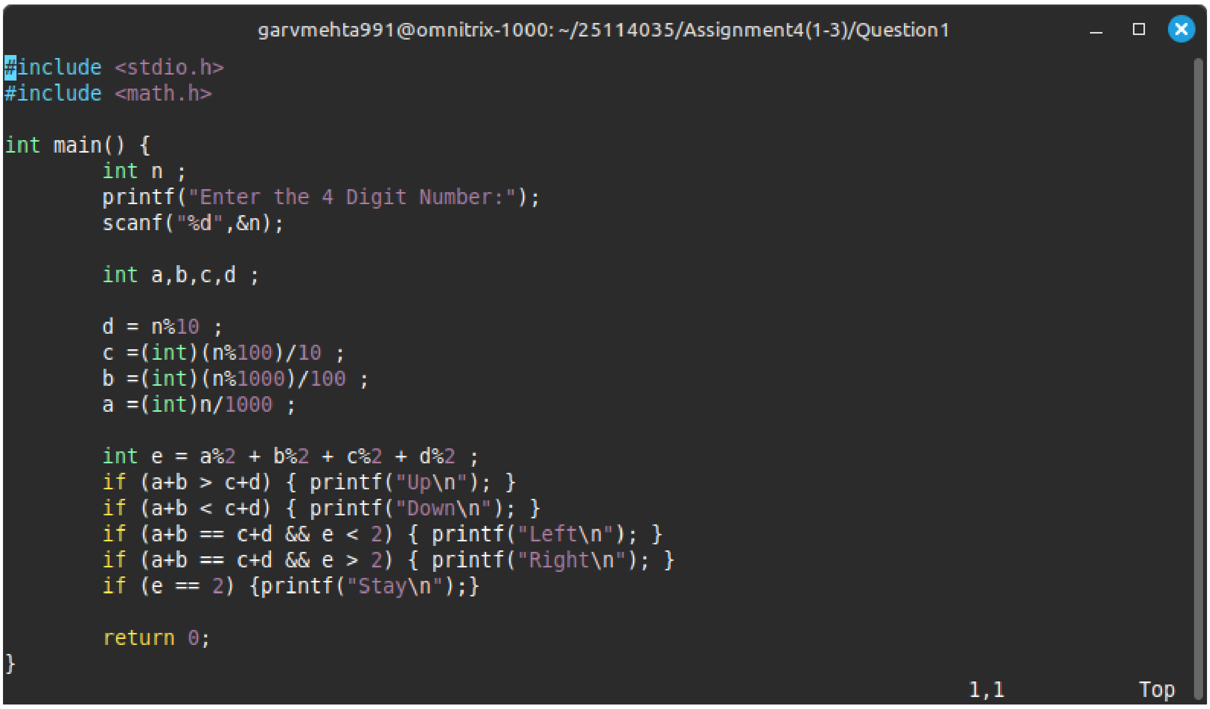
	INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
Programming in C and C++ (CSC-101)	

Assignment 4 40 marks

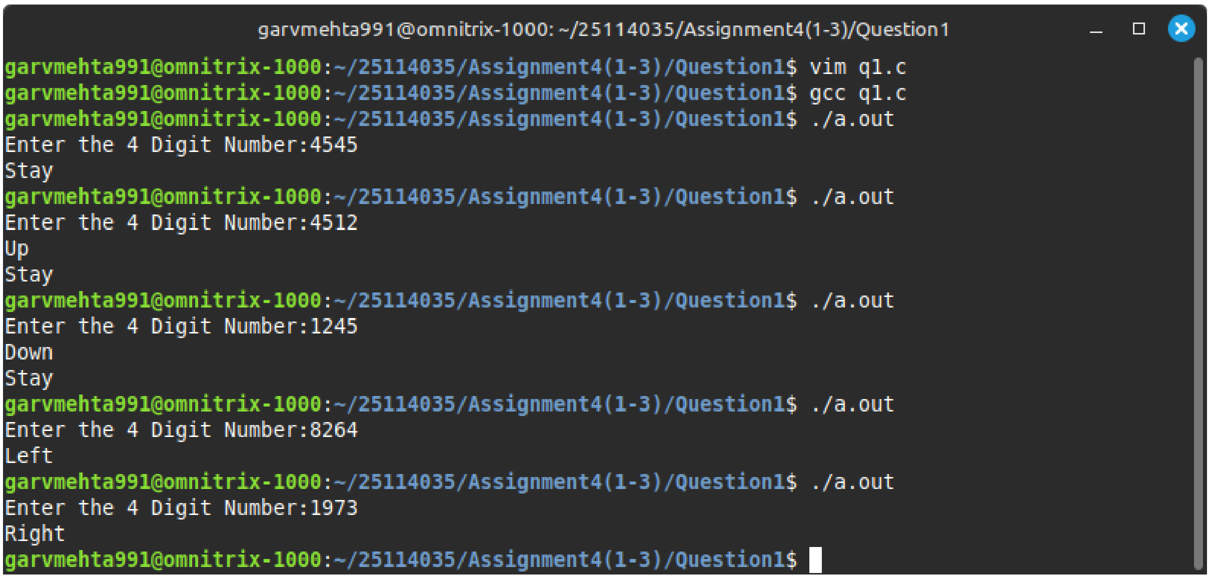
Note : (No loops/ switch condition are allowed to be used)

