

```
    head = temp->next;
    free(temp);
    return;
}
while (temp != NULL && temp->data != x) {
    prev = temp;
    temp = temp->next;
}
if (temp == NULL) {
    printf("Value not found!\n");
    return;
}
prev->next = temp->next;
free(temp);
}
void reverseDisplay(struct Node *node) {
    if (node == NULL) return;
    reverseDisplay(node->next);
    printf("%d -> ", node->data);
}
void reverseList() {
    struct Node *prev = NULL, *curr = head, *next = NULL;
    while (curr != NULL) {
        next = curr->next;
        curr->next = prev;
        prev = curr;
        curr = next;
    }
    head = prev;
}
```

```
void search(int x) {
    struct Node *temp = head;
    int pos = 1;
    while (temp != NULL) {
        if (temp->data == x) {
            printf("Found at position %d\n", pos);
            return;
        }
        temp = temp->next;
        pos++;
    }
    printf("Not found!\n");
}

void selectionSort() {
    struct Node *i, *j;
    for (i = head; i != NULL; i = i->next) {
        for (j = i->next; j != NULL; j = j->next) {
            if (i->data > j->data) {
                int temp = i->data;
                i->data = j->data;
                j->data = temp;
            }
        }
    }
}

int main() {
    int choice, val, pos;
    while (1) {
        printf("\n----- MENU ----- \n");
        printf("1. Insert at Beginning\n2. Insert at End\n3. Insert at Position\n");
```

```
printf("4. Display\n5. Delete\n6. Reverse Display\n7. Reverse List\n");
printf("8. Search\n9. Sort\n10. Exit\n");
printf("Enter choice: ");
scanf("%d", &choice);
switch (choice) {
    case 1: printf("Enter value: "); scanf("%d", &val); insertBeg(val); break;
    case 2: printf("Enter value: "); scanf("%d", &val); insertEnd(val); break;
    case 3: printf("Enter pos & value: "); scanf("%d%d", &pos, &val); insertPos(pos, val);
break;
    case 4: display(); break;
    case 5: printf("Enter value to delete: "); scanf("%d", &val); deleteValue(val); break;
    case 6: reverseDisplay(head); printf("NULL\n"); break;
    case 7: reverseList(); break;
    case 8: printf("Enter value: "); scanf("%d", &val); search(val); break;
    case 9: selectionSort(); break;
    case 10: exit(0);
    default: printf("Invalid choice!\n");
}
}
}
```

