

Name: Payal Patel

Roll No: 17

Subject: Fundamentals of Programming

course: PGDCSA Sem1

```
// Print Program No. 1
// Write a program to determine the maximum of 3 numbers.
#include <stdio.h>
int main() {
    int num1, num2, num3;
    int max;
    printf("Enter the first number: ");
    scanf("%d", &num1);
    printf("Enter the second number: ");
    scanf("%d", &num2);
    printf("Enter the third number: ");
    scanf("%d", &num3);
    if (num1 >= num2 && num1 >= num3) {
        max = num1;
    } else if (num2 >= num1 && num2 >= num3) {
        max = num2;
    } else {
        max = num3;
    }
    printf("The maximum of the three int numbers is %d.\n", max);
    return 0;
}
```

Output:

Enter the first number: 1

Enter the second number: 2

Enter the third number: 3

The maximum of the three int numbers is 3.

```
// Print Program No. 2
// Write a program to swap the values of two variables.
#include <stdio.h>
int main() {
    int a, b;
    printf("Enter the value for a: ");
    scanf("%d", &a);
    printf("Enter the value for b: ");
    scanf("%d", &b);
    printf("Before swapping: a = %d, b = %d\n", a, b);
    a = a + b;
    b = a - b;
    a = a - b;
    printf("After swapping: a = %d, b = %d\n", a, b);
    return 0;
}
```

Output:

Enter the value for a: 10

Enter the value for b: 20

Before swapping: a = 10, b = 20

After swapping: a = 20, b = 10

// Print Program No. 3

/\*Write a program that reads the percentage obtained by the students and determines and prints the class obtained by the student as per the following rules

Percentage Class

0 - 39 Fail

40 - 59 Second class

60 - 79 First class

80 - 100 Distinction\*/

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

```
int main() {
```

```
    float percentage;
```

```
    printf("Enter the percentage obtained by the student: ");
```

```
    scanf("%f", &percentage);
```

```
    if (percentage >= 0 && percentage <= 39) {
```

```
        printf("Class: Fail\n");
```

```
    } else if (percentage >= 40 && percentage <= 59) {
```

```
        printf("Class: Second class\n");
```

```
    } else if (percentage >= 60 && percentage <= 79) {
```

```
        printf("Class: First class\n");
```

```
    } else if (percentage >= 80 && percentage <= 100) {
```

```
        printf("Class: Distinction\n");
```

```
    } else {
```

```
        printf("Invalid percentage. Please enter a value between 0 and 100.\n");
```

```
    }
```

```
    return 0;
```

```
}
```

Output:

Enter the percentage obtained by the student: 75

Class: First class

```
// Print Program No. 4
/*Write a program to calculate the area of circle/rectangle/triangle.
C indicate circle ,
R indicate rectangle,
T indicate triangle.
use symbolic constant to define the value of pie.*/
```

```
#include <stdio.h>
#define PI 3.14159265358979323846
int main() {
    char shape;
    float area;
    printf("Enter the shape type (C for circle, R for rectangle, T for triangle): ");
    scanf(" %c", &shape);
    switch (shape) {
        case 'C':
        case 'c': {
            float radius;
            printf("Enter the radius of the circle: ");
            scanf("%f", &radius);
            area = PI * radius * radius;
            printf("The area of the circle is %.2f\n", area);
            break;
        }
        case 'R':
        case 'r': {
            float length, width;
            printf("Enter the length of the rectangle: ");
            scanf("%f", &length);
            printf("Enter the width of the rectangle: ");
            scanf("%f", &width);
            area = length * width;
            printf("The area of the rectangle is %.2f\n", area);
            break;
        }
        case 'T':
        case 't': {
            float base, height;
            printf("Enter the base of the triangle: ");
            scanf("%f", &base);
            printf("Enter the height of the triangle: ");
            scanf("%f", &height);
            area = 0.5 * base * height;
            printf("The area of the triangle is %.2f\n", area);
            break;
        }
        default:
            printf("Invalid shape type. Please enter C, R, or T.\n");
            break;
    }
    return 0;
}
```

output:

Enter the shape type (C for circle, R for rectangle, T for triangle): r

Enter the length of the rectangle: 50

Enter the width of the rectangle: 30

The area of the rectangle is 1500.00

// Print Program No. 5

// Write a program that accept basic, HRA, and DA from the user and calculate total salary.

```
#include <stdio.h>
int main() {
    float basic, hra, da, total_salary;
    printf("Enter the basic salary: ");
    scanf("%f", &basic);
    printf("Enter the House Rent Allowance (HRA): ");
    scanf("%f", &hra);
    printf("Enter the Dearness Allowance (DA): ");
    scanf("%f", &da);
    total_salary = basic + hra + da;
    printf("The total salary is %.2f\n", total_salary);
    return 0;
}
```

