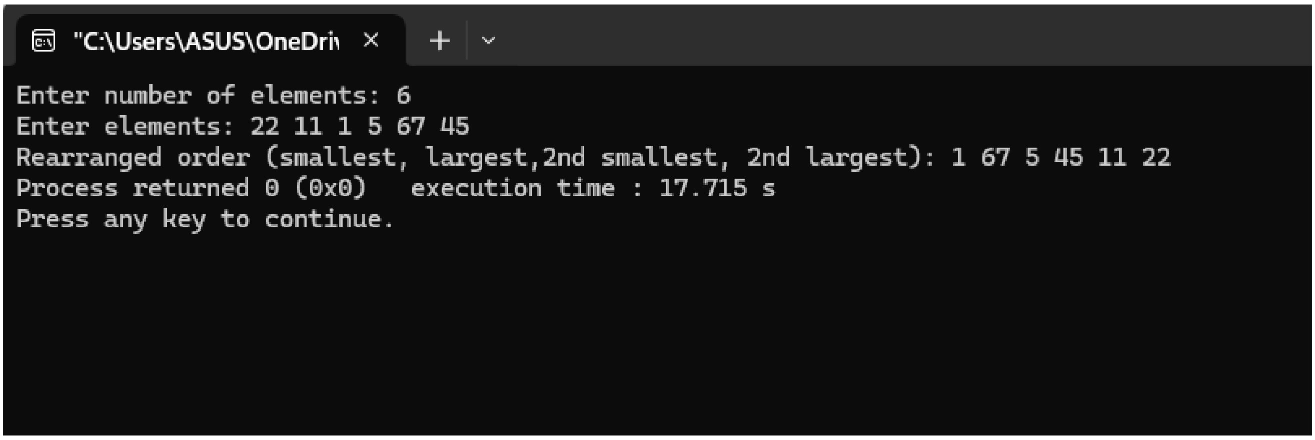
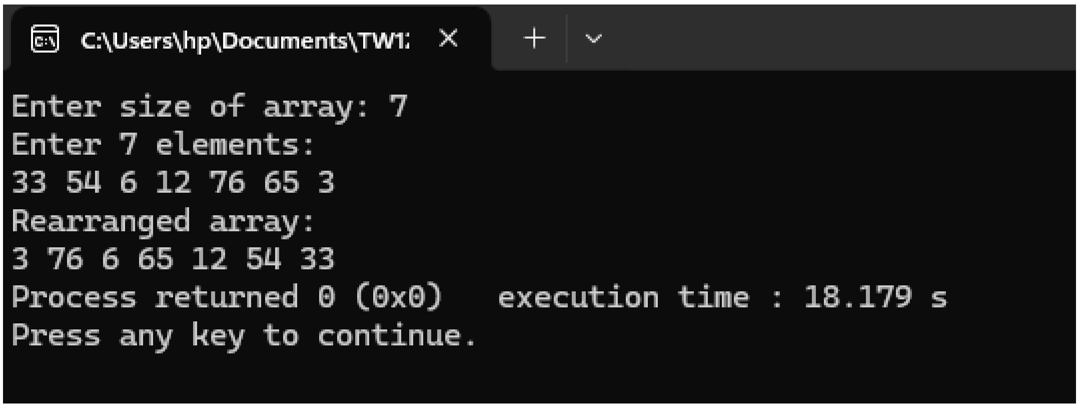
Code:

```
#include <stdio.h>
void sort(int a[], int n)
{
    int i,j,temp;
    for(i=0;i<n;i++)
{
        for(j=i+1;j<n;j++)
{
            if(a[j]<a[i])
{
                temp=a[i];
                a[i]=a[j];
                a[j]=temp;
            }
        }
    }
}

int main()
{
    int n,i,left,right,k=0;
    scanf("%d",&n);
    int a[n],res[n];
    for(i=0;i<n;i++) scanf("%d",&a[i]);
    sort(a,n);
    left=0;
    right=n-1;
    while(left<=right)
{
        res[k++]=a[left++];
        if(left<=right) res[k++]=a[right--];
    }
    for(i=0;i<n;i++) printf("%d ",res[i]);
    return 0;
}
```

Output:



```
Enter number of elements: 6
Enter elements: 22 11 1 5 67 45
Rearranged order (smallest, largest,2nd smallest, 2nd largest): 1 67 5 45 11 22
Process returned 0 (0x0)   execution time : 17.715 s
Press any key to continue.
```

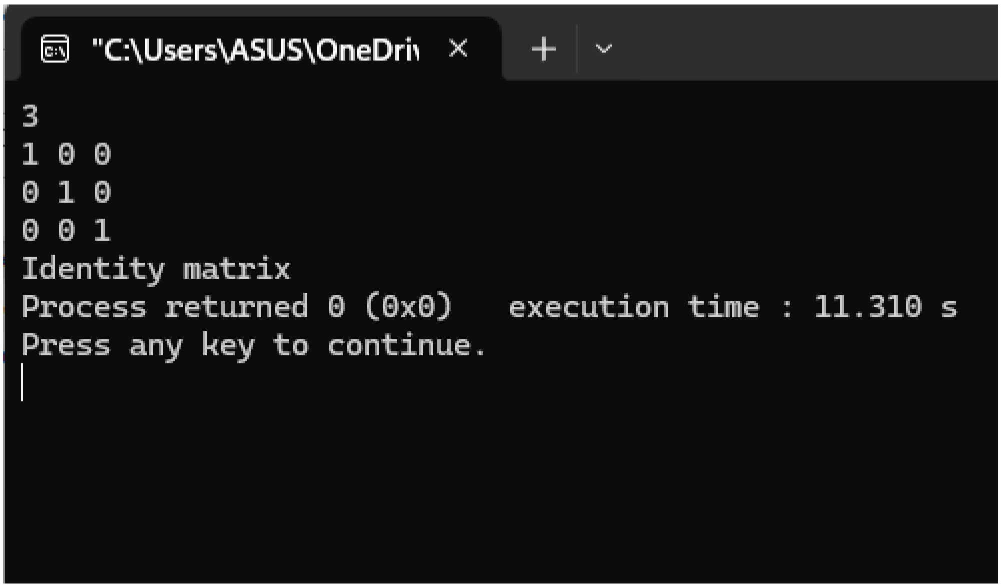


```
Enter size of array: 7
Enter 7 elements:
33 54 6 12 76 65 3
Rearranged array:
3 76 6 65 12 54 33
Process returned 0 (0x0)   execution time : 18.179 s
Press any key to continue.
```

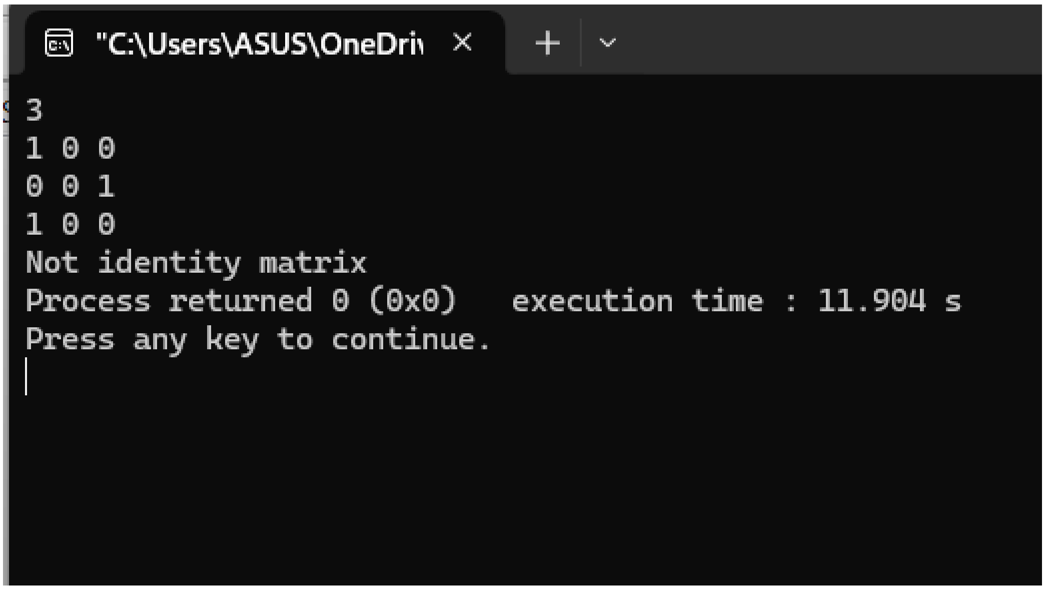
Code:

```
#include <stdio.h>
int main()
{
    int n, i, j, flag = 1;
    scanf("%d", &n);
    int a[n][n];
    for(i = 0; i < n; i++)
        for(j = 0; j < n; j++)
            scanf("%d", &a[i][j]);
    for(i = 0; i < n; i++)
{
        for(j = 0; j < n; j++)
{
            if(i == j && a[i][j] != 1)
                flag = 0;
            if(i != j && a[i][j] != 0)
                flag = 0;
        }
    }

    if(flag)
        printf("Identity matrix");
    else
        printf("Not identity matrix");

    return 0;
}
```

Output:



```
3
1 0 0
0 1 0
0 0 1
Identity matrix
Process returned 0 (0x0)   execution time : 11.310 s
Press any key to continue.
```