## Output :-

```
C:\Users\Sanat\OneDrive\Doc  X  +  v

----- MENU -----
1. Insert at Beginning
2. Insert at End
3. Insert at Position
4. Display
5. Delete
6. Reverse Display
7. Reverse List
8. Search
9. Sort
10. Exit
Enter choice: 1
Enter value: 10

----- MENU -----
1. Insert at Beginning
2. Insert at End
3. Insert at Position
4. Display
5. Delete
6. Reverse Display
7. Reverse List
8. Search
9. Sort
10. Exit
Enter choice: 2
Enter value: 20

----- MENU -----
1. Insert at Beginning
2. Insert at End
3. Insert at Position
4. Display
5. Delete
6. Reverse Display
7. Reverse List
8. Search
9. Sort
10. Exit
Enter choice: 2
Enter value: 30

C:\Users\Sanat\OneDrive\Doc  X  +  v

----- MENU -----
1. Insert at Beginning
2. Insert at End
3. Insert at Position
4. Display
5. Delete
6. Reverse Display
7. Reverse List
8. Search
9. Sort
10. Exit
Enter choice: 4
10 -> 20 -> 30 -> NULL

----- MENU -----
1. Insert at Beginning
2. Insert at End
3. Insert at Position
4. Display
5. Delete
6. Reverse Display
7. Reverse List
8. Search
9. Sort
10. Exit
Enter choice: 3
Enter pos & value: 2 15

----- MENU -----
1. Insert at Beginning
2. Insert at End
3. Insert at Position
4. Display
5. Delete
6. Reverse Display
7. Reverse List
8. Search
9. Sort
10. Exit
Enter choice: 4
10 -> 15 -> 20 -> 30 -> NULL
```

```
C:\Users\Sanat\OneDrive\Doc × + v

----- MENU -----
1. Insert at Beginning
2. Insert at End
3. Insert at Position
4. Display
5. Delete
6. Reverse Display
7. Reverse List
8. Search
9. Sort
10. Exit
Enter choice: 5
Enter value to delete: 20

----- MENU -----
1. Insert at Beginning
2. Insert at End
3. Insert at Position
4. Display
5. Delete
6. Reverse Display