Q-1) Given a 4-digit number, write a program that prints: 5 marks

1. "Up" : if the sum of the first two digits is greater than the sum of the last two digits
2. "Down" : if the sum of the last two digits is greater
3. "Left" : if both sums are equal but the number has more even digits
4. "Right" : if both sums are equal but the number has more odd digits
5. "Stay" : if the count of even and odd digits are equal
6. If the input number satisfies multiple criteria, it prints all applicable results

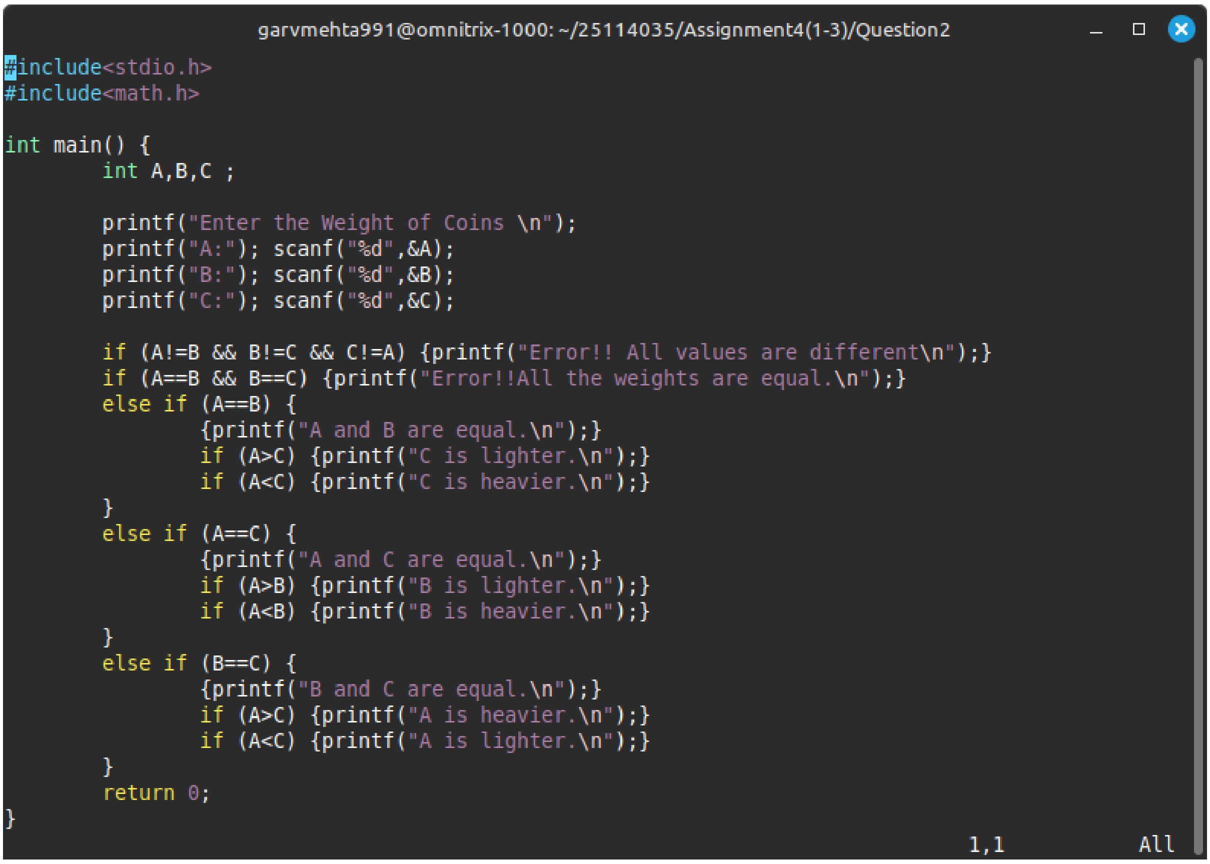

garvmehta991@omnitrix-1000: ~/25114035/Assignment4(1-3)/Question1

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

int main() {
        int n ;
        printf("Enter the 4 Digit Number:");
        scanf("%d",&n);

        int a,b,c,d ;

        d = n%10 ;
        c =(int)(n%100)/10 ;
        b =(int)(n%1000)/100 ;
        a =(int)n/1000 ;

        int e = a%2 + b%2 + c%2 + d%2 ;
        if (a+b > c+d) { printf("Up\n"); }
        if (a+b < c+d) { printf("Down\n"); }
        if (a+b == c+d && e < 2) { printf("Left\n"); }
        if (a+b == c+d && e > 2) { printf("Right\n"); }
        if (e == 2) {printf("Stay\n");}

        return 0;
}
                                                                  1,1           Top
```

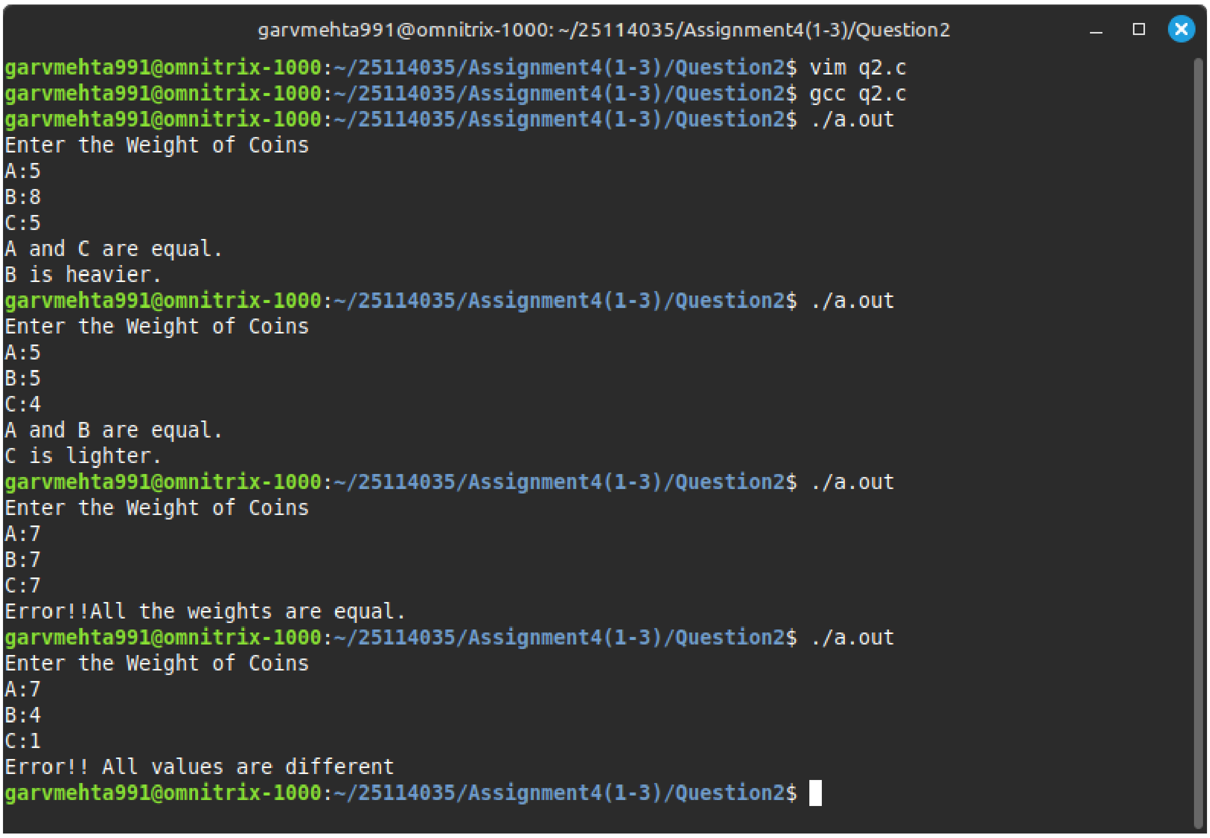

garvmehta991@omnitrix-1000: ~/25114035/Assignment4(1-3)/Question1