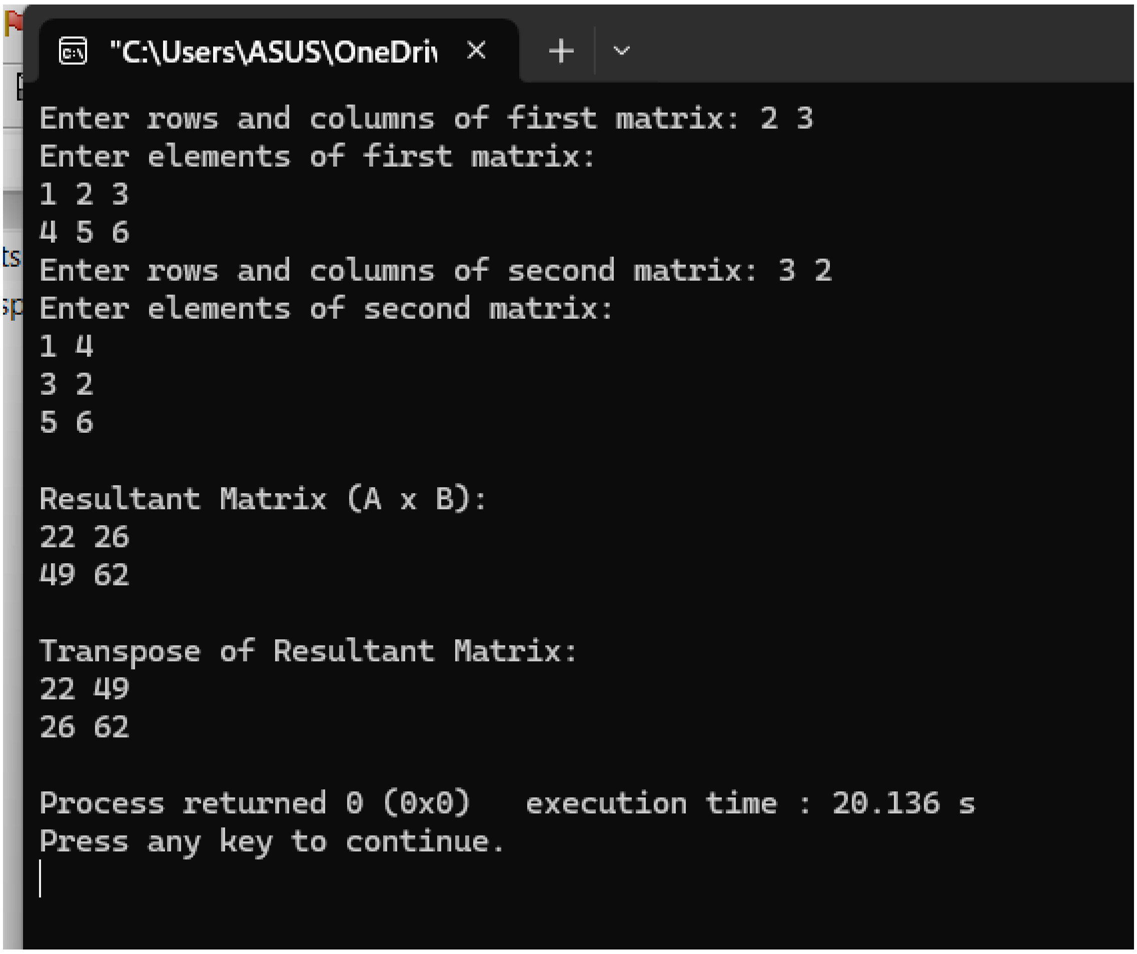


```
3
1 0 0
0 0 1
1 0 0
Not identity matrix
Process returned 0 (0x0)   execution time : 11.904 s
Press any key to continue.
```

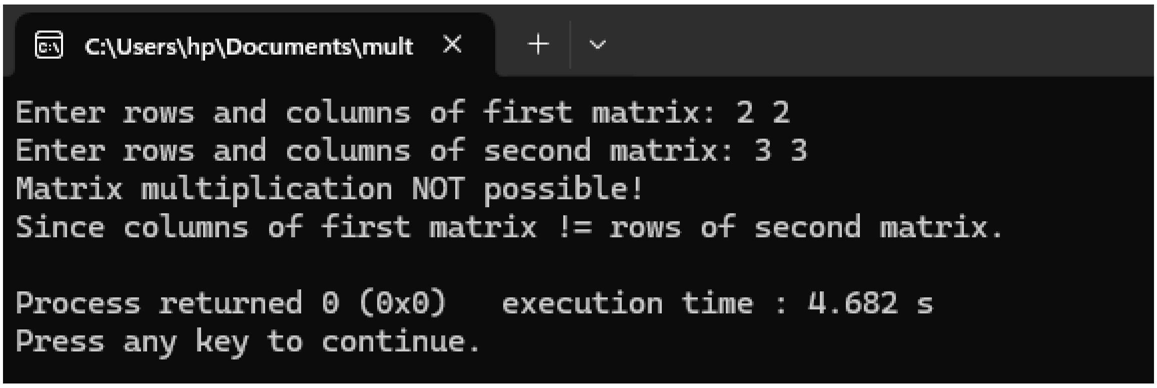
Code:

```
#include <stdio.h>
int main()
{
    int r1, c1, r2, c2, i, j, k;
    scanf("%d %d", &r1, &c1);
    int A[r1][c1];
    for(i = 0; i < r1; i++)
        for(j = 0; j < c1; j++)
            scanf("%d", &A[i][j]);
    scanf("%d %d", &r2, &c2);
    int B[r2][c2];
    for(i = 0; i < r2; i++)
        for(j = 0; j < c2; j++)
            scanf("%d", &B[i][j]);
    if(c1 != r2)
{
        printf("Multiplication not possible");
        return 0;
    }
    int C[r1][c2];
    for(i = 0; i < r1; i++)
        for(j = 0; j < c2; j++)
{
            C[i][j] = 0;
            for(k = 0; k < c1; k++)
                C[i][j] += A[i][k] * B[k][j];
        }

    for(i = 0; i < r1; i++)
{
        for(j = 0; j < c2; j++)
            printf("%d ", C[i][j]);
        printf("\n");
    }
    printf("\n");
    for(i = 0; i < c2; i++)
 {
        for(j = 0; j < r1; j++)
            printf("%d ", C[j][i]);
        printf("\n");
    }
    return 0;
}
```

Output:



```
Enter rows and columns of first matrix: 2 3
Enter elements of first matrix:
1 2 3
4 5 6
Enter rows and columns of second matrix: 3 2
Enter elements of second matrix:
1 4
3 2
5 6

Resultant Matrix (A x B):
22 26
49 62

Transpose of Resultant Matrix:
22 49
26 62

Process returned 0 (0x0)   execution time : 20.136 s
Press any key to continue.
```

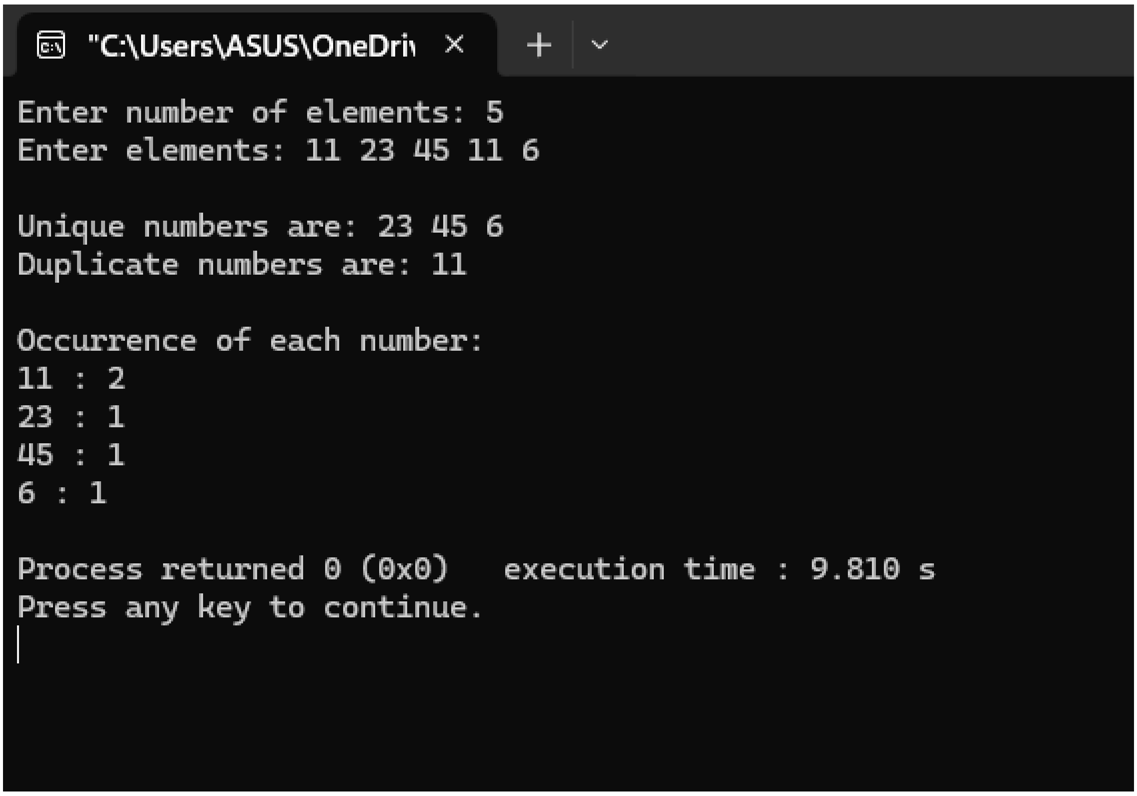


```
Enter rows and columns of first matrix: 2 2
Enter rows and columns of second matrix: 3 3
Matrix multiplication NOT possible!
Since columns of first matrix != rows of second matrix.

Process returned 0 (0x0)   execution time : 4.682 s
Press any key to continue.
```

