Abstract Page

Here in this project, we are trying to make game ie; **Tower of Hanoi.** It is mathematical puzzle game consists of three rods and multiple number of disk with different diameter. In this game, the person who is playing ie; gamer have to shift disk from one rod to another rod using third rod in shortest moves. In this process, disk is not allowed to place above a disk of smaller size. We can also find the shortest moves required to complete the game by using formula 2^n-1, where n is number of disk.

C Programming language was used for coding. For better result, algorithm and flowchart were designed and used during coding process.

For this game,function, recursion, recursion function and conditional statement were used where recursion function and function helped to recall itself.

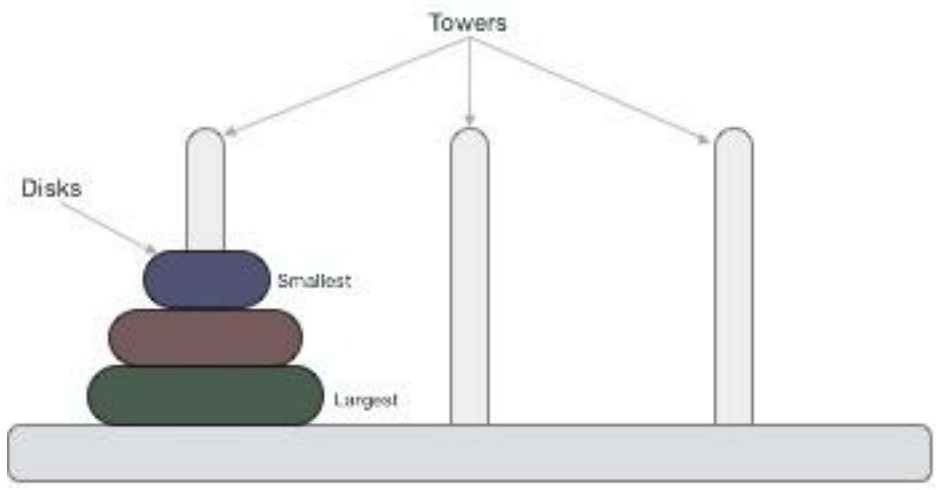


Fig 1: Tower of Hanoi

CONTENTS **Page Number**

1.Main Report

1.1 Introduction:C Programming is a general purpose, programming language that was developed int the early 1970s by Dennis Ritchie at bells lab.It was designed for system programming and has since become one of the most powerful and widely used programming language for developing on wide range of software applications. It is translated into machine level language by a compiler before being executed. It is used for developing operating system device, drivers embedded system,scientific simulations, and other application software.

1.1.1 Function: The function is a self combined block of statements that perform a coherent task of some kind. Every C program can be thought as a collection of these functions. Functions are used to break a large program into smaller manageable pieces, which manages the code and it's easier to read, understand and maintain. A function has a name written type in a list of parameters. The name of function is used to call the function from another part of the program will return to specify the type of valued the function returns after completing the task.

1.1.2 Recursion: In C-Programming it is possible for the function to call themselves. A function is called recursive within a body of function calls the same function. Sometimes call Circular definition recursion is thus the process of defining something in terms of itself.

1.2 Objectives of the Report

a) The project helps us to work in a team and helps use to learn about team work in the field of computer science. It had helped us how we students can use the knowledge of each other for better learning.

b) The project also helps us to understand the concept of C-programming.

c) The project helps us to use conditional statements, loops, functions and recursive functions.

d) The project has helped us to write a program that is able to solve the game of tower of Hanoi. This shows that we can even generate different types of the algorithm using programming languages that can be used in real life.

e) The program in the project has also helped us to understand the concept of different data types and use of the function in C-Programming.

f) The program is able to generate algorithm to solve the tower of Hanoi. This helps us to understand that we can solve different types of puzzle games using algorithms. These algorithms can be generated using computer program.

1.3 System Design & Methodology

1.3.1 Algorithm

Step 1: Start

Step 2: Declare an integer variable ‘n’

Step 3: Display a message asking the user to enter the number of disks.

Step 4: Read the input value of ‘n’ from the user.

Step 5: Check if the number of disks is less than 3 using an if condition. If it is true, display a message indicating that the user can’t play with number of disks and go to label1.

Step 6: Display a message indicating the algorithm to solve the game.

Step 7: Call the recursive function ‘towerofHanoi’ with arguments ‘n’,’A’,’C’,’B’.