Output:

Enter the basic salary: 16000

Enter the House Rent Allowance (HRA): 5000

Enter the Dearness Allowance (DA): 3000

The total salary is 24000.00

// Print Program No. 6  
// Write a program to print the multiplication table of given number.

```
#include <stdio.h>
int main() {
    int num, i;
    printf("Enter a number to print its multiplication table: ");
    scanf("%d", &num);
    printf("Multiplication table for %d:\n", num);
    for (i = 1; i <= 10; i++) {
        printf("%d x %d = %d\n", num, i, num * i);
    } return 0;
}
```

Output:

Enter a number to print its multiplication table: 15

Multiplication table for 15:

15 x 1 = 15  
15 x 2 = 30  
15 x 3 = 45  
15 x 4 = 60  
15 x 5 = 75  
15 x 6 = 90  
15 x 7 = 105  
15 x 8 = 120  
15 x 9 = 135  
15 x 10 = 150

```
// Print Program No. 7
// Write a program to determine given number is prime or not.
```

```
#include <stdio.h>
int main() {
    int n, i, flag = 0;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    if (n == 0 || n == 1)
        flag = 1;
    for (i = 2; i <= n / 2; ++i) {
        if (n % i == 0) {
            flag = 1;
            break;
        }
    }
    if (flag == 0)
        printf("%d is a prime number.", n);
    else
        printf("%d is not a prime number.", n);
    return 0;
}
```

Output:

```
Enter a positive integer: 16
16 is not a prime number.
```

Output:

```
Enter a positive integer: 2
2 is a prime number.
```



// Print Program No. 8

// Write a program to reverse a given number and display the sum of all digits.

```
#include <stdio.h>
int main() {
    int number, reversedNumber = 0, digit, sumOfDigits = 0;
    printf("Enter an integer: ");
    scanf("%d", &number);
    int originalNumber = number;
    while (number != 0) {
        digit = number % 10;
        reversedNumber = reversedNumber * 10 + digit;
        sumOfDigits += digit;
        number /= 10;
    }
    printf("Reversed number: %d\n", reversedNumber);
    printf("Sum of all digits: %d\n", sumOfDigits);
    return 0;
}
```

Output:

Enter an integer: 12

Reversed number: 21

Sum of all digits: 3

// Print Program No. 9

/\*Write a program to accept two numbers and perform basic operation of calculator (+, -, \*, /).(Use switch...case)\*/

```
#include <stdio.h>
int main() {
    int num1, num2;
    char operator;

    printf("Enter the first number: ");
    scanf("%d", &num1);
    printf("Enter the second number: ");
    scanf("%d", &num2);
    printf("Enter the operator (+, -, *, /): ");
    scanf(" %c", &operator); // Notice the space before %c to consume any leftover newline
    switch (operator) {
        case '+':
            printf("Result: %d\n", num1 + num2);
            break;
        case '-':
            printf("Result: %d\n", num1 - num2);
            break;
        case '*':
            printf("Result: %d\n", num1 * num2);
            break;
        case '/':
            if (num2 != 0) {
                printf("Result: %.2f\n", (float)num1 / num2);
            } else {
                printf("Error: Division by zero is not allowed.\n");
            }
            break;
        default:
            printf("Error: Invalid operator.\n");
            break;
    }

    return 0;
}
```

Output:

Enter the first number: 12

Enter the second number: 21

Enter the operator (+, -, \*, /): +

Result: 33

// Print Program No. 10

// Write a program to find maximum and minimum element from 1-Dimensional array.

```
#include<stdio.h>
int main()
{
    int a[100], i, size,max,min;
    printf("\n enter the size of arr : ");
    scanf("%d",&size);
    for(i=0;i<size;i++)
    {
        printf("\n enter the element of arr %d : ", i);
        scanf("%d",&a[i]);
    }
    max=a[0];
    min=a[0];
    for(i=0;i<size;i++)
    {
        if(max<a[i])
        { max=a[i]; }
        if(min>a[i])
        { min=a[i]; }
    }
    printf("\n maximum element is : %d",max);
    printf("\n minimum element is : %d",min);
}
```

Output:

```
enter the size of arr : 5
enter the element of arr 0 : 11
enter the element of arr 1 : 22
enter the element of arr 2 : 33
enter the element of arr 3 : 44
enter the element of arr 4 : 55
maximum element is : 55
minimum element is : 11
```