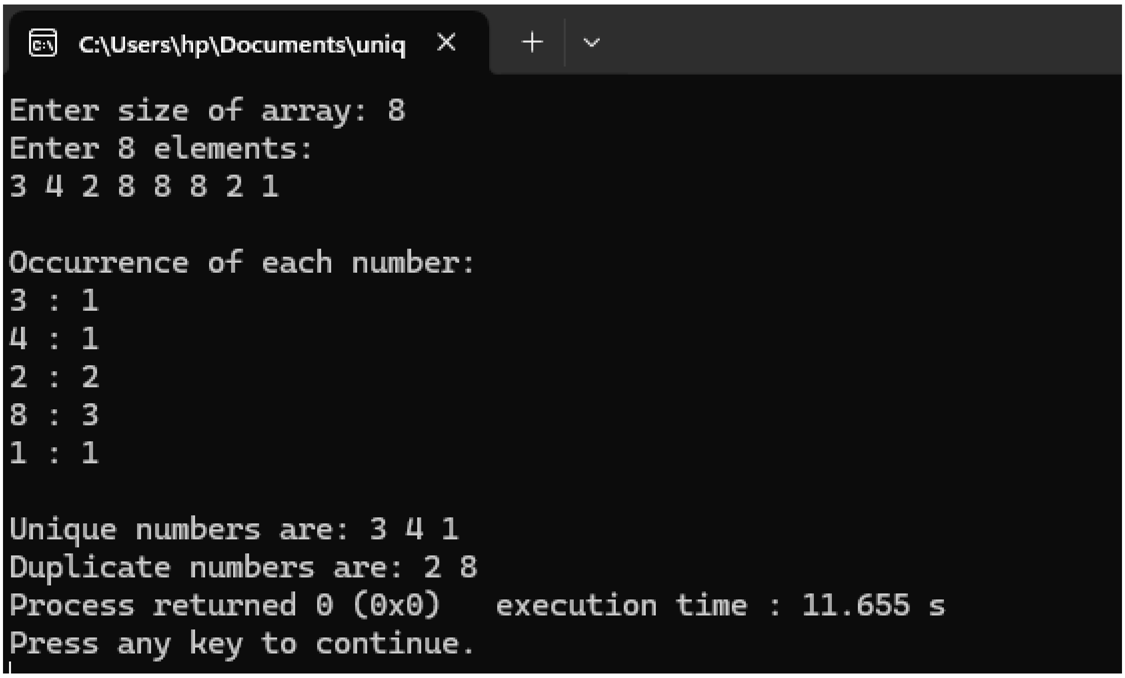
Code:

```
#include <stdio.h>
int main()
{
    int n, i, j;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    int a[n], freq[n];
    printf("Enter elements: ");
    for(i = 0; i < n; i++)
 {
        scanf("%d", &a[i]);
        freq[i] = -1;
    }

    for(i = 0; i < n; i++)
{
        int count = 1;
        if(freq[i] == -1) {
            for(j = i + 1; j < n; j++)
 {
                if(a[i] == a[j])
 {
                    count++;
                    freq[j] = 0;
                }
            }
            freq[i] = count;
        }
    }
    printf("\nUnique numbers are: ");
    for(i = 0; i < n; i++)
        if(freq[i] == 1)
            printf("%d ", a[i]);
    printf("\nDuplicate numbers are: ");
    for(i = 0; i < n; i++)
        if(freq[i] > 1)
            printf("%d ", a[i]);
    printf("\n\nOccurrence of each number:\n");
    for(i = 0; i < n; i++)
        if(freq[i] > 0)
            printf("%d : %d\n", a[i], freq[i]);
    return 0;
}
```

Output:



```
Enter number of elements: 5
Enter elements: 11 23 45 11 6

Unique numbers are: 23 45 6
Duplicate numbers are: 11

Occurrence of each number:
11 : 2
23 : 1
45 : 1
6 : 1

Process returned 0 (0x0)   execution time : 9.810 s
Press any key to continue.
```

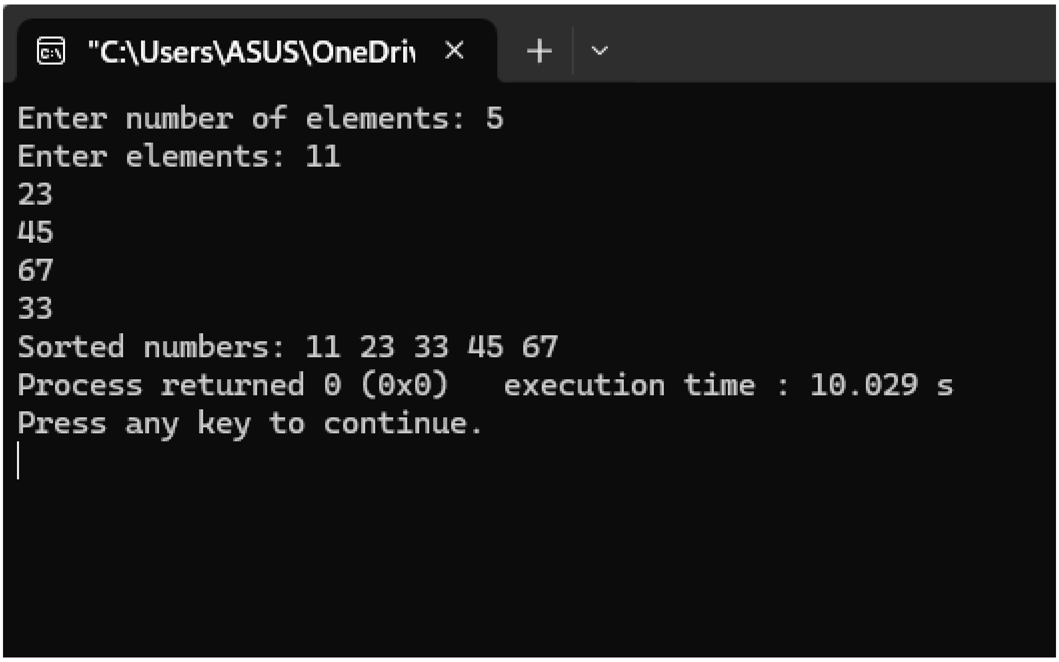


```
Enter size of array: 8
Enter 8 elements:
3 4 2 8 8 8 2 1

Occurrence of each number:
3 : 1
4 : 1
2 : 2
8 : 3
1 : 1

Unique numbers are: 3 4 1
Duplicate numbers are: 2 8
Process returned 0 (0x0)   execution time : 11.655 s
Press any key to continue.
```

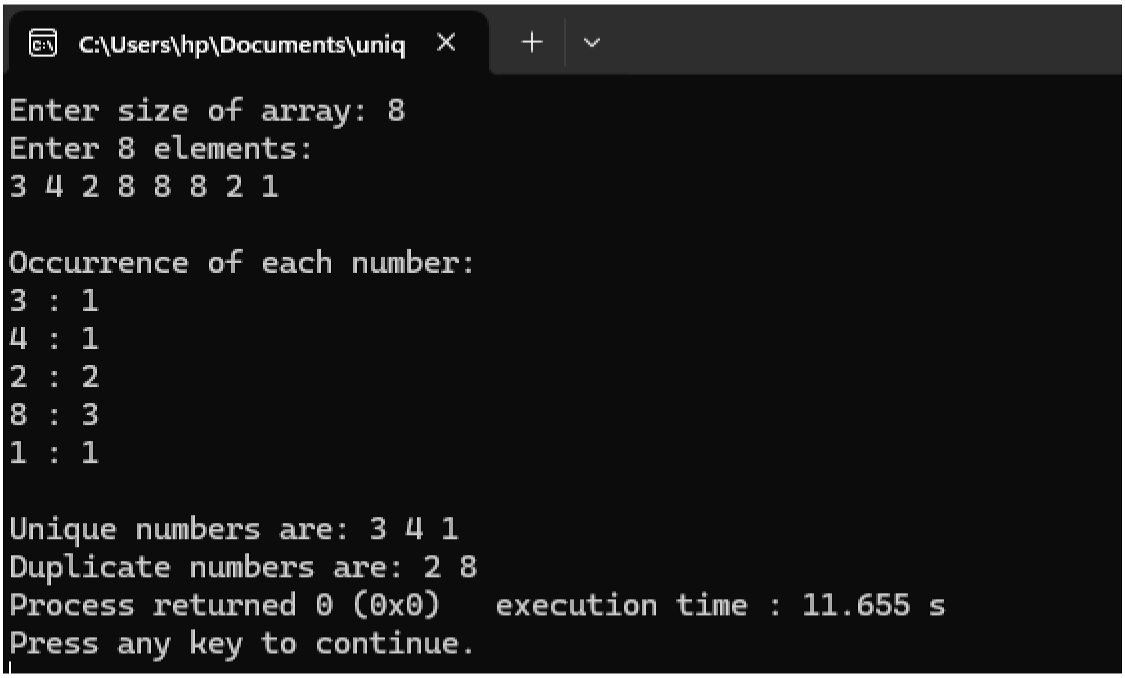
Code:

```
#include <stdio.h>
int main()
{
    int n, i, j, min, temp;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    int a[n];
    printf("Enter elements: ");
    for(i = 0; i < n; i++)
        scanf("%d", &a[i]);
    for(i = 0; i < n - 1; i++)
 {
        min = i;
        for(j = i + 1; j < n; j++)
            if(a[j] < a[min])
                min = j;
        temp = a[i];
        a[i] = a[min];
        a[min] = temp;
    }
    printf("Sorted numbers: ");
    for(i = 0; i < n; i++)
        printf("%d ", a[i]);
    return 0;
}
```