```
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question1$ vim q1.c
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question1$ gcc q1.c
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question1$ ./a.out
Enter the 4 Digit Number:4545
Stay
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question1$ ./a.out
Enter the 4 Digit Number:4512
Up
Stay
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question1$ ./a.out
Enter the 4 Digit Number:1245
Down
Stay
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question1$ ./a.out
Enter the 4 Digit Number:8264
Left
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question1$ ./a.out
Enter the 4 Digit Number:1973
Right
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question1$ 
```

Q-2) Given three integer weights A, B, and C representing the weights of coins, one of which is fake (lighter or heavier), and two of them are equal. Write a program to print the fake coin's position (1, 2, or 3) and print "Lighter" or "Heavier" depending on whether it's less than or more than the others. Also, handle the error condition to check the user gives at least two equal values as input. 10 marks


garvmehta991@omnitrix-1000: ~/25114035/Assignment4(1-3)/Question2

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

int main() {
        int A,B,C ;

        printf("Enter the Weight of Coins \n");
        printf("A:"); scanf("%d",&A);
        printf("B:"); scanf("%d",&B);
        printf("C:"); scanf("%d",&C);

        if (A!=B && B!=C && C!=A) {printf("Error!! All values are different\n");}
        if (A==B && B==C) {printf("Error!!All the weights are equal.\n");}
        else if (A==B) {
                {printf("A and B are equal.\n");}
                if (A>C) {printf("C is lighter.\n");}
                if (A<C) {printf("C is heavier.\n");}
        }
        else if (A==C) {
                {printf("A and C are equal.\n");}
                if (A>B) {printf("B is lighter.\n");}
                if (A<B) {printf("B is heavier.\n");}
        }
        else if (B==C) {
                {printf("B and C are equal.\n");}
                if (A>C) {printf("A is heavier.\n");}
                if (A<C) {printf("A is lighter.\n");}
        }
        return 0;
}
                                                                                               1,1           All
```


garvmehta991@omnitrix-1000: ~/25114035/Assignment4(1-3)/Question2

```
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question2$ vim q2.c
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question2$ gcc q2.c
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question2$ ./a.out
Enter the Weight of Coins
A:5
B:8
C:5
A and C are equal.
B is heavier.
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question2$ ./a.out
Enter the Weight of Coins
A:5
B:5
C:4
A and B are equal.
C is lighter.
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question2$ ./a.out
Enter the Weight of Coins
A:7
B:7
C:7
Error!!All the weights are equal.
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question2$ ./a.out
Enter the Weight of Coins
A:7
B:4
C:1
Error!! All values are different
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question2$ 
```

Q-3) Take a 3-digit number as input from the user: 10 marks

1. Print "Armstrong" if the number equals the sum of cubes of its digits.
2. Print "Palindrome" if it reads the same forward and backward.
3. Print "Mirror" if reversing the digits gives the same number when each digit is replaced with its mirror image:

   Mirror pairs: 0↔0, 1↔1, 6↔9, 9↔6, 8↔8

4. Print "None" if it satisfies none of the above.
5. If the input number satisfies multiple criteria, it prints all applicable results