// Print Program No. 11  
// Write a program to sort given array in ascending order.

```
#include <stdio.h>
void selectionSort(int arr[], int size) {
    int i, j;
    for (i = 0; i < size - 1; i++) {
        int minIndex = i;
        for (j = i + 1; j < size; j++) {
            if (arr[j] < arr[minIndex]) {
                minIndex = j;
            }
        }
        if (minIndex != i) {
            int temp = arr[i];
            arr[i] = arr[minIndex];
            arr[minIndex] = temp;
        }
    }
}

void printArray(int arr[], int size)
{
    int i;
    for (i = 0; i < size; i++)
    {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

int main()
{
    int n, i;
    printf("Enter the number of elements: ");
    scanf("%d", &n);
    int arr[n];
    printf("Enter %d elements: ", n);
    for (i = 0; i < n; i++)
    {
        scanf("%d", &arr[i]);
    }
    selectionSort(arr, n);
    printf("Sorted array in ascending order: ");
    printArray(arr, n);
    return 0;
}
```

Output:

Enter the number of elements: 4

Enter 4 elements: 22

55

11

99

Sorted array in ascending order: 11 22 55 99

// Print Program No. 12

// Write a program to add two matrices.

```
#include<stdio.h>
int main()
{
    int a[100][100], b[100][100], r, c, i, j, sum[100][100];
    printf("\n Enter the number of rows : ");
    scanf("%d",&r);
    printf("\n Enter the number of columns : ");
    scanf("%d",&c);
    printf("\n Enter the elements of 1st matrices :- ");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            printf("\n Enter the element of a[%d %d] : ",i,j);
            scanf("%d",&a[i][j]);
        }
    }
    printf("\n Enter the elements of 2nd matrices :- ");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            printf("\n Enter the element of a[%d %d] : ",i,j);
            scanf("%d",&b[i][j]);
        }
    }
    printf("\n The elements of 1st matrices is :- \n");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            printf(" %d",a[i][j]);
        }
        printf("\n");
    }
    printf("\n The elements of 2nd matrices is :- \n");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            printf(" %d",b[i][j]);
        }
        printf("\n");
    }
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            sum[i][j] = a[i][j] + b[i][j];
        }
    }
    printf("\n Sum of two matrices: \n");
```

```
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            printf(" %d",sum[i][j]);
        }
        printf("\n");
    }
}
```

Output:

Enter the number of rows : 2  
Enter the number of columns : 2  
Enter the elements of 1st matrices :-  
Enter the element of a[0 0] : 789  
Enter the element of a[0 1] : 456  
Enter the element of a[1 0] : 123  
Enter the element of a[1 1] : 000  
Enter the elements of 2nd matrices :-  
Enter the element of a[0 0] : 147  
Enter the element of a[0 1] : 258  
Enter the element of a[1 0] : 369  
Enter the element of a[1 1] : 159  
The elements of 1st matrices is :-  
789 456  
123 0

The elements of 2nd matrices is :-  
147 258  
369 159

Sum of two matrices:  
936 714  
492 159

// Print Program No. 13

// Write a program to find element at given position from 2-Dimensional array.

```
#include<stdio.h>
int main()
{
    int a[100][100], r, c, i, j, find, count=0;
    printf("\n Enter the number of rows : ");
    scanf("%d",&r);
    printf("\n Enter the number of columns : ");
    scanf("%d",&c);
    printf("\n Enter the elements of matrices :- ");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            printf("\n Enter the element of a[%d %d] : ",i,j);
            scanf("%d",&a[i][j]);
        }
    }

    printf("\n Enter the element you want to find position : ");
    scanf("%d",&find);

    for(i=0; i<r; i++)
    {
        for(j=0; j<c; j++)
        {
            if(a[i][j] == find)
            {
                printf("\n Item found at [%d, %d]", i, j);
                count++;
            }
        }
    }
    if(count==0)
        printf("\n Item Not found");
}
```

Output:

```
Enter the number of rows : 2
Enter the number of columns : 2
Enter the elements of matrices :-
Enter the element of a[0 0] : 00
Enter the element of a[0 1] : 11
Enter the element of a[1 0] : 22
Enter the element of a[1 1] : 33
Enter the element you want to find position : 22
Item found at [1, 0]
```

// Print Program No. 14

/\* Write a program that will read a text and count all occurrences of a particular character using function. \*/

```
#include <stdio.h>
int countOccurrences(const char *text, char target) {
    int count = 0;
    while (*text) {
        if (*text == target) {
            count++;
        }
        text++;
    }
    return count;
}

int main() {
    char text[200];
    char target;
    printf("Enter a text: ");
    fgets(text, sizeof(text), stdin);
    text[strcspn(text, "\n")] = '\0';
    printf("Enter the character to count: ");
    scanf(" %c", &target);
    int count = countOccurrences(text, target);
    printf("The character '%c' occurs %d times in the text.\n", target, count);
    return 0;
}
```

Output:

Enter a text: PGSCAS\_Sem1

Enter the character to count: S

The character 'S' occurs 3 times in the text.