7. Reverse List
8. Search
9. Sort
10. Exit
Enter choice: 4
10 -> 15 -> 30 -> NULL

----- MENU -----
1. Insert at Beginning
2. Insert at End
3. Insert at Position
4. Display
5. Delete
6. Reverse Display
7. Reverse List
8. Search
9. Sort
10. Exit
Enter choice: 6
30 -> 15 -> 10 -> NULL

C:\Users\Sanat\OneDrive\Doc × + v

----- MENU -----
1. Insert at Beginning
2. Insert at End
3. Insert at Position
4. Display
5. Delete
6. Reverse Display
7. Reverse List
8. Search
9. Sort
10. Exit
Enter choice: 7

----- MENU -----
1. Insert at Beginning
2. Insert at End
3. Insert at Position
4. Display
5. Delete
6. Reverse Display
7. Reverse List
8. Search
9. Sort
10. Exit
Enter choice: 4
30 -> 15 -> 10 -> NULL

----- MENU -----
1. Insert at Beginning
2. Insert at End
3. Insert at Position
4. Display
5. Delete
6. Reverse Display
7. Reverse List
8. Search
9. Sort
10. Exit
Enter choice: 8
Enter value: 15
Found at position 2
```

```
C:\Users\Sanat\OneDrive\Doc  ×  +  v

----- MENU -----
1. Insert at Beginning
2. Insert at End
3. Insert at Position
4. Display
5. Delete
6. Reverse Display
7. Reverse List
8. Search
9. Sort
10. Exit
Enter choice: 9

----- MENU -----
1. Insert at Beginning
2. Insert at End
3. Insert at Position
4. Display
5. Delete
6. Reverse Display
7. Reverse List
8. Search
9. Sort
10. Exit
Enter choice: 4
10 -> 15 -> 30 -> NULL

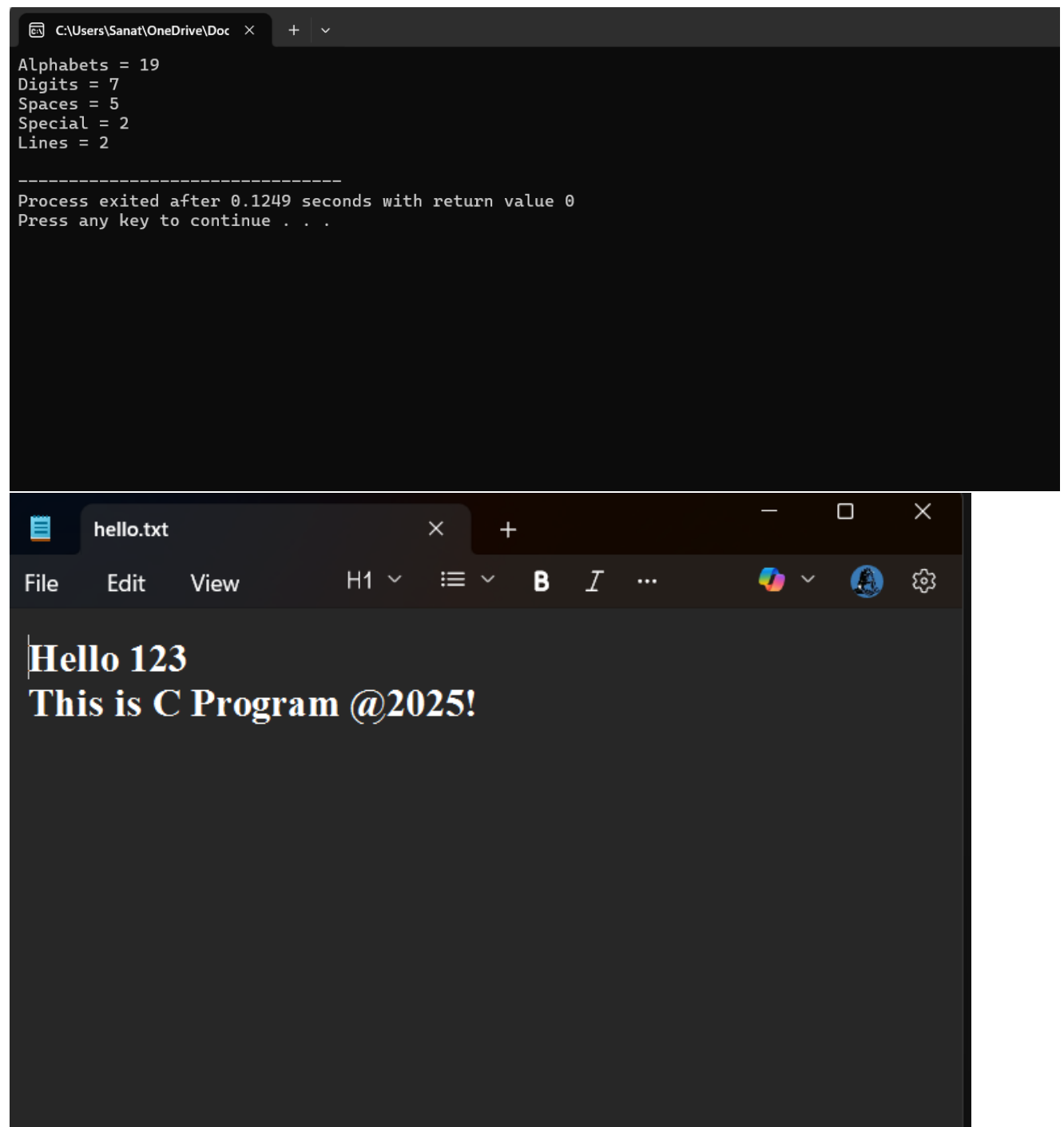
----- MENU -----
1. Insert at Beginning
2. Insert at End
3. Insert at Position
4. Display
5. Delete
6. Reverse Display
7. Reverse List
8. Search
9. Sort
10. Exit
Enter choice: 10

-----
Process exited after 214.4 seconds with return value 0
Press any key to continue . . . |
```