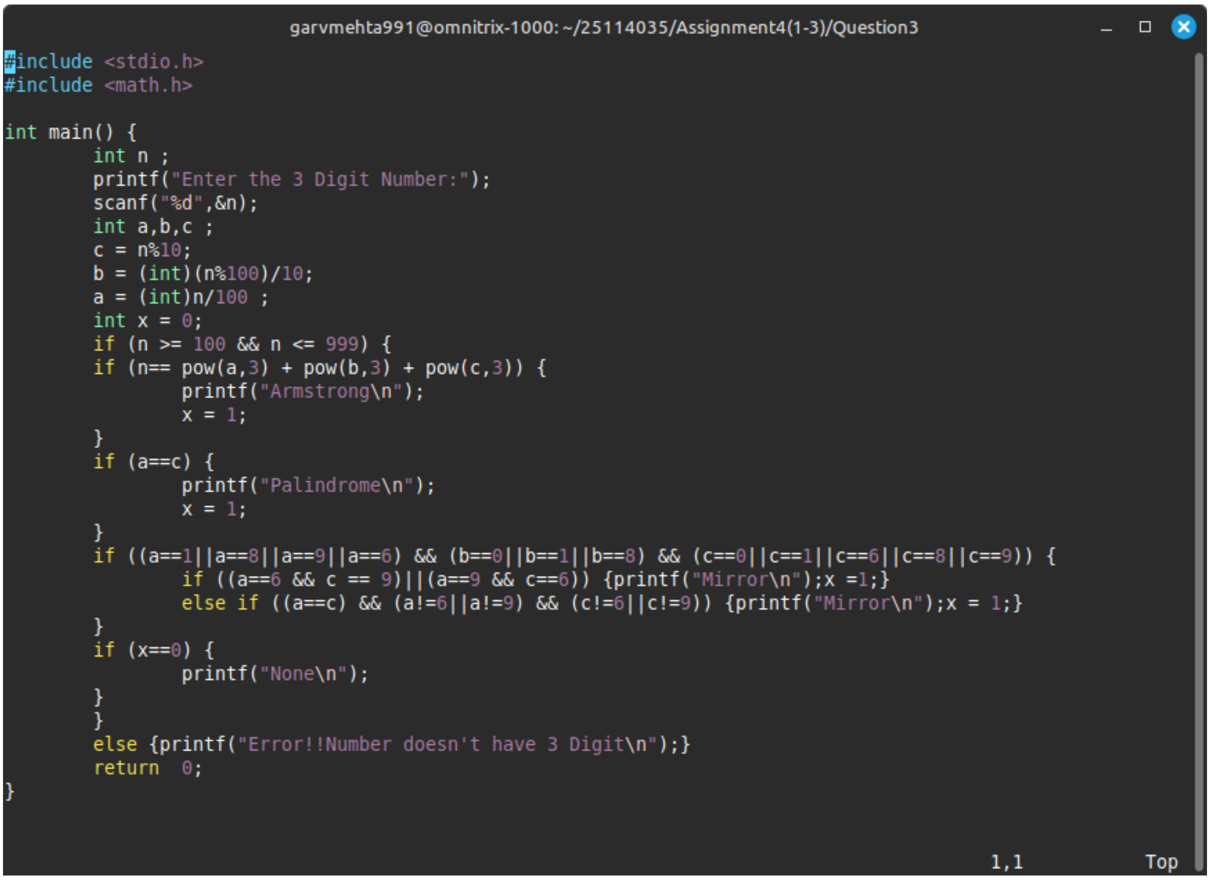
Example : Suppose the number is 619. Reverse the digits: 916. Now replace each digit with its mirror pair:

9 → 6

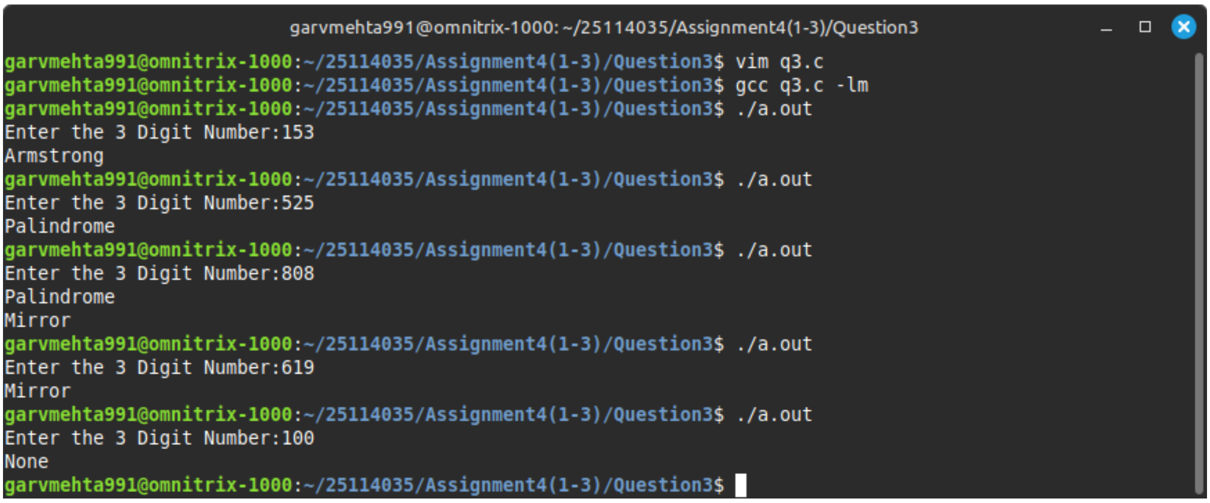
1 → 1

6 → 9

So, the mirrored reversed number is 619. Since the mirrored reversed number 619 is the same as the original number 619, print "Mirror".