Output:



```
Enter number of elements: 5
Enter elements: 11
23
45
67
33
Sorted numbers: 11 23 33 45 67
Process returned 0 (0x0)   execution time : 10.029 s
Press any key to continue.
```

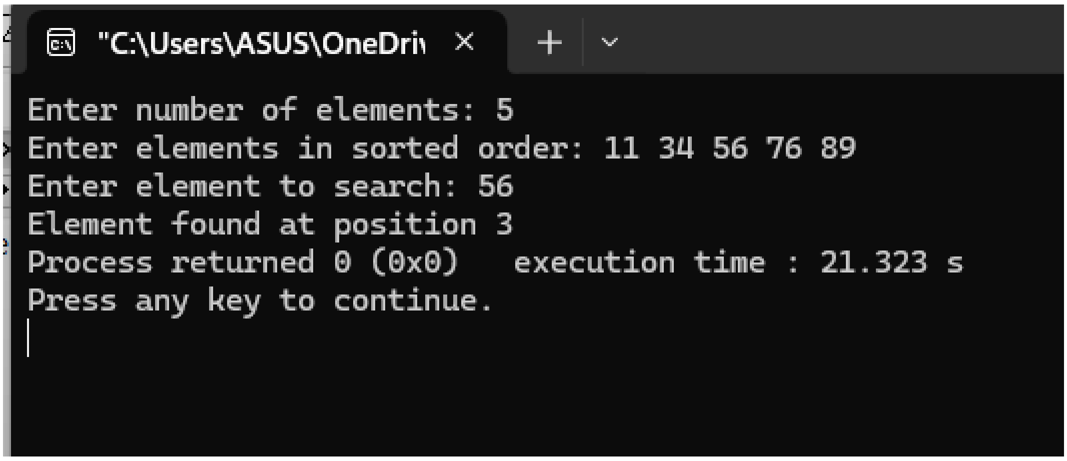


```
Enter size of array: 8
Enter 8 elements:
3 4 2 8 8 8 2 1

Occurrence of each number:
3 : 1
4 : 1
2 : 2
8 : 3
1 : 1

Unique numbers are: 3 4 1
Duplicate numbers are: 2 8
Process returned 0 (0x0)   execution time : 11.655 s
Press any key to continue.
```

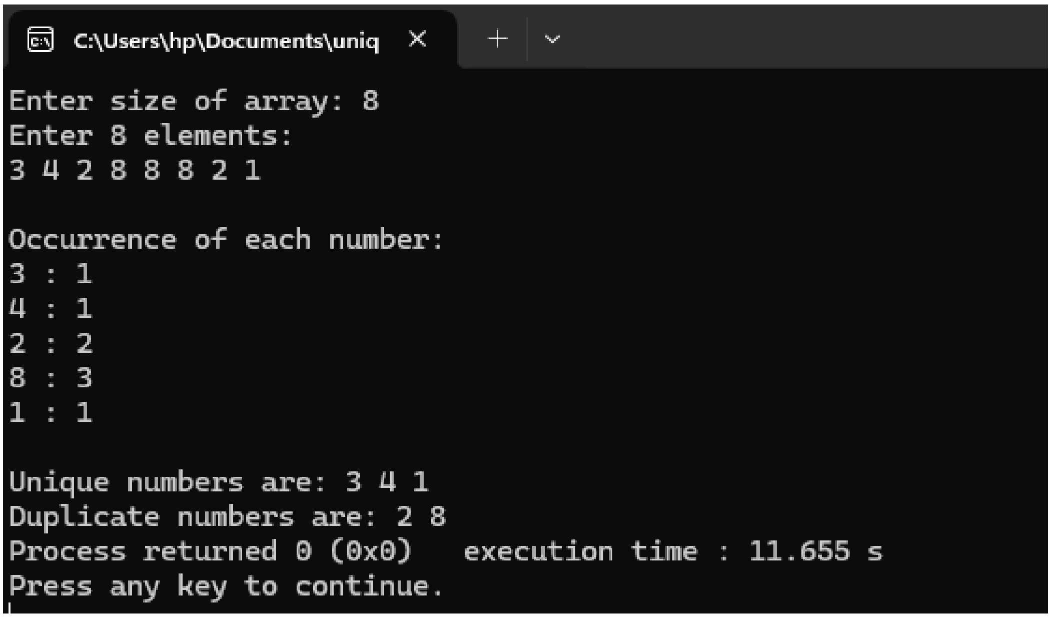
Code:

```
#include <stdio.h>
int main()
 {
   int n, i, key, low, high, mid, found = 0;
   printf("Enter number of elements: ");
   scanf("%d", &n);
   int a[n];
   printf("Enter elements in sorted order: ");
   for(i = 0; i < n; i++)
      scanf("%d", &a[i]);
   printf("Enter element to search: ");
   scanf("%d", &key);
   low = 0;
   high = n - 1;
   while(low <= high)
 {
      mid = (low + high) / 2;
      if(a[mid] == key)
{
         found = 1;
         break;
      }
      else if(key < a[mid])
         high = mid - 1;
      else
         low = mid + 1;
   }

   if(found)
      printf("Element found at position %d", mid + 1);
   else
      printf("Element not found");
   return 0;
}
```

Output:



```
Enter number of elements: 5
Enter elements in sorted order: 11 34 56 76 89
Enter element to search: 56
Element found at position 3
Process returned 0 (0x0)   execution time : 21.323 s
Press any key to continue.
```

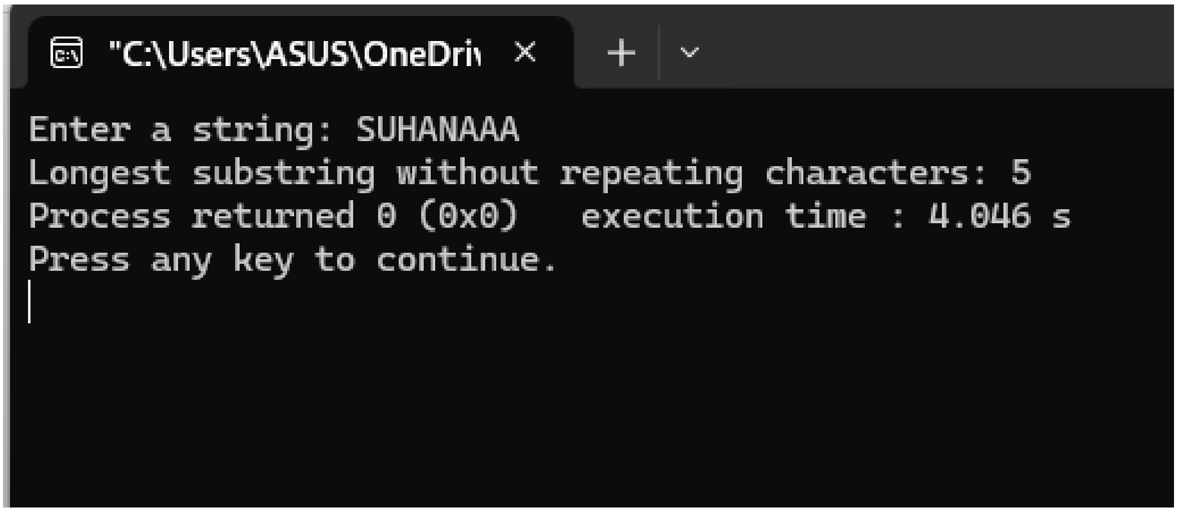


```
Enter size of array: 8
Enter 8 elements:
3 4 2 8 8 8 2 1

Occurrence of each number:
3 : 1
4 : 1
2 : 2
8 : 3
1 : 1

Unique numbers are: 3 4 1
Duplicate numbers are: 2 8
Process returned 0 (0x0)   execution time : 11.655 s
Press any key to continue.
```

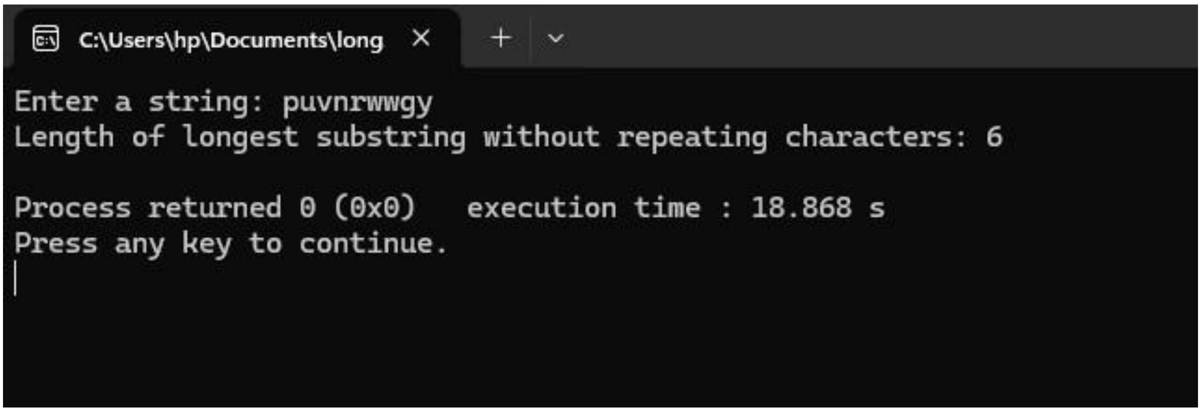
Code:

```
#include <stdio.h>
#include <string.h>
int main()
{
    char s[100];
    int i, start = 0, max = 0, last[256];
    printf("Enter a string: ");
    scanf("%s", s);
    for(i = 0; i < 256; i++)
        last[i] = -1;
    for(i = 0; s[i] != '\0'; i++)
{
        if(last[(unsigned char)s[i]] >= start)
            start = last[(unsigned char)s[i]] + 1;
        last[(unsigned char)s[i]] = i;
        if(i - start + 1 > max)
            max = i - start + 1;
    }
    printf("Longest substring without repeating characters: %d", max);
    return 0;
}
```