// Print Program No. 15

// Write a function which returns 1 if the given number is palindrome otherwise returns 0.

```
#include <stdio.h>
int isPalindrome(int num) {
    int original = num;
    int reversed = 0;

    while (num > 0) {
        int digit = num % 10;
        reversed = reversed * 10 + digit;
        num /= 10;
    }

    return (original == reversed) ? 1 : 0;
}

int main() {
    int number;
    printf("Enter a number: ");
    scanf("%d", &number);
    if (isPalindrome(number)) {
        printf("%d is a palindrome.\n", number);
    } else {
        printf("%d is not a palindrome.\n", number);
    }

    return 0;
}
```

Output:

Enter a number: 4567

4567 is not a palindrome.

Output:

Enter a number: 424

424 is a palindrome.

```
// Print Program No. 16
// Write a recursive function for finding the factorial of a number.
```

```
#include <stdio.h>
int factorial(int n) {
    if (n <= 1) {
        return 1;
    }

    return n * factorial(n - 1);
}
int main() {
    int number;
    printf("Enter a positive integer: ");
    scanf("%d", &number);
    if (number < 0) {
        printf("Factorial is not defined for negative numbers.\n");
    } else {
        printf("The factorial of %d is %d.\n", number, factorial(number));
    }
    return 0;
}
```

Output:

```
Enter a positive integer: 12
The factorial of 12 is 479001600.
```

Output:

```
Enter a positive integer: 88
The factorial of 88 is 0.
```

// Print Program No. 17

// Write a program to perform summation of all elements of array using pointers.

```
#include <stdio.h>
int main() {
    int n, i, sum = 0;
    printf("Enter the number of elements in the array: ");
    scanf("%d", &n);
    int arr[n];
    printf("Enter %d elements: ", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    int *ptr = arr;
    for (i = 0; i < n; i++) {
        sum += *(ptr + i);
    }
    printf("Sum of the elements: %d\n", sum);

    return 0;
}
```

Output:

Enter the number of elements in the array: 4

Enter 4 elements: 11

22

33

44

Sum of the elements: 110

// Print Program No. 18

// Write a function using pointers to exchange the value stored in two locations in the memory.

```
#include <stdio.h>
void swap(int *a, int *b) {
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
}

int main() {
    int x,y;
    printf("Enter value for x: ");
    scanf("%d", &x);
    printf("Enter value for y: ");
    scanf("%d", &y);
    printf("Before swap: x = %d, y = %d\n", x, y);
    swap(&x, &y);
    printf("After swap: x = %d, y = %d\n", x, y);
    return 0;
}
```

Output:

Enter value for x: 23

Enter value for y: 32

Before swap: x = 23, y = 32

After swap: x = 32, y = 23

// Print Program No. 19

/\* Write a program to create structure Student with student's roll no, name and marks of three subjects (Maths, Science and English) and display the details of student with total marks of all subjects along with the percentage and passing class in proper format. \*/

```
#include <stdio.h>
struct Student {
    int roll_no;
    char name[50];
    float marks[3];
};
int main() {
    struct Student s;
    float total = 0.0, percentage;
    char *passing_class;
    int i;
    printf("Enter Roll No: ");
    scanf("%d", &s.roll_no);
    printf("Enter Name: ");
    scanf("%s", s.name);
    printf("Enter obtained marks in Maths: ");
    scanf("%f", &s.marks[0]);
    printf("Enter obtained marks in Science: ");
    scanf("%f", &s.marks[1]);
    printf("Enter obtained marks in English: ");
    scanf("%f", &s.marks[2]);
    for (i = 0; i < 3; i++) {
        total += s.marks[i];
    }
    percentage = total / 3;
    if (percentage >= 60) {
        passing_class = "First Class";
    } else if (percentage >= 50) {
        passing_class = "Second Class";
    } else if (percentage >= 40) {
        passing_class = "Third Class";
    } else {
        passing_class = "Fail";
    }
    printf("\nStudent Details:\n");
    printf("Roll No: %d\n", s.roll_no);
    printf("Name: %s\n", s.name);
    printf("Maths: %.2f\n", s.marks[0]);
    printf("Science: %.2f\n", s.marks[1]);
    printf("English: %.2f\n", s.marks[2]);
    printf("Total Marks: %.2f\n", total);
    printf("Percentage: %.2f%%\n", percentage);
    printf("Passing Class: %s\n", passing_class);
    return 0;
}
```

Enter Roll No: 17

Enter Name: Payal

Enter obtained marks in Maths: 98  
Enter obtained marks in Science: 86  
Enter obtained marks in English: 81