**Ques 11 :-****Code :-** #include <stdio.h>

#include &lt;ctype.h&gt;

```
int main() {  
    FILE *fp;  
    char ch;  
    int alph=0, digit=0, space=0, special=0, lines=1;  
    fp = fopen("C:hello.txt", "w");  
    fprintf(fp, "Hello 123\nThis is C Program @2025!");  
    fclose(fp);  
    fp = fopen("C:hello.txt", "r");  
    while ((ch = fgetc(fp)) != EOF) {  
        if (isalpha(ch)) alph++;  
        else if (isdigit(ch)) digit++;  
        else if (ch == ' ') space++;  
        else if (ch == '\n') lines++;  
        else special++;  
    }  
    fclose(fp);  
    printf("Alphabets = %d\nDigits = %d\nSpaces = %d\nSpecial = %d\nLines = %d\n",  
        alph, digit, space, special, lines);  
    return 0;  
}
```

**Output :-**

The image shows two screenshots. The top screenshot is a terminal window displaying the output of a C program. It lists the counts for various characters: Alphabets = 19, Digits = 7, Spaces = 5, Special = 2, and Lines = 2. It also shows a message indicating the process exited after 0.1249 seconds with a return value of 0, and a prompt to press any key to continue. The bottom screenshot is a text editor window titled 'hello.txt' showing the source code of the program. The code uses the `printf` function to display the character counts and a separator line, followed by a message about the program's execution time and a prompt to press any key to continue.

```
C:\Users\Sanat\OneDrive\Doc X + v
Alphabets = 19
Digits = 7
Spaces = 5
Special = 2
Lines = 2

-----
Process exited after 0.1249 seconds with return value 0
Press any key to continue . . .

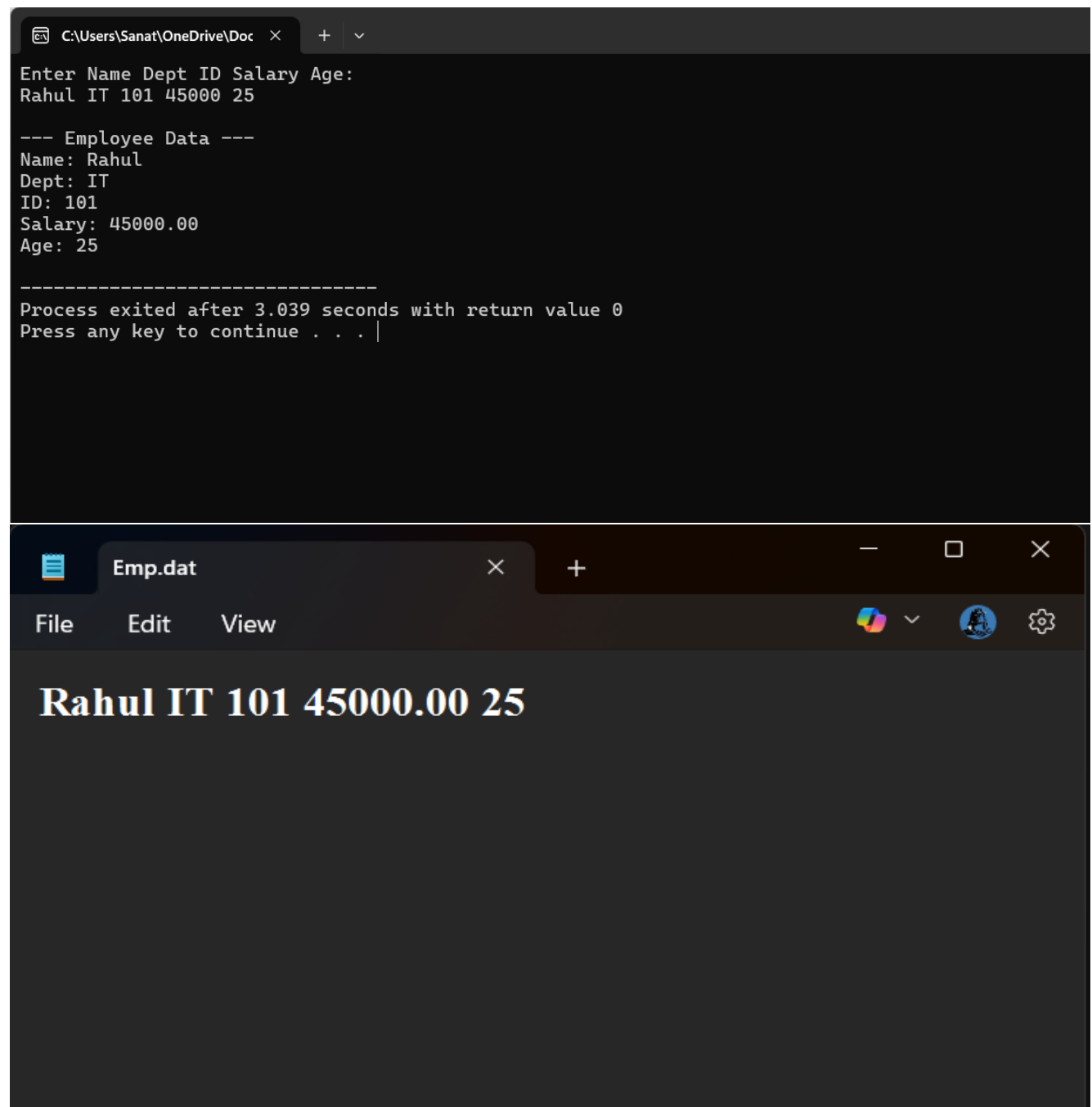
hello.txt X + - □ X
File Edit View H1 v ≡ v B I ...
Hello 123
This is C Program @2025!
```



**Ques 12 :-****Code :-** #include <stdio.h>



```
struct Emp {
    char name[20];
    char dept[20];
    int id;
    float sal;
    int age;
};

int main() {
    FILE *fp;
    struct Emp e;
    fp = fopen("C:Emp.dat", "w");
    printf("Enter Name Dept ID Salary Age:\n");
    scanf("%s %s %d %f %d", e.name, e.dept, &e.id, &e.sal, &e.age);
    fprintf(fp, "%s %s %d %.2f %d", e.name, e.dept, e.id, e.sal, e.age);
    fclose(fp);
    fp = fopen("C:Emp.dat", "r");
    fscanf(fp, "%s %s %d %f %d", e.name, e.dept, &e.id, &e.sal, &e.age);
    printf("\n--- Employee Data ---\n");
    printf("Name: %s\nDept: %s\nID: %d\nSalary: %.2f\nAge: %d\n", e.name, e.dept, e.id,
e.sal, e.age);
    fclose(fp);
    return 0;
}
```