Output:



```
Enter a string: SUHANAAA
Longest substring without repeating characters: 5
Process returned 0 (0x0)   execution time : 4.046 s
Press any key to continue.
```

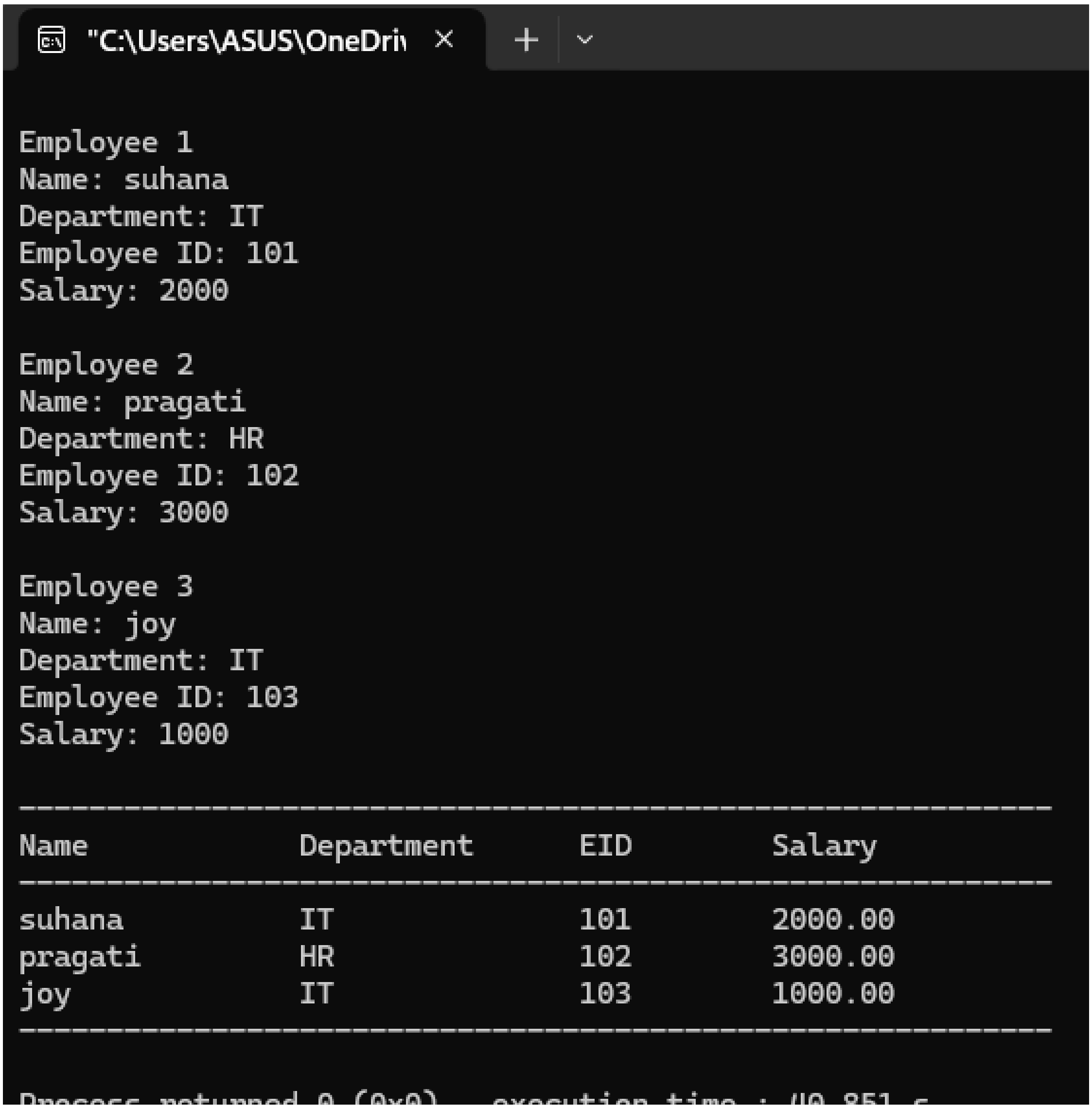


```
Enter a string: puvnrwwgy
Length of longest substring without repeating characters: 6

Process returned 0 (0x0)   execution time : 18.868 s
Press any key to continue.
```

Code:

```
#include <stdio.h>
struct Employee
{
   char name[50];
   char department[50];
   int eid;
   float salary;
};
int main()
{
   int n, i;
   printf("Enter number of employees: ");
   scanf("%d", &n);
   struct Employee e[n];
   struct Employee *p = e;
   printf("Enter employee details:\n");
   for(i = 0; i < n; i++)
{
      printf("Employee %d\n", i + 1);
      printf("Name: ");
      scanf("%s", (p + i)->name);
      printf("Department: ");
      scanf("%s", (p + i)->department);
      printf("Employee ID: ");
      scanf("%d", &(p + i)->eid);
      printf("Salary: ");
      scanf("%f", &(p + i)->salary);
   }
   printf("\nEmployee Records:\n");
   for(i = 0; i < n; i++)
{
      printf("\nEmployee %d\n", i + 1);
      printf("Name: %s\n", (p + i)->name);
      printf("Department: %s\n", (p + i)->department);
      printf("Employee ID: %d\n", (p + i)->eid);
      printf("Salary: %.2f\n", (p + i)->salary);
   }
   return 0;
}
```

Output:



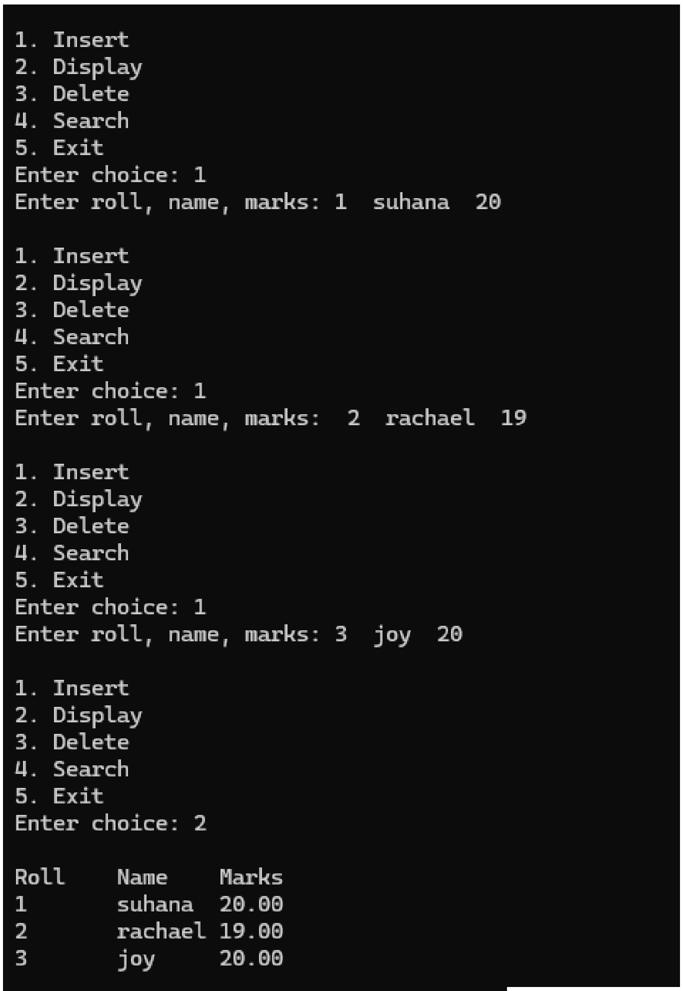
```
Employee 1
Name: suhana
Department: IT
Employee ID: 101
Salary: 2000

Employee 2
Name: pragati
Department: HR
Employee ID: 102
Salary: 3000

Employee 3
Name: joy
Department: IT
Employee ID: 103
Salary: 1000

------------------------------------------------------------
Name            Department      EID        Salary
------------------------------------------------------------
suhana          IT              102        2000.00
pragati         HR              102        3000.00
joy             IT              103        1000.00
------------------------------------------------------------

Process returned 0 (0x0)   execution time : 40.851 s
```

Code:

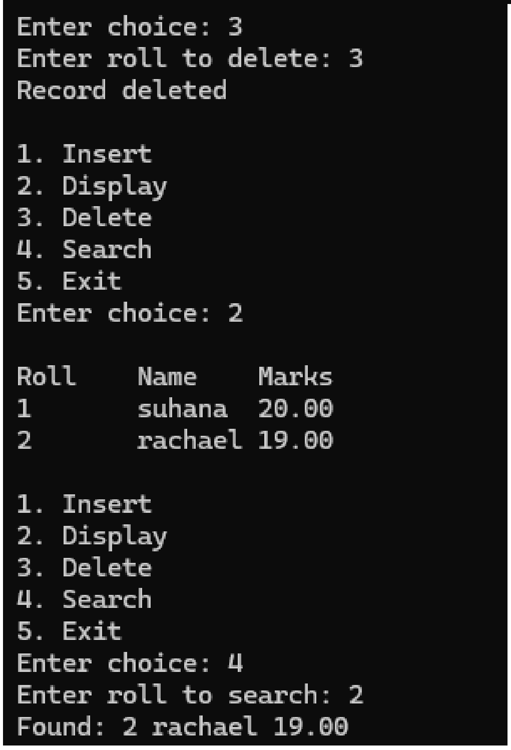
```
#include <stdio.h>
#include <string.h>
struct Student
{
    int roll;
    char name[50];
    float marks;
};
int main()
 {
    struct Student s[100];
    int n = 0, choice, i, pos, roll, found;
    while(1)
{
        printf("\n1. Insert\n2. Display\n3. Delete\n4. Search\n5. Exit\n");
        printf("Enter choice: ");
        scanf("%d", &choice);
        if(choice == 1)
{
            printf("Enter roll, name, marks: ");
            scanf("%d %s %f", &s[n].roll, s[n].name, &s[n].marks);
            n++;
        }

        else if(choice == 2)
 {
            if(n == 0)
                printf("No records\n");
            else
{
                printf("\nRoll\tName\tMarks\n");
                for(i = 0; i < n; i++)
                    printf("%d\t%s\t%.2f\n", s[i].roll, s[i].name, s[i].marks);
            }
        }

        else if(choice == 3)
{
            printf("Enter roll to delete: ");
            scanf("%d", &roll);
            found = 0;
            for(i = 0; i < n; i++)
{
                if(s[i].roll == roll)
 {
                    found = 1;
                    for(int j = i; j < n - 1; j++)
                        s[j] = s[j + 1];
                    n--;
                    printf("Record deleted\n");
                    break;
                }
            }
```

```
      if(!found)
        printf("Record not found\n");
    }

    else if(choice == 4)
{
      printf("Enter roll to search: ");
      scanf("%d", &roll);
      found = 0;
      for(i = 0; i < n; i++)
{
        if(s[i].roll == roll)
 {
          found = 1;
          printf("Found: %d %s %.2f\n", s[i].roll, s[i].name, s[i].marks);
          break;
        }
      }

      if(!found)
        printf("Record not found\n");
    }

    else if(choice == 5)
      break;
    else
      printf("Invalid choice\n");
  }
  return 0;
}
```

Output:



```
1. Insert
2. Display
3. Delete
4. Search
5. Exit
Enter choice: 1
Enter roll, name, marks: 1  suhana  20

1. Insert
2. Display
3. Delete
4. Search
5. Exit
Enter choice: 1
Enter roll, name, marks:  2  rachael  19

1. Insert
2. Display
3. Delete
4. Search
5. Exit
Enter choice: 1
Enter roll, name, marks: 3  joy  20

1. Insert
2. Display
3. Delete
4. Search
5. Exit
Enter choice: 2

Roll    Name    Marks
1       suhana  20.00
2       rachael 19.00
3       joy     20.00
```



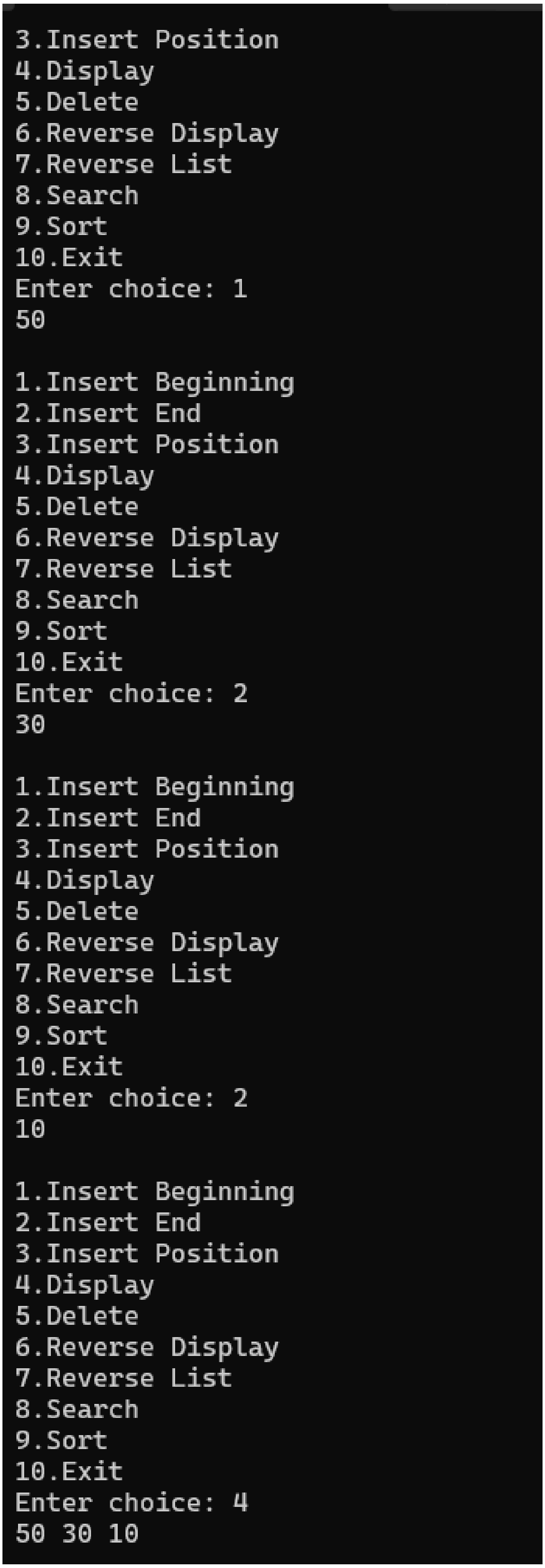
```
Enter choice: 3
Enter roll to delete: 3
Record deleted

1. Insert
2. Display
3. Delete
4. Search
5. Exit
Enter choice: 2

Roll    Name    Marks
1       suhana  20.00
2       rachael 19.00

1. Insert
2. Display
3. Delete
4. Search
5. Exit
Enter choice: 4
Enter roll to search: 2
Found: 2 rachael 19.00
```

Code:

```
#include <stdio.h>
#include <stdlib.h>
struct Node
{
    int data;
    struct Node *next;
};
struct Node *head = NULL;
void insertBeginning(int val)
{
    struct Node *newnode = malloc(sizeof(struct Node));
    newnode->data = val;
    newnode->next = head;
    head = newnode;
}
void insertEnd(int val)
{
    struct Node *newnode = malloc(sizeof(struct Node));
    newnode->data = val;
    newnode->next = NULL;
    if(head == NULL) head = newnode;
    else
 {
        struct Node *temp = head;
        while(temp->next) temp = temp->next;
        temp->next = newnode;
    }
}
void insertPosition(int val, int pos)
 {
    struct Node *newnode = malloc(sizeof(struct Node));
    newnode->data = val;
    if(pos == 1)
{
        newnode->next = head;
        head = newnode;
        return;
    }
    struct Node *temp = head;
    for(int i = 1; i < pos - 1 && temp; i++)
        temp = temp->next;
    if(temp == NULL) return;
    newnode->next = temp->next;
    temp->next = newnode;
}
void display()
{
    struct Node *temp = head;
    if(temp == NULL) {
        printf("List Empty\n");
        return;
    }
    while(temp)
```

```
{
        printf("%d ", temp->data);
        temp = temp->next;
    }
    printf("\n");
}
void deleteNode(int val)
{
    if(head == NULL) return;
    if(head->data == val)
{
        struct Node *temp = head;
        head = head->next;
        free(temp);
        return;
    }
    struct Node *temp = head;
    while(temp->next && temp->next->data != val)
        temp = temp->next;
    if(temp->next)
 {
        struct Node *del = temp->next;
        temp->next = del->next;
        free(del);
    }
}
void reverseDisplay(struct Node *node)
{
    if(node == NULL) return;
    reverseDisplay(node->next);
    printf("%d ", node->data);
}
void reverseList()
{
    struct Node *prev = NULL, *cur = head, *next;
    while(cur)
 {
        next = cur->next;
        cur->next = prev;
        prev = cur;
        cur = next;
    }
    head = prev;
}
void search(int val)
{
    struct Node *temp = head;
    int pos = 1;
    while(temp)
 {
        if(temp->data == val) {
            printf("Found at position %d\n", pos);
            return;
        }
        temp = temp->next;
        pos++;
    }
    printf("Not found\n");
}
void sortList()
 {
    struct Node *i, *j;
    for(i = head; i && i->next; i = i->next)
        for(j = i->next; j; j = j->next)
            if(i->data > j->data)
 {
                int t = i->data;
                i->data = j->data;
                j->data = t;
            }
}
int main()
{
    int choice, val, pos;
    while(1)
{
        printf("\n1.Insert Beginning\n2.Insert End\n3.Insert
Position\n4.Display\n5.Delete\n6.Reverse Display\n7.Reverse
List\n8.Search\n9.Sort\n10.Exit\n");
        printf("Enter choice: ");
        scanf("%d", &choice);
        if(choice == 1)
{
            scanf("%d", &val);
            insertBeginning(val);
        }
        else if(choice == 2)
 {
            scanf("%d", &val);
            insertEnd(val);
        }
        else if(choice == 3)
{
            scanf("%d %d", &val, &pos);
            insertPosition(val, pos);
        }
        else if(choice == 4) display();
        else if(choice == 5)
{
            scanf("%d", &val);
            deleteNode(val);
        }
        else if(choice == 6)
{
            reverseDisplay(head);
            printf("\n");
        }
        else if(choice == 7) reverseList();
        else if(choice == 8)
{
      scanf("%d", &val);
      search(val);
    }
    else if(choice == 9) sortList();
    else if(choice == 10) break;
    else printf("Invalid choice\n");
  }
  return 0;
}
```