**Student Details:**

Roll No: 17

Name: Payal

Maths: 98.00

Science: 86.00

English: 81.00

Total Marks: 265.00

Percentage: 88.33%

Passing Class: First Class

Enter value for x: 23

Enter value for y: 32

Before swap:  $x = 23$ ,  $y = 32$

After swap:  $x = 32$ ,  $y = 23$

// Print Program No. 20

/\* Write a program to create structure Time (data members: int h, int m, int sec). Read a value as seconds from user to display new time after adding seconds to Time structure. \*/

```
#include <stdio.h>
struct Time {
    int h;
    int m;
    int sec;
};
void addSeconds(struct Time* time, int secondsToAdd) {
    time->sec += secondsToAdd;
    while (time->sec >= 60) {
        time->sec -= 60;
        time->m++;
    }
    while (time->m >= 60) {
        time->m -= 60;
        time->h++;
    }
    while (time->h >= 24) {
        time->h -= 24;
    }
}
int main() {
    struct Time currentTime;
    int secondsToAdd;
    printf("Enter current time (hours, minutes, seconds): ");
    scanf("%d %d %d", &currentTime.h, &currentTime.m, &currentTime.sec);
    printf("Enter seconds to add: ");
    scanf("%d", &secondsToAdd);
    addSeconds(&currentTime, secondsToAdd);
    printf("New time: %02d:%02d:%02d\n", currentTime.h, currentTime.m, currentTime.sec);
    return 0;
}
```

Output:

Enter current time (hours, minutes, seconds): 6 0 0  
Enter seconds to add: 60  
New time: 06:01:00

// Print Program No. 21

/\* Write a program to define a structure called book. Write a program to read information about 5 books and display books details in ascending order of price in proper format. \*/

```
#include <stdio.h>
#include <string.h>
struct Book {
    char title[50];
    char author[50];
    float price;
};
void sortBooksByPrice(struct Book books[], int n) {
    struct Book temp;
    int i,j;
    for (i = 0; i < n - 1; i++) {
        for (j = i + 1; j < n; j++) {
            if (books[i].price > books[j].price) {
                temp = books[i];
                books[i] = books[j];
                books[j] = temp;
            }
        }
    }
}
int main() {
    struct Book books[5];
    int n = 5,i;
    for (i = 0; i < n; i++) {
        printf("Enter the details of book no: %d\n", i + 1);
        printf("Title: ");
        scanf(" %[^\\n]s", books[i].title);
        printf("Author: ");
        scanf(" %[^\\n]s", books[i].author);
        printf("Price: ");
        scanf("%f", &books[i].price);
    }
    sortBooksByPrice(books, n);
    printf("\\nBooks in ascending order of price:\\n");
    for (i = 0; i < n; i++) {
        printf("Title: %s, Author: %s, Price: %.2f\\n", books[i].title, books[i].author, books[i].price);
    }
    return 0;
}
```

Output:

```
Enter the details of book no: 1
Title: Programming in ANSI C 9th Edition
Author: Balagurusamy
Price: 615
Enter the details of book no: 2
Title: Wise and Otherwise: A salute to Life
Author: Sudha Murty
```



Price: 190

Enter the details of book no: 3

Title: The Monk Who Sold His Ferrari

Author: Robin Sharma

Price: 261

Enter the details of book no: 4

Title: Atomic Habits

Author: James Clear

Price: 460

Enter the details of book no: 5

Title: The Alchemist

Author: Paulo Coelho

Price: 259

Books in ascending order of price:

Title: Wise and Otherwise: A salute to Life, Author: Sudha Murty, Price: 190.00

Title: The Alchemist, Author: Paulo Coelho, Price: 259.00

Title: The Monk Who Sold His Ferrari, Author: Robin Sharma, Price: 261.00

Title: Atomic Habits, Author: James Clear , Price: 460.00

Title: Programming in ANSI C 9th Edition, Author: Balagurusamy, Price: 615.00

// Print Program No. 22

/\* Write a program to copy the contents of one file to another and also print the no. of lines in the first file. \*/

```
#include <stdio.h>
#include <stdlib.h>
void copyFileAndCountLines(const char *sourceFile, const char *destFile) {
    FILE *src, *dest;
    char ch;
    int lineCount = 0;
    src = fopen(sourceFile, "r");
    if (src == NULL) {
        printf("Cannot open file %s \n", sourceFile);
        exit(1);
    }
    dest = fopen(destFile, "w");
    if (dest == NULL) {
        printf("Cannot open file %s \n", destFile);
        fclose(src);
        exit(1);
    }
    while ((ch = fgetc(src)) != EOF) {
        fputc(ch, dest);
        if (ch == '\n') {
            lineCount++;
        }
    }
    printf("Number of lines in the 1st file: %d\n", lineCount);
    fclose(src);
    fclose(dest);
}
int main() {
    char sourceFile[100], destFile[100];
    printf("Enter the source file name: ");
    scanf("%s", sourceFile);
    printf("Enter the destination file name: ");
    scanf("%s", destFile);
    copyFileAndCountLines(sourceFile, destFile);
    return 0;
}
```