**Output :-**

The image shows two overlapping windows from a Windows operating system. The top window is a Command Prompt with a dark background. It displays the following text: "Enter Name Dept ID Salary Age:", "Rahul IT 101 45000 25", "--- Employee Data ---", "Name: Rahul", "Dept: IT", "ID: 101", "Salary: 45000.00", "Age: 25", a separator line of dashes, and "Process exited after 3.039 seconds with return value 0". The bottom window is a Notepad++ editor with a dark theme, showing a file named "Emp.dat". The menu bar includes "File", "Edit", and "View". The main text area displays "Rahul IT 101 45000.00 25" in a large, bold, yellow font.

```
C:\Users\Sanat\OneDrive\Doc × + ∨  
Enter Name Dept ID Salary Age:  
Rahul IT 101 45000 25  
  
--- Employee Data ---  
Name: Rahul  
Dept: IT  
ID: 101  
Salary: 45000.00  
Age: 25  
  
-----  
Process exited after 3.039 seconds with return value 0  
Press any key to continue . . . |
```

Emp.dat × + - □ ×  
File Edit View  ∨  ⚙️

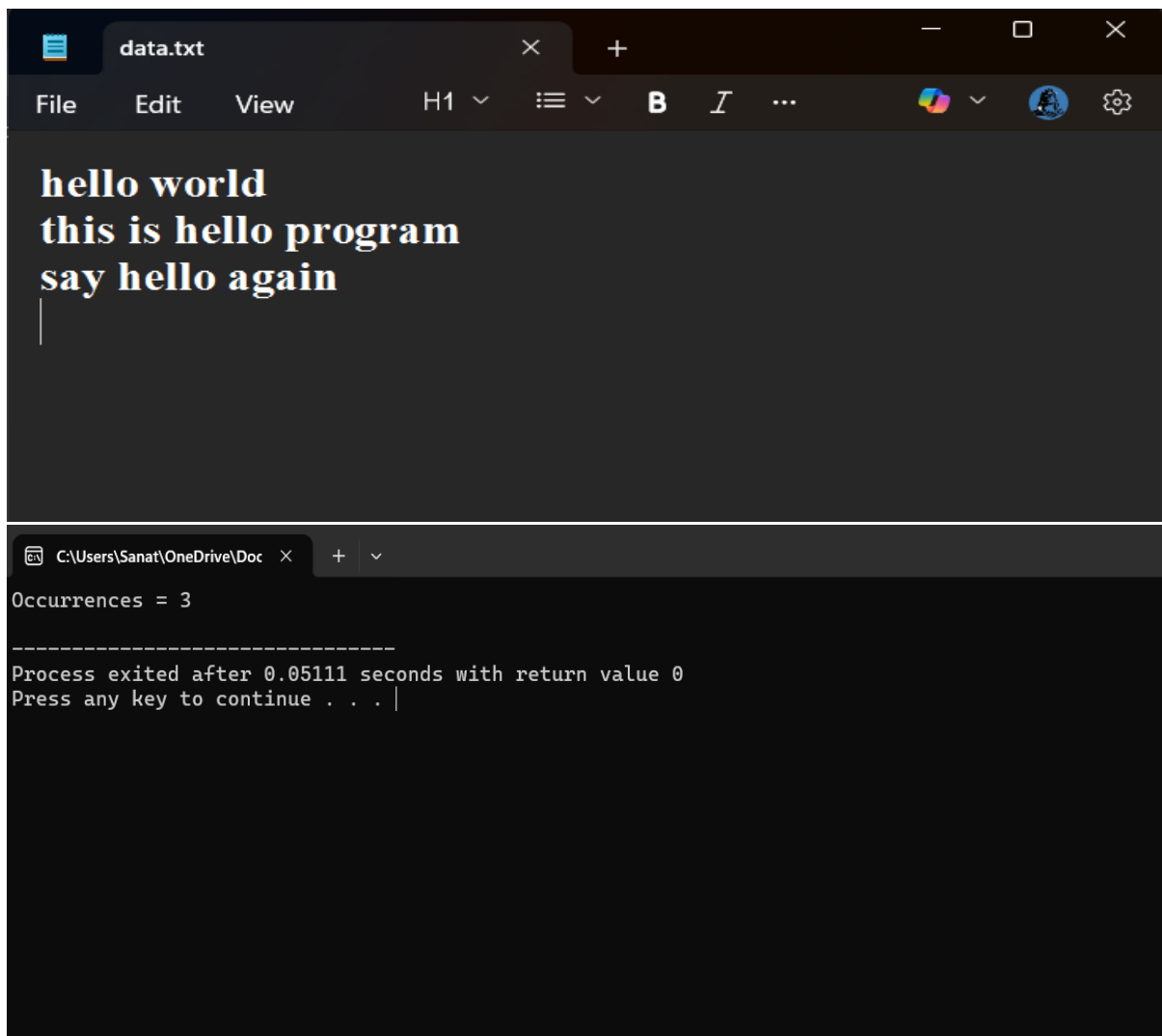
**Rahul IT 101 45000.00 25**

**Ques 13 :-****Code :-** #include <stdio.h>

#include &lt;string.h&gt;

```
int main(int argc, char *argv[]) {
    FILE *fp;
    char str[200], *p;
    int count = 0;
    if (argc != 3) {
        printf("Usage: prog filename substring\n");
        return 0;
    }
    fp = fopen(argv[1], "r");
    if (!fp) {
        printf("File not found!");