Output:

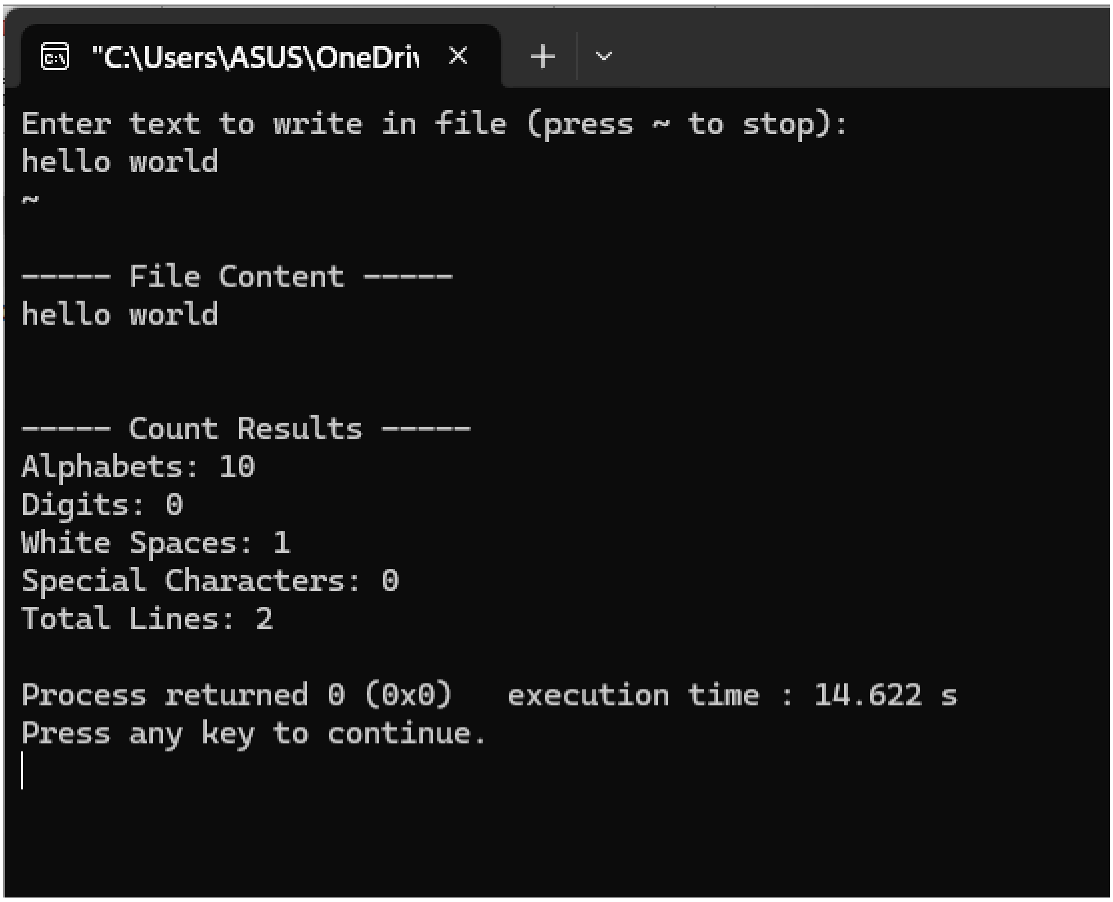


```
3.Insert Position
4.Display
5.Delete
6.Reverse Display
7.Reverse List
8.Search
9.Sort
10.Exit
Enter choice: 1
50

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

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

1.Insert Beginning
2.Insert End
3.Insert Position
4.Display
5.Delete
6.Reverse Display
7.Reverse List
8.Search
9.Sort
10.Exit
Enter choice: 4
50 30 10
```

Code:

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

int main()
 {
    FILE *fp;
    char ch;
    int alpha = 0, digit = 0, space = 0, special = 0, lines = 1;

    fp = fopen("C:\\Data.txt", "w");
    if(fp == NULL){
        printf("Cannot open file.");
        return 0;
    }

    printf("Enter text to write in file (press ~ to stop):\n");
    while((ch = getchar()) != '~')
{
        fputc(ch, fp);
    }
    fclose(fp);

    fp = fopen("C:\\Data.txt", "r");
    if(fp == NULL){
        printf("Cannot open file.");
        return 0;
    }

    printf("\n----- File Content -----\n");
    while((ch = fgetc(fp)) != EOF)
{
        putchar(ch);

        if(isalpha(ch)) alpha++;
        else if(isdigit(ch)) digit++;
        else if(ch == ' ' || ch == '\t') space++;
        else if(ch == '\n') lines++;
        else special++;
    }
    fclose(fp);

    printf("\n\n----- Count Results -----\n");
    printf("Alphabets: %d\n", alpha);
    printf("Digits: %d\n", digit);
    printf("White Spaces: %d\n", space);
    printf("Special Characters: %d\n", special);
    printf("Total Lines: %d\n", lines);

    return 0;
}
```

Output:



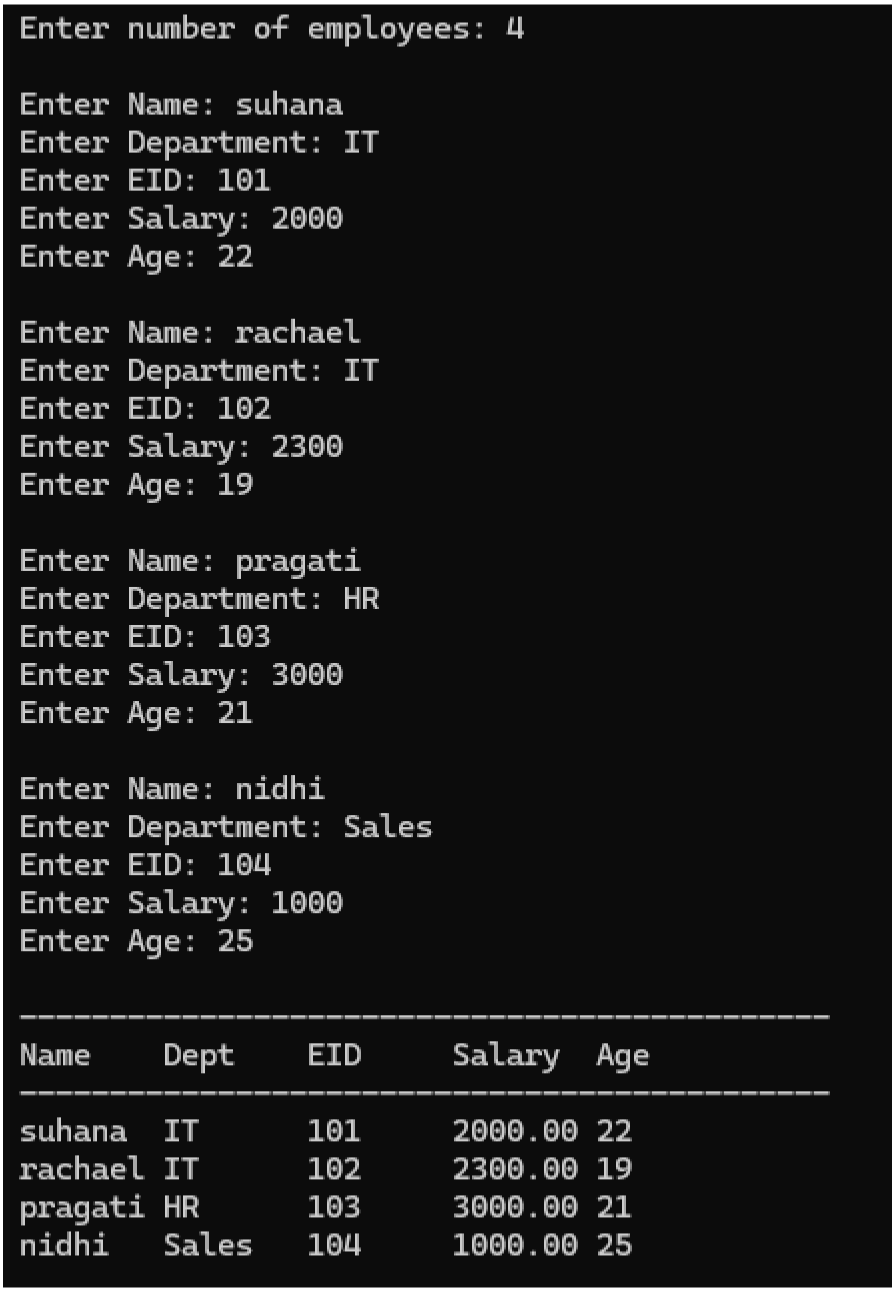
```
Enter text to write in file (press ~ to stop):
hello world
~

----- File Content -----
hello world


----- Count Results -----
Alphabets: 10
Digits: 0
White Spaces: 1
Special Characters: 0
Total Lines: 2