Output:

Enter the source file name: example1.txt

Enter the destination file name: example2.txt

Number of lines in the 1st file: 2

// Print Program No. 23

/\* Write a function to read a file and count the no. of characters, spaces, newlines and no. of words in a given text file. \*/

```
#include <stdio.h>
#include <ctype.h>
void countFileContents(const char *filename) {
    FILE *file;
    char ch;
    int chars = 0, spaces = 0, newlines = 0, words = 0;
    int inWord = 0;

    file = fopen(filename, "r");
    if (file == NULL) {
        printf("Cannot open file %s \n", filename);
        return;
    }
    while ((ch = fgetc(file)) != EOF) {
        chars++;

        if (ch == ' ') {
            spaces++;
            inWord = 0;
        } else if (ch == '\n') {
            newlines++;
            inWord = 0;
        } else {
            if (!inWord) {
                words++;
                inWord = 1;
            }
        }
    }
    fclose(file);
    printf("Number of characters: %d\n", chars);
    printf("Number of words: %d\n", words);
    printf("Number of newlines: %d\n", newlines);
    printf("Number of spaces: %d\n", spaces);
}

int main() {
    char filename[100];
    printf("Enter the filename: ");
    scanf("%s", filename);
    countFileContents(filename);
    return 0;
}
```

Output:

Enter the filename: Example1.txt

Number of words: 27

Number of newlines: 1

Number of spaces: 26

// Print Program No. 24

/\* Write an interactive menu driven program that will access the data file created in the above problem to do one of the following tasks: a. Determine the telephone number of a specific customers. b. Determine the customer whose telephone no. is specified. c. Add a new record. d. Delete a record e. Generate the listing of all the customers and their telephone numbers. \*/

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define FILENAME "telephone_dairy.txt"
struct Customer {
    char name[50];
    char phone[15];
};
void addCustomer() {
    FILE *file = fopen(FILENAME, "a");
    struct Customer cust;
    if (file == NULL) {
        printf("Unable to open file.\n");
        return;
    }
    printf("Enter customer name: ");
    scanf("%[^\n]s", cust.name);
    printf("Enter telephone number: ");
    scanf("%[^\n]s", cust.phone);
    fprintf(file, "%s %s\n", cust.name, cust.phone);
    fclose(file);
    printf("Customer added successfully.\n");
}
void findPhoneByName() {
    FILE *file = fopen(FILENAME, "r");
    struct Customer cust;
    char name[50];
    int found = 0;
    if (file == NULL) {
        printf("Unable to open file.\n");
        return;
    }
    printf("Enter customer name: ");
    scanf("%[^\n]s", name);
    while (fscanf(file, "%s %s", cust.name, cust.phone) != EOF) {
        if (strcmp(cust.name, name) == 0) {
            printf("Telephone number of %s: %s\n", name, cust.phone);
            found = 1;
            break;
        }
    }
    if (!found) {
        printf("Customer not found.\n");
    }
    fclose(file);
}
void findNameByPhone() {
```

```
FILE *file = fopen(FILENAME, "r");
struct Customer cust;
char phone[15];
int found = 0;
if (file == NULL) {
    printf("Unable to open file.\n");
    return;
}
printf("Enter telephone number: ");
scanf("%[^\n]s", phone);
while (fscanf(file, "%s %s", cust.name, cust.phone) != EOF) {
    if (strcmp(cust.phone, phone) == 0) {
        printf("Customer with telephone number %s: %s\n", phone, cust.name);
        found = 1;
        break;
    }
}
if (!found) {
    printf("Telephone number not found.\n");
}
fclose(file);
}

void deleteCustomer() {
    FILE *file = fopen(FILENAME, "r");
    FILE *temp = fopen("temp.txt", "w");
    struct Customer cust;
    char name[50];
    int found = 0;
    if (file == NULL || temp == NULL) {
        printf("Unable to open file.\n");
        return;
    }
    printf("Enter customer name to delete: ");
    scanf("%[^\n]s", name);
    while (fscanf(file, "%s %s", cust.name, cust.phone) != EOF) {
        if (strcmp(cust.name, name) != 0) {
            fprintf(temp, "%s %s\n", cust.name, cust.phone);
        } else {
            found = 1;
        }
    }
    fclose(file);
    fclose(temp);
    remove(FILENAME);
    rename("temp.txt", FILENAME);
    if (found) {
        printf("Customer deleted successfully.\n");
    } else {
        printf("Customer not found.\n");
    }
}