        return 0;
    }
    while (fgets(str, 200, fp)) {
        p = strstr(str, argv[2]);
        while (p) {
            count++;
            p = strstr(p + 1, argv[2]);
        }
    }
    printf("Occurrences = %d\n", count);
    fclose(fp);
    return 0;
}
```

Output :-



The image shows two windows. The top window is a code editor titled 'data.txt' with a dark theme. It contains the following text:

```
hello world  
this is hello program  
say hello again
```

The bottom window is a terminal window with a dark theme. It shows the output of a program:

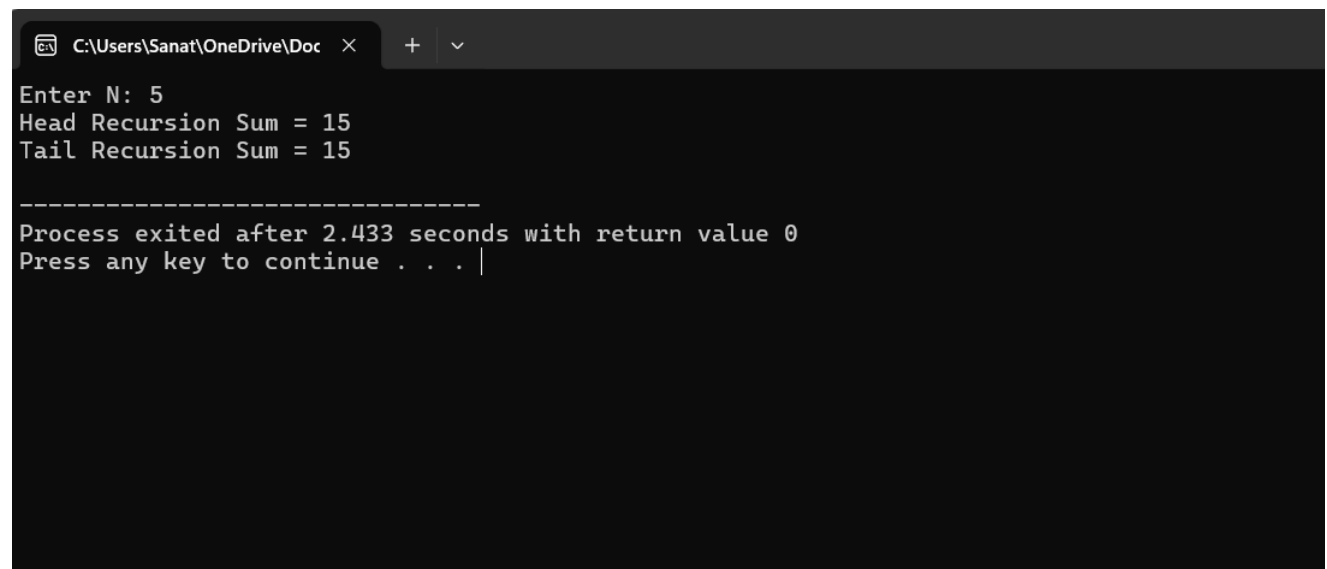
```
Occurrences = 3  
-----  
Process exited after 0.05111 seconds with return value 0  
Press any key to continue . . .
```

**Ques 14 :-****Code :-** #include <stdio.h>

```
int headSum(int n) {
    if (n == 0) return 0;
    return headSum(n - 1) + n;
}

int tailSum(int n, int acc) {
    if (n == 0) return acc;
    return tailSum(n - 1, acc + n);
}

int main() {
    int n;
    printf("Enter N: ");
    scanf("%d", &n);
    printf("Head Recursion Sum = %d\n", headSum(n));
    printf("Tail Recursion Sum = %d\n", tailSum(n, 0));
    return 0;
}
```

**Output :-**

```
C:\Users\Sanat\OneDrive\Doc × + v
Enter N: 5
Head Recursion Sum = 15
Tail Recursion Sum = 15

-----
Process exited after 2.433 seconds with return value 0
Press any key to continue . . . |
```