Step 8: Return 0 to indicate successful program execution.

Step 9: Define the recursive function ‘towerofHanoi’ with arguments ‘n’,’s’,’d’, and ‘a’.

Step 10: Check if the number of disks is 1 using an if condition. It it is true, then print a message to move the disk 1 from ‘s’ to ‘d’ and return

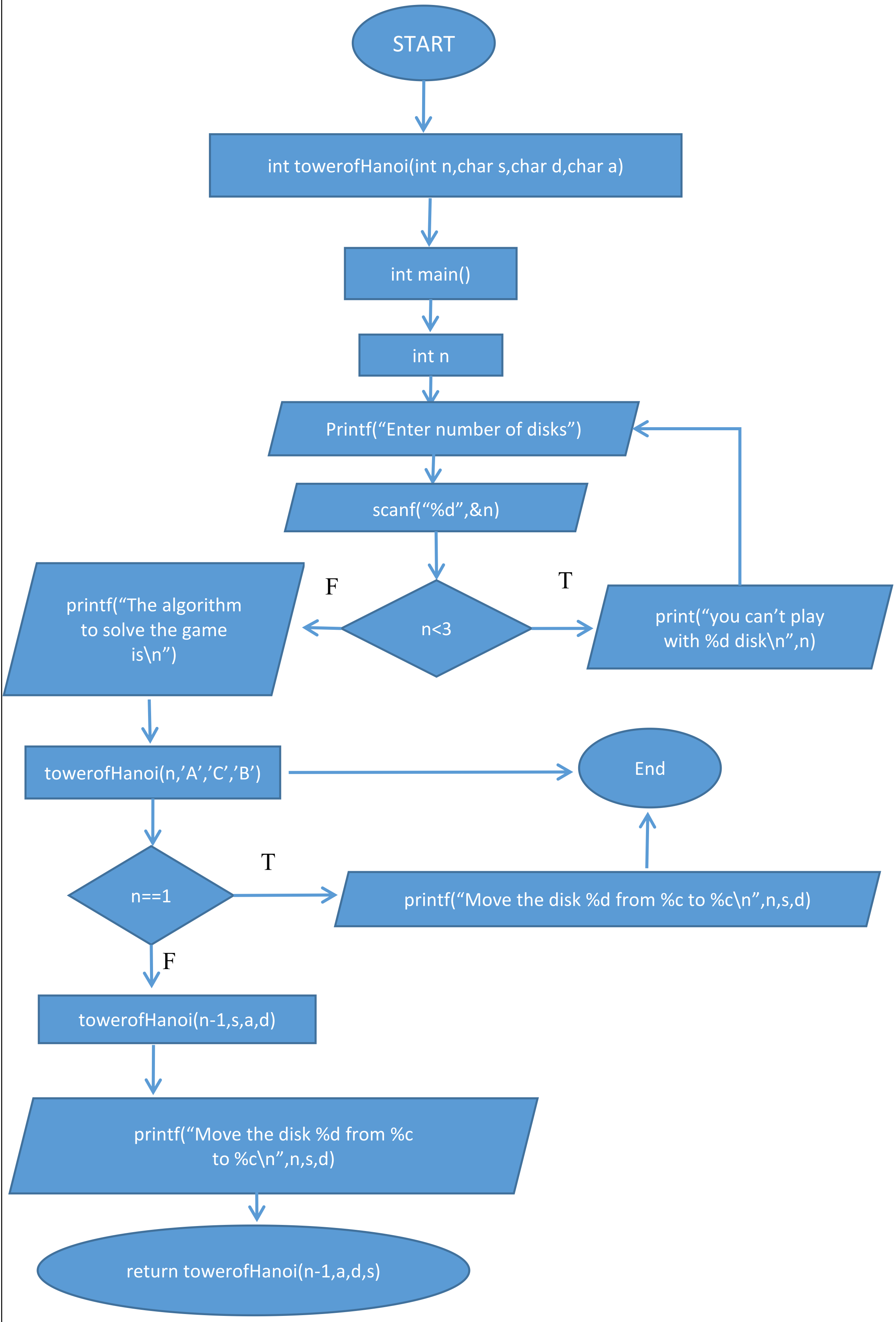
Step 11: Call the function ‘towerofHanoi’ recursively with arguments ‘n-1’,’s’,’a’,’d’

Step 12: Print a message to move the disk ‘n’ from ‘s’ to ‘d’.