Process returned 0 (0x0)   execution time : 14.622 s
Press any key to continue.
```

Code:

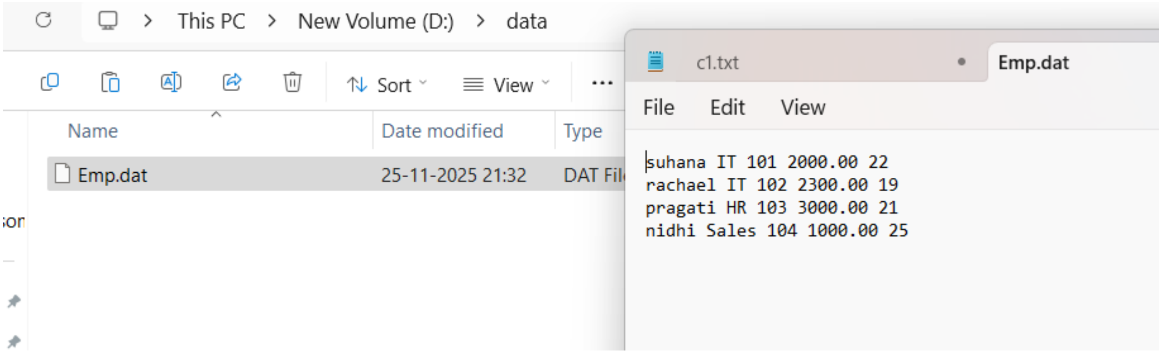
```
#include <stdio.h>

struct Employee
 {
   char name[30];
   char dept[20];
   int eid;
   float sal;
   int age;
};

int main()
 {
   FILE *fp;
   struct Employee e;
   int n, i;

   fp = fopen("D:\\data\\Emp.dat", "w");
   if(fp == NULL)
 {
      printf("Cannot open file.");
      return 0;
   }

   printf("Enter number of employees: ");
   scanf("%d", &n);

   for(i = 0; i < n; i++)
 {
      printf("\nEnter Name: ");
      scanf("%s", e.name);
      printf("Enter Department: ");
      scanf("%s", e.dept);
      printf("Enter EID: ");
      scanf("%d", &e.eid);
      printf("Enter Salary: ");
      scanf("%f", &e.sal);
      printf("Enter Age: ");
      scanf("%d", &e.age);

      fprintf(fp, "%s %s %d %.2f %d\n", e.name, e.dept, e.eid, e.sal, e.age);
   }

   fclose(fp);

   fp = fopen("D:\\data\\Emp.dat", "r");
   if(fp == NULL) {
      printf("Cannot open file for reading.");
      return 0;
   }

   printf("\n-------------------------------------------\n");
   printf("Name\tDept\tEID\tSalary\tAge\n");
```

```
    printf("--------------------------------------------\n");

    while(fscanf(fp, "%s %s %d %f %d", e.name, e.dept, &e.eid, &e.sal, &e.age) != EOF)
 {
        printf("%s\t%s\t%d\t%.2f\t%d\n", e.name, e.dept, e.eid, e.sal, e.age);
    }

    fclose(fp);
    return 0;
}
```

Output:



```
Enter number of employees: 4

Enter Name: suhana
Enter Department: IT
Enter EID: 101
Enter Salary: 2000
Enter Age: 22

Enter Name: rachael
Enter Department: IT
Enter EID: 102
Enter Salary: 2300
Enter Age: 19

Enter Name: pragati
Enter Department: HR
Enter EID: 103
Enter Salary: 3000
Enter Age: 21

Enter Name: nidhi
Enter Department: Sales
Enter EID: 104
Enter Salary: 1000
Enter Age: 25

---------------------------------------------
Name    Dept    EID     Salary  Age
---------------------------------------------
suhana  IT      101     2000.00 22
rachael IT      102     2300.00 19
pragati HR      103     3000.00 21
nidhi   Sales   104     1000.00 25
```

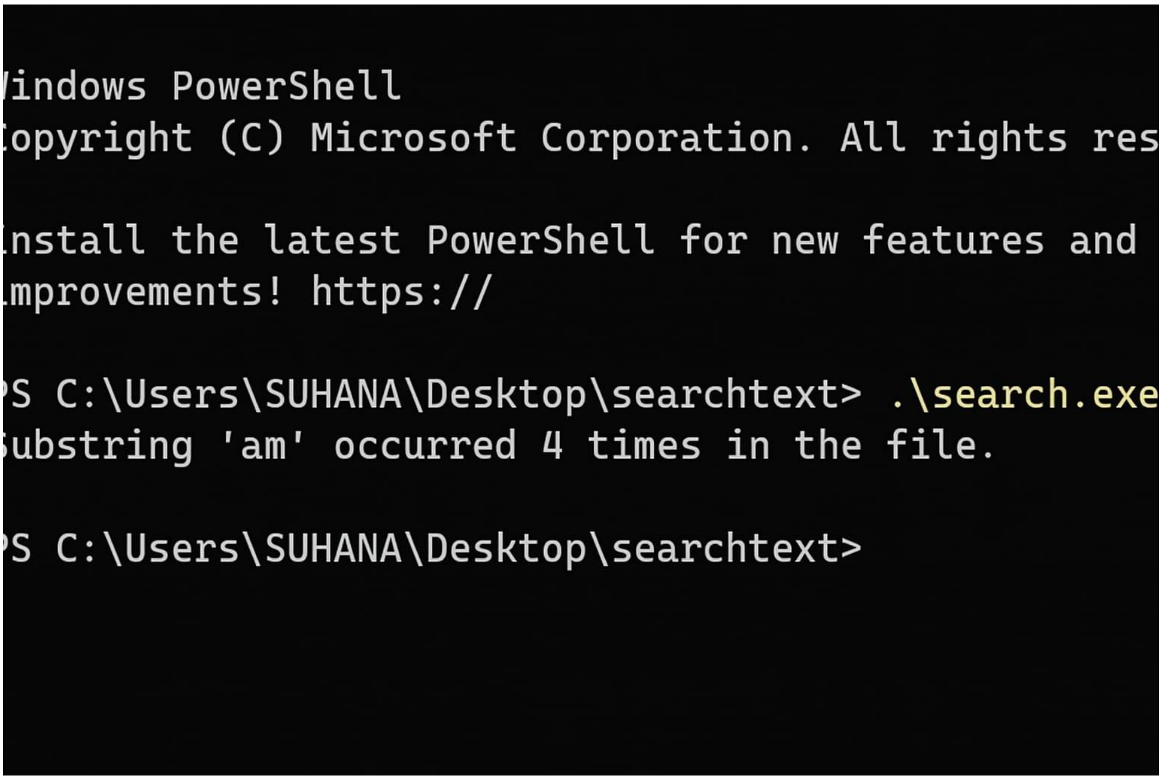


```
suhana IT 101 2000.00 22
rachael IT 102 2300.00 19
pragati HR 103 3000.00 21
nidhi Sales 104 1000.00 25
```

Code:

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

int main(int argc, char *argv[])
{
   FILE *fp;
   char ch, buffer[1000];
   int i = 0, count = 0;
   char *pos;

   if(argc != 3)
 {
      printf("Usage: <program> <filename> <substring>");
      return 0;
   }

   fp = fopen(argv[1], "r");
   if(fp == NULL)
 {
      printf("Cannot open file.");
      return 0;
   }

   while((ch = fgetc(fp)) != EOF)
{
      buffer[i++] = ch;
   }
   buffer[i] = '\0';
   fclose(fp);

   pos = buffer;
   while((pos = strstr(pos, argv[2])) != NULL)
 {
      count++;
      pos += strlen(argv[2]);
   }

   printf("Substring '%s' found %d times in file '%s'.\n",
        argv[2], count, argv[1]);

   return 0;
}
```

Output:

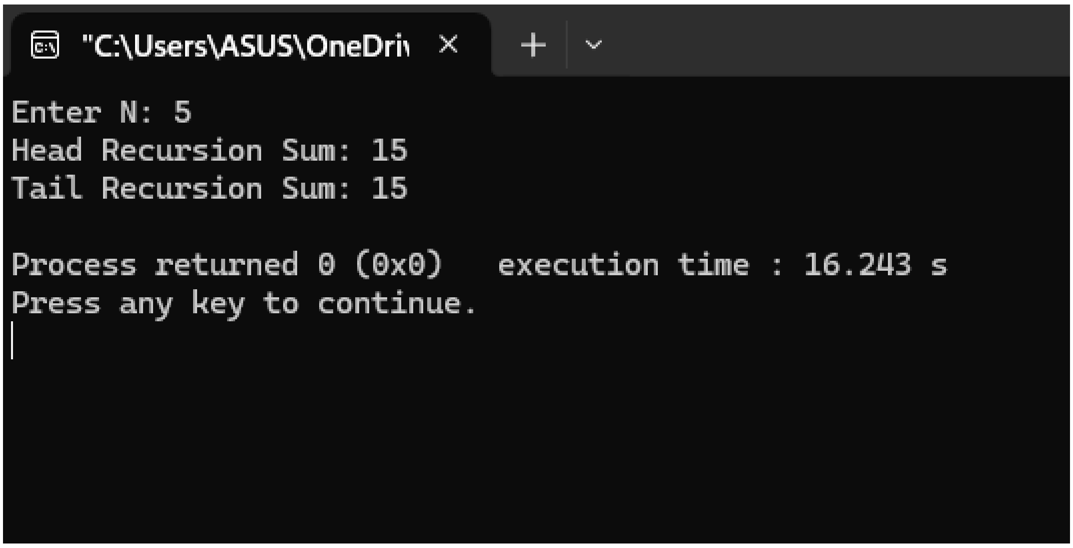


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights res

Install the latest PowerShell for new features and
improvements! https://

PS C:\Users\SUHANA\Desktop\searchtext> .\search.exe
Substring 'am' occurred 4 times in the file.

PS C:\Users\SUHANA\Desktop\searchtext>
```

Code:

```
#include <stdio.h>

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

int tailSum(int n, int total)
{
   if(n == 0)
      return total;
   return tailSum(n - 1, total + 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:



```
Enter N: 5
Head Recursion Sum: 15
Tail Recursion Sum: 15

Process returned 0 (0x0)   execution time : 16.243 s
Press any key to continue.
```