garvmehta991@omnitrix-1000: ~/25114035/Assignment4(1-3)/Question3

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

int main() {
        int n ;
        printf("Enter the 3 Digit Number:");
        scanf("%d",&n);
        int a,b,c ;
        c = n%10;
        b = (int)(n%100)/10;
        a = (int)n/100 ;
        int x = 0;
        if (n >= 100 && n <= 999) {
        if (n== pow(a,3) + pow(b,3) + pow(c,3)) {
                printf("Armstrong\n");
                x = 1;
        }
        if (a==c) {
                printf("Palindrome\n");
                x = 1;
        }
        if ((a==1||a==8||a==9||a==6) && (b==0||b==1||b==8) && (c==0||c==1||c==6||c==8||c==9)) {
                if ((a==6 && c == 9)||(a==9 && c==6)) {printf("Mirror\n");x =1;}
                else if ((a==c) && (a!=6||a!=9) && (c!=6||c!=9)) {printf("Mirror\n");x = 1;}
        }
        if (x==0) {
                printf("None\n");
        }
        }
        else {printf("Error!!Number doesn't have 3 Digit\n");}
        return  0;
}
```



garvmehta991@omnitrix-1000: ~/25114035/Assignment4(1-3)/Question3

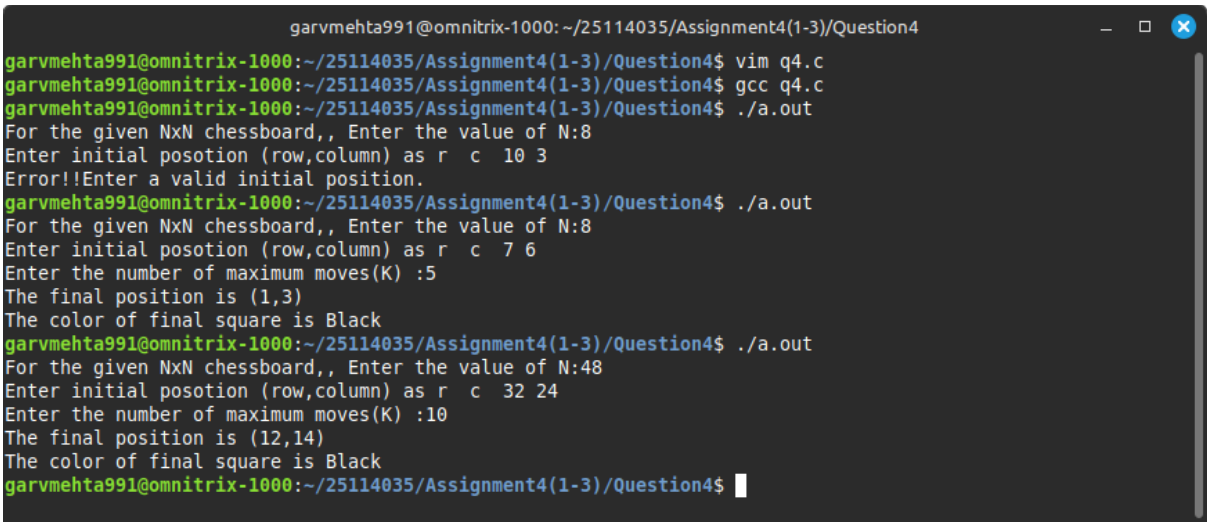
```
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question3$ vim q3.c
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question3$ gcc q3.c -lm
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question3$ ./a.out
Enter the 3 Digit Number:153
Armstrong
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question3$ ./a.out