Step 13: Call the function ‘towerofHanoi’ recursively with arguments with arguments ‘n-1’,’a’,’d’,’s’

Step 14: End the function

1.3.2 Flowchart


START
int towerofHanoi(int n,char s,char d,char a)
int main()
int n
Printf("Enter number of disks")
scanf("%d",&n)
n<3
F
T
printf("The algorithm to solve the game is\n")
print("you can't play with %d disk\n",n)
towerofHanoi(n,'A','C','B')
End
n==1
T
F
printf("Move the disk %d from %c to %c\n",n,s,d)
towerofHanoi(n-1,s,a,d)
printf("Move the disk %d from %c to %c\n",n,s,d)
return towerofHanoi(n-1,a,d,s)

1.4 Conclusion

For decades, mathematicians and computer scientists have been attracted by the Tower of Hanoi. It entails shifting a stack of disks of varying sizes from one peg to another while according to certain criteria.

The Tower of Hanoi can be a wonderful project for learning about algorithm design, recursion, and data structures. It can also be a fun and difficult problem to solve.

Reflecting on the problem-solving process and the approaches employed to build a solution could be one possible conclusion to a Tower of Hanoi project. This could include debating the trade-offs between different algorithms, examining the solution's performance, and considering alternate alternatives.

Another possibility is to expand the project by adding more features.For example, you could investigate ways to modify the Tower of Hanoi to allow more pegs, or you could design a graphical user interface to make the challenge more interactive.

Overall, anyone interested in programming and algorithm design will find the Tower of Hanoi to be a satisfying project. Its rich history and complex gameplay are likely to bring hours of entertainment and intellectual stimulation.

3.2 Project C Code

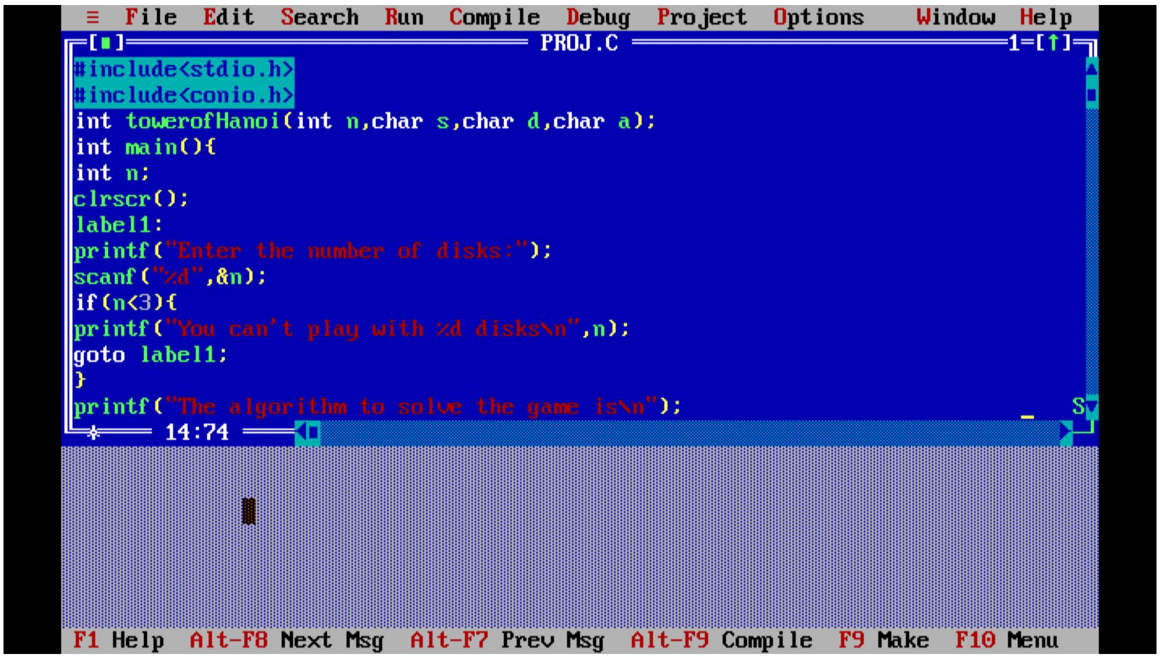
```
#include <stdio.h>

#include<conio.h>

int towerofHanoi(int n, char s, char d, char a);

int main() {

   int n;

clrscr();

   label_1:

   printf("Enter the number of disks: ");

   scanf ("%d", &n);

   if(n<3){

      printf("You cant play with %d disks \n",n);

      goto label_1;

   }

   printf("The algorithm to solve the game is\n");

   towerofHanoi(n, 'A', 'C', ‘B');

getch();

   return 0;

}

int towerofHanoi(int n, char s, char d, char a) {

   if (n == 1) {

      printf("Move the disk 1 from %c to %c\n", s, d);
```

```
        return 0;

    }

    towerofHanoi(n-1, s, a, d);

    printf("Move the disk %d from %c to %c\n", n, s, d);

    return towerofHanoi(n-1, a, d, s);

}
```