void listAllCustomers() {
    FILE *file = fopen(FILENAME, "r");
```

```
struct Customer cust;
if (file == NULL) {
    printf("Unable to open file.\n");
    return;
}
printf("List of all customers and their telephone numbers:\n");
while (fscanf(file, "%s %s", cust.name, cust.phone) != EOF) {
    printf("Name: %s, Phone: %s\n", cust.name, cust.phone);
}
fclose(file);
}
int main() {
    int choice;
    while (1) {
        printf("\nMenu:\n");
        printf("1. Determine the telephone number of a specific customer\n");
        printf("2. Determine the customer whose telephone number is specified\n");
        printf("3. Add a new record\n");
        printf("4. Delete a record\n");
        printf("5. Generate listing of all customers and their telephone numbers\n");
        printf("6. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                findPhoneByName();
                break;
            case 2:
                findNameByPhone();
                break;
            case 3:
                addCustomer();
                break;
            case 4:
                deleteCustomer();
                break;
            case 5:
                listAllCustomers();
                break;
            case 6:
                exit(0);
            default:
                printf("Invalid choice.\n");
        }
    }
    return 0;
}
```

Output:

Menu:

1. Determine the telephone number of a specific customer
2. Determine the customer whose telephone number is specified
3. Add a new record

4. Delete a record
5. Generate listing of all customers and their telephone numbers
6. Exit

Enter your choice: 1

Enter customer name: Aruna

Telephone number of Aruna: 1234567890

Menu:

1. Determine the telephone number of a specific customer
2. Determine the customer whose telephone number is specified
3. Add a new record
4. Delete a record
5. Generate listing of all customers and their telephone numbers
6. Exit

Enter your choice: 5

List of all customers and their telephone numbers:

Name: Aruna, Phone: 1234567890

Name: Ankush, Phone: 1133557799

Name: Annu, Phone: 2244668800

Name: Aayushi, Phone: 2143658709

Menu:

1. Determine the telephone number of a specific customer
2. Determine the customer whose telephone number is specified
3. Add a new record
4. Delete a record
5. Generate listing of all customers and their telephone numbers
6. Exit

Enter your choice: 3

Enter customer name: Aaku

Enter telephone number: 3216549870

Customer added successfully.

Menu:

1. Determine the telephone number of a specific customer
2. Determine the customer whose telephone number is specified
3. Add a new record
4. Delete a record
5. Generate listing of all customers and their telephone numbers
6. Exit

Enter your choice: 4

Enter customer name to delete: Aaku

Customer deleted successfully.

```
// Print Program No. 25
```

```
/* Use a structure of Employee to write records of employee to a file. Include a menu that will allow the user to select any of the following feature's a. Add a new record. b. Delete a record. c. Modify an existing record. d. Retrieve and display an entire record for a given ID/Name. e. Generate a complete list of all employee names, addresses and telephone numbers. f. End of the computation/Exit. */
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX_EMPLOYEES 100
#define FILENAME "employee_details.txt"
```

```
struct Employee {
    int id;
    char name[30];
    char address[100];
    char phone[10];
};
```

```
void addRecord();
void deleteRecord();
void modifyRecord();
void retrieveRecord();
void listRecords();
void clearBuffer();
```

```
int main() {
    int choice;

    do {
        printf("\nEmployee Management System:\n");
        printf("1. Add a new record.\n");
        printf("2. Delete a record.\n");
        printf("3. Modify an existing record.\n");
        printf("4. Retrieve and display a record.\n");
        printf("5. Generate a complete list of employees.\n");
        printf("6. Exit.\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        clearBuffer();

        switch (choice) {
            case 1: addRecord(); break;
            case 2: deleteRecord(); break;
            case 3: modifyRecord(); break;
            case 4: retrieveRecord(); break;
            case 5: listRecords(); break;
            case 6: printf("Exiting...\n"); break;
            default: printf("Invalid choice! Please try again.\n");
        }
    } while (choice != 6);
}
```