Enter the 3 Digit Number:525
Palindrome
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question3$ ./a.out
Enter the 3 Digit Number:808
Palindrome
Mirror
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question3$ ./a.out
Enter the 3 Digit Number:619
Mirror
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question3$ ./a.out
Enter the 3 Digit Number:100
None
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question3$ 
```

Q-4) You are given an N×N chessboard and an initial position (row, column) of a knight. The knight can only move in one restricted way: two steps up and one step left. Given an integer K representing the number of moves, determine the knight's final position after making exactly K valid moves. If at any move the knight cannot move without going off the board, it must stop moving. Also, determine the color of the final square assuming the top-left corner (1,1) is black and colors alternate like a standard chessboard. Also check if the user enters a valid initial position, otherwise print error. 15 marks

Input: Board size N, starting position (r, c), number of moves K **(assume K ≤ 10)**.

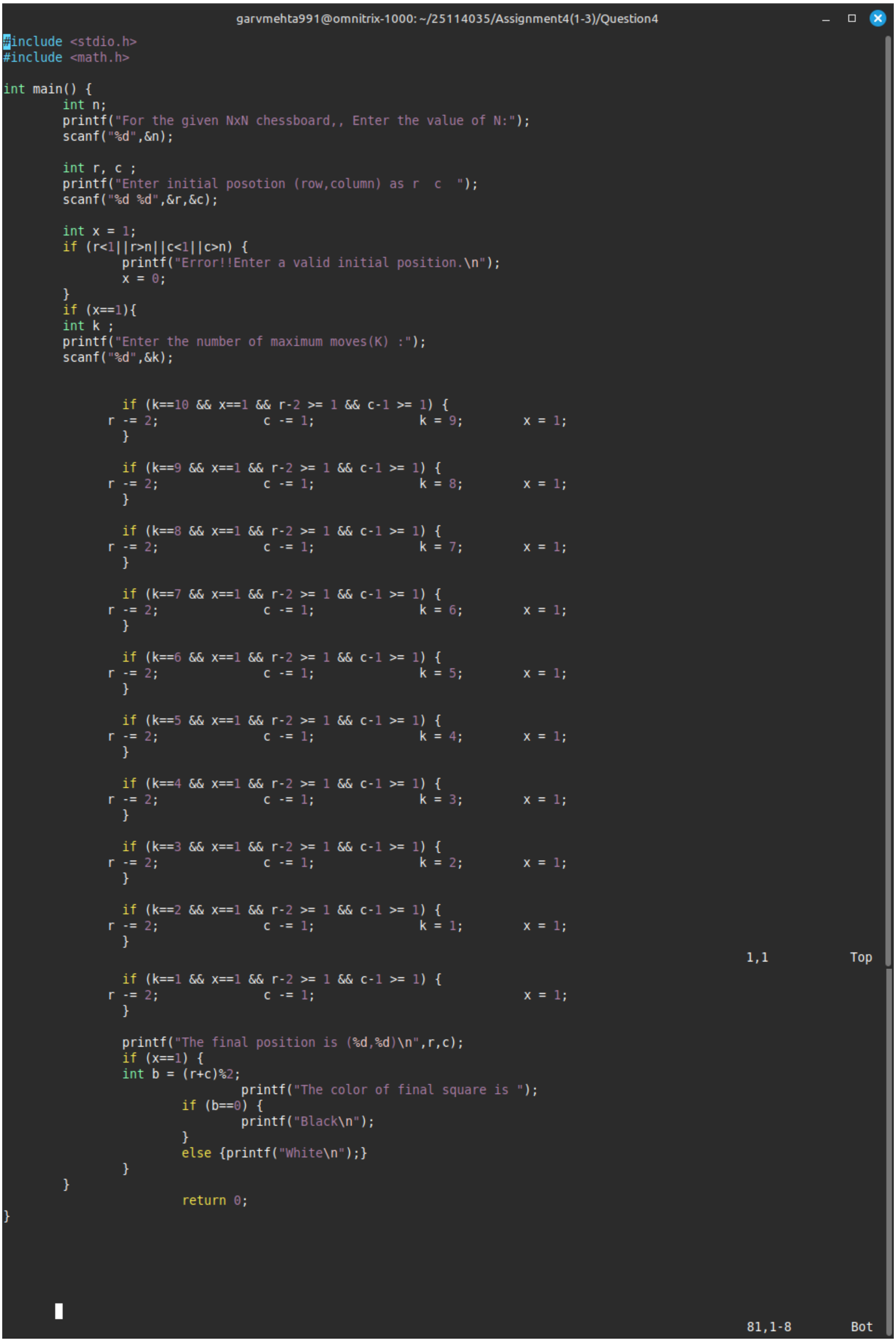
Output: Final position and the color of the square.

Constraint: Implement the entire logic using **only nested if-else statements without any loops or functions.**



garvmehta991@omnitrix-1000: ~/25114035/Assignment4(1-3)/Question4