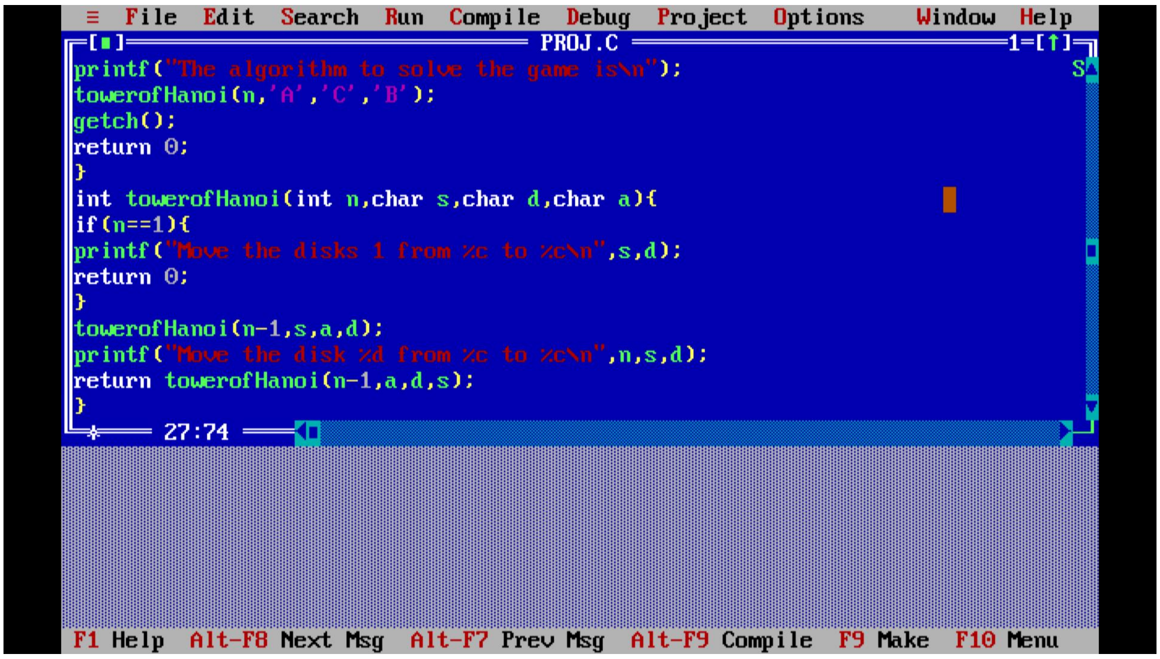
3. Appendices

3.1 Screenshots

3.1.1Code Screenshot

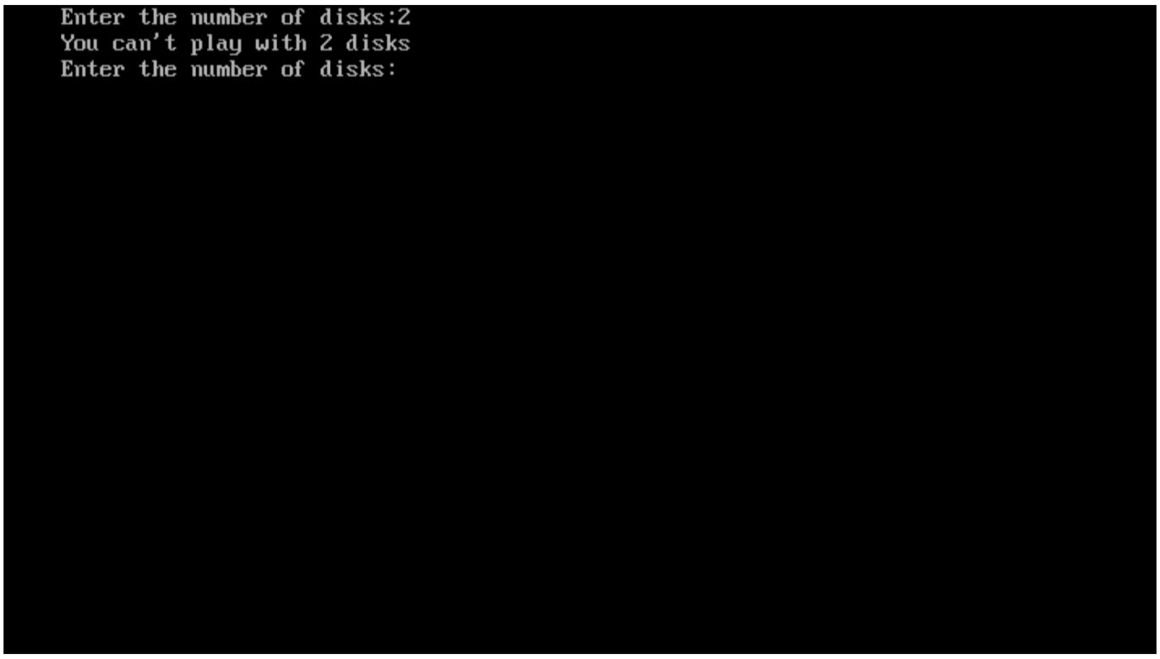

≡ File Edit Search Run Compile Debug Project Options Window Help
PROJ.C
1=[↑]
14:74
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

```
#include<stdio.h>
#include<conio.h>
int towerofHanoi(int n,char s,char d,char a);
int main(){
int n;
clrscr();
label1:
printf("Enter the number of disks:");
scanf("%d",&n);
if(n<3){
printf("You can't play with %d disks\n",n);
goto label1;
}
printf("The algorithm to solve the game is\n");
```

