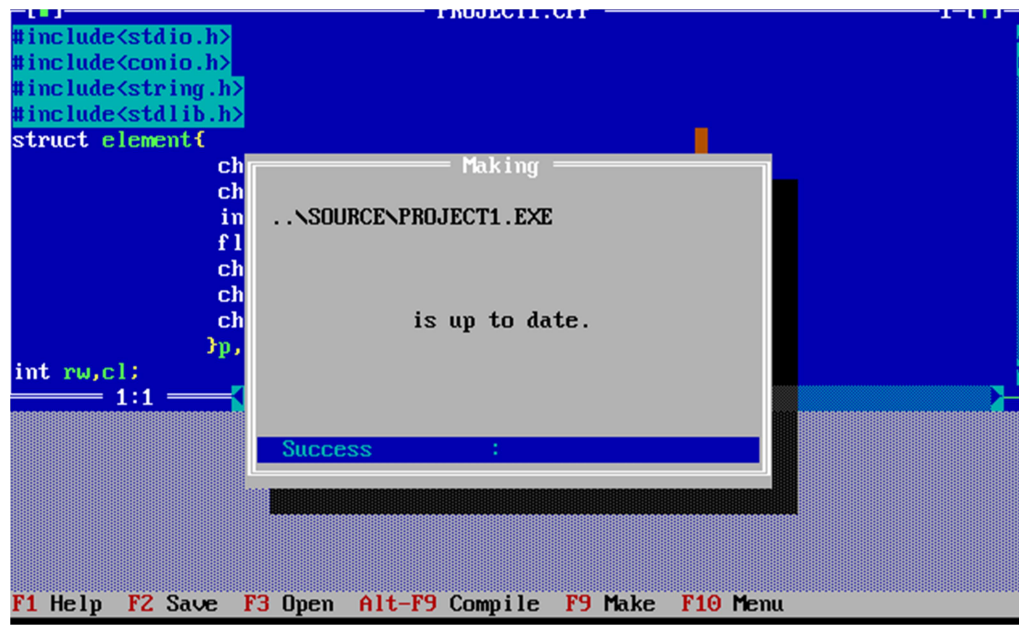


Fig – 7.2 1st Alpha test

7.2 Black Box Tesing

In the black box texting user only check the project is running error free or not.The below screen shot tells that it satisfy the black box texting.

CHAPTER 8

RESULTS AND DISCUSSIONS

8.1 Conclusion

This C mini project on Modern Periodic Table for college mini project for use of functions and file handling in C. The project formulates a clear concept of application of C programming language.

This project will help to understand file handling in C i.e. creating a file and accessing the stored data in the file, modifying and removing the stored data. It will also help you to understand the use of functions as well as different parameters of C programming language.

CHAPTER 9

REFERENCES

9.1 Text References

- Petrucci, Ralph H., William S. Harwood, F. G. Herring, and Jeffrey D. Madura. General Chemistry: Principles and Modern Applications. 9th ed. Upper Saddle River: Pearson Education, Inc., 2007.
- Sisler, Harry H. Electronic structure, properties, and the periodic law. New York: Reinhold publishing corporation, 1963.