```
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question4$ vim q4.c
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question4$ gcc q4.c
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question4$ ./a.out
For the given NxN chessboard,, Enter the value of N:8
Enter initial posotion (row,column) as r  c  10 3
Error!!Enter a valid initial position.
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question4$ ./a.out
For the given NxN chessboard,, Enter the value of N:8
Enter initial posotion (row,column) as r  c  7 6
Enter the number of maximum moves(K) :5
The final position is (1,3)
The color of final square is Black
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question4$ ./a.out
For the given NxN chessboard,, Enter the value of N:48
Enter initial posotion (row,column) as r  c  32 24
Enter the number of maximum moves(K) :10
The final position is (12,14)
The color of final square is Black
garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question4$ 
```


garvmehta991@omnitrix-1000:~/25114035/Assignment4(1-3)/Question4
1,1 Top
81,1-8 Bot

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

int main() {
        int n;
        printf("For the given NxN chessboard,, Enter the value of N:");
        scanf("%d",&n);

        int r, c ;
        printf("Enter initial posotion (row,column) as r  c  ");
        scanf("%d %d",&r,&c);

        int x = 1;
        if (r<1||r>n||c<1||c>n) {
                printf("Error!!Enter a valid initial position.\n");
                x = 0;
        }
        if (x==1){
        int k ;
        printf("Enter the number of maximum moves(K) :");
        scanf("%d",&k);


                if (k==10 && x==1 && r-2 >= 1 && c-1 >= 1) {
              r -= 2;              c -= 1;              k = 9;        x = 1;
                }

                if (k==9 && x==1 && r-2 >= 1 && c-1 >= 1) {
              r -= 2;              c -= 1;              k = 8;        x = 1;
                }

                if (k==8 && x==1 && r-2 >= 1 && c-1 >= 1) {
              r -= 2;              c -= 1;              k = 7;        x = 1;
                }

                if (k==7 && x==1 && r-2 >= 1 && c-1 >= 1) {
              r -= 2;              c -= 1;              k = 6;        x = 1;
                }

                if (k==6 && x==1 && r-2 >= 1 && c-1 >= 1) {
              r -= 2;              c -= 1;              k = 5;        x = 1;
                }

                if (k==5 && x==1 && r-2 >= 1 && c-1 >= 1) {
              r -= 2;              c -= 1;              k = 4;        x = 1;
                }

                if (k==4 && x==1 && r-2 >= 1 && c-1 >= 1) {
              r -= 2;              c -= 1;              k = 3;        x = 1;
                }

                if (k==3 && x==1 && r-2 >= 1 && c-1 >= 1) {
              r -= 2;              c -= 1;              k = 2;        x = 1;
                }

                if (k==2 && x==1 && r-2 >= 1 && c-1 >= 1) {
              r -= 2;              c -= 1;              k = 1;        x = 1;
                }

                if (k==1 && x==1 && r-2 >= 1 && c-1 >= 1) {
              r -= 2;              c -= 1;                            x = 1;
                }

                printf("The final position is (%d,%d)\n",r,c);
                if (x==1) {
                int b = (r+c)%2;
                                printf("The color of final square is ");
                        if (b==0) {
                                printf("Black\n");
                        }
                        else {printf("White\n");}
                }
        }
                        return 0;
}
```