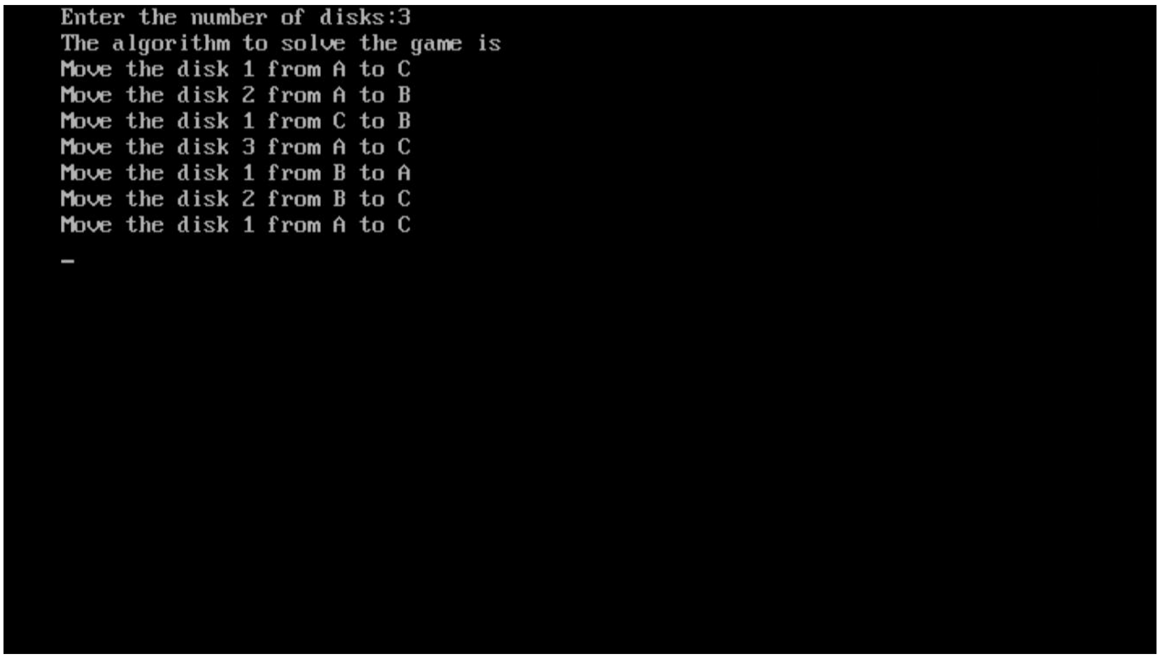

≡ File Edit Search Run Compile Debug Project Options Window Help
PROJ.C
1=[↑]
27:74
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

```
printf("The algorithm to solve the game is\n");
towerofHanoi(n,'A','C','B');
getch();
return 0;
}
int towerofHanoi(int n,char s,char d,char a){
if(n==1){
printf("Move the disks 1 from %c to %c\n",s,d);
return 0;
}
towerofHanoi(n-1,s,a,d);
printf("Move the disk %d from %c to %c\n",n,s,d);
return towerofHanoi(n-1,a,d,s);
}
```

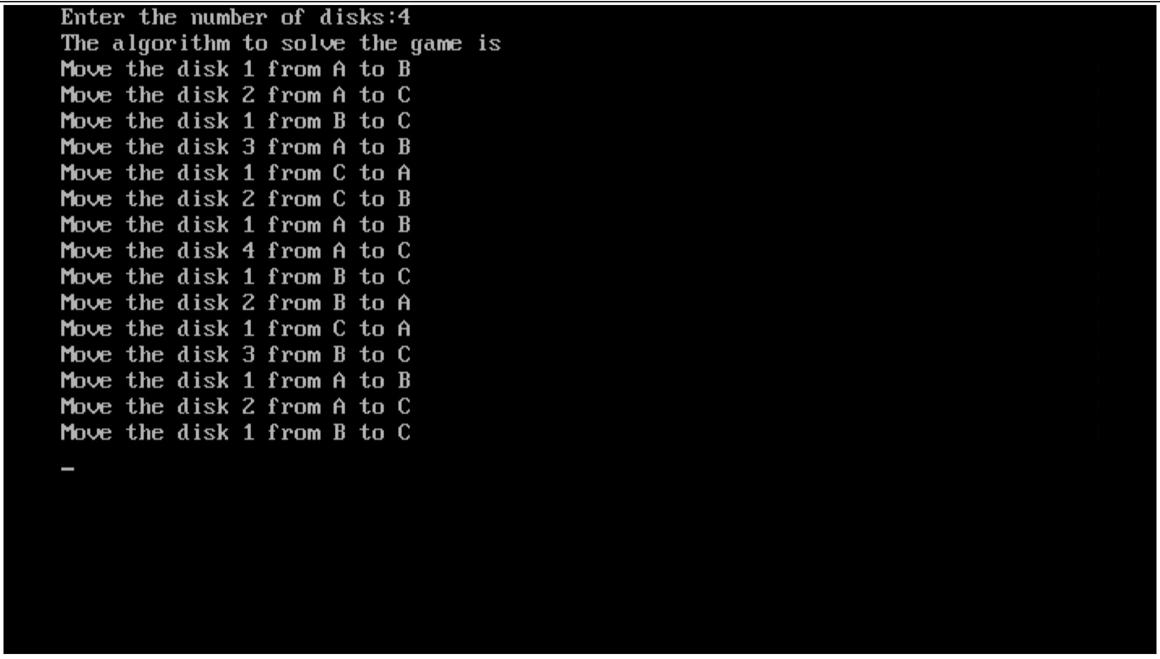
3.1.2 Code Output:



```
Enter the number of disks:2
You can't play with 2 disks
Enter the number of disks:
```



```
Enter the number of disks:3
The algorithm to solve the game is
Move the disk 1 from A to C
Move the disk 2 from A to B
Move the disk 1 from C to B
Move the disk 3 from A to C
Move the disk 1 from B to A
Move the disk 2 from B to C
Move the disk 1 from A to C

_
```



```
Enter the number of disks:4
The algorithm to solve the game is
Move the disk 1 from A to B
Move the disk 2 from A to C
Move the disk 1 from B to C
Move the disk 3 from A to B
Move the disk 1 from C to A
Move the disk 2 from C to B
Move the disk 1 from A to B
Move the disk 4 from A to C
Move the disk 1 from B to C
Move the disk 2 from B to A
Move the disk 1 from C to A
Move the disk 3 from B to C
Move the disk 1 from A to B
Move the disk 2 from A to C
Move the disk 1 from B to C
_
```