```
    return 0;
}

void clearBuffer() {
    while (getchar() != '\n');
}

void addRecord() {
    struct Employee emp;
    FILE *file = fopen(FILENAME, "ab");

    if (file == NULL) {
        printf("Error opening file!\n");
        return;
    }

    printf("Enter ID: ");
    scanf("%d", &emp.id);
    clearBuffer();
    printf("Enter Name: ");
    fgets(emp.name, sizeof(emp.name), stdin);
    strtok(emp.name, "\n");
    printf("Enter Address: ");
    fgets(emp.address, sizeof(emp.address), stdin);
    strtok(emp.address, "\n");
    printf("Enter Phone: ");
    fgets(emp.phone, sizeof(emp.phone), stdin);
    strtok(emp.phone, "\n");

    fwrite(&emp, sizeof(struct Employee), 1, file);
    fclose(file);
    printf("Record added successfully.\n");
}

void deleteRecord() {
    int id;
    struct Employee emp;
    FILE *file = fopen(FILENAME, "rb");
    FILE *tempFile = fopen("temp.dat", "wb");

    if (file == NULL || tempFile == NULL) {
        printf("Error opening file!\n");
        return;
    }

    printf("Enter ID of the employee to delete: ");
    scanf("%d", &id);

    int found = 0;
    while (fread(&emp, sizeof(struct Employee), 1, file)) {
        if (emp.id != id) {
            fwrite(&emp, sizeof(struct Employee), 1, tempFile);
        }
    }
}
```

```
    } else {
        found = 1;
    }
}

fclose(file);
fclose(tempFile);
remove(FILENAME);
rename("priyanka10_22.dat", FILENAME);

if (found) {
    printf("Record deleted successfully.\n");
} else {
    printf("Record with ID %d not found.\n", id);
}
}

void modifyRecord() {
    int id;
    struct Employee emp;
    FILE *file = fopen(FILENAME, "r+b");

    if (file == NULL) {
        printf("Error opening file!\n");
        return;
    }

    printf("Enter ID of the employee to modify: ");
    scanf("%d", &id);

    int found = 0;
    while (fread(&emp, sizeof(struct Employee), 1, file)) {
        if (emp.id == id) {
            found = 1;
            printf("Enter new Name: ");
            clearBuffer();
            fgets(emp.name, sizeof(emp.name), stdin);
            strtok(emp.name, "\n"); // Remove newline character
            printf("Enter new Address: ");
            fgets(emp.address, sizeof(emp.address), stdin);
            strtok(emp.address, "\n"); // Remove newline character
            printf("Enter new Phone: ");
            fgets(emp.phone, sizeof(emp.phone), stdin);
            strtok(emp.phone, "\n"); // Remove newline character

            fseek(file, -sizeof(struct Employee), SEEK_CUR); // Move to the record's position
            fwrite(&emp, sizeof(struct Employee), 1, file);
            break;
        }
    }

    fclose(file);
```

```
    if (found) {
        printf("Record modified successfully.\n");
    } else {
        printf("Record with ID %d not found.\n", id);
    }
}

void retrieveRecord() {
    int id;
    struct Employee emp;
    FILE *file = fopen(FILENAME, "rb");

    if (file == NULL) {
        printf("Error opening file!\n");
        return;
    }

    printf("Enter ID of the employee to retrieve: ");
    scanf("%d", &id);

    int found = 0;
    while (fread(&emp, sizeof(struct Employee), 1, file)) {
        if (emp.id == id) {
            found = 1;
            printf("ID: %d\n", emp.id);
            printf("Name: %s\n", emp.name);
            printf("Address: %s\n", emp.address);
            printf("Phone: %s\n", emp.phone);
            break;
        }
    }

    fclose(file);

    if (!found) {
        printf("Record with ID %d not found.\n", id);
    }
}

void listRecords() {
    struct Employee emp;
    FILE *file = fopen(FILENAME, "rb");

    if (file == NULL) {
        printf("Error opening file!\n");
        return;
    }

    printf("\nList of Employees:\n");
    while (fread(&emp, sizeof(struct Employee), 1, file)) {
        printf("ID: %d, Name: %s, Address: %s, Phone: %s\n", emp.id, emp.name, emp.address,
emp.phone);
    }
}
```

```
    fclose(file);  
}
```

Output:

Employee Management System:

1. Add a new record.
2. Delete a record.
3. Modify an existing record.
4. Retrieve and display a record.
5. Generate a complete list of employees.
6. Exit.

Enter your choice: 1

Enter ID: 005

Enter Name: Aaku

Enter Address: Ahmedabad

Enter Phone: 1236547890

Record added successfully.

Employee Management System:

1. Add a new record.
2. Delete a record.
3. Modify an existing record.
4. Retrieve and display a record.
5. Generate a complete list of employees.
6. Exit.

Enter your choice: 2

Enter ID of the employee to delete: 005

Record deleted successfully.

Employee Management System:

1. Add a new record.
2. Delete a record.
3. Modify an existing record.
4. Retrieve and display a record.
5. Generate a complete list of employees.
6. Exit.

Enter your choice: 3

Enter ID of the employee to modify: 001

Enter new Name: Aruna

Enter new Address: Ahmedabad

Enter new Phone: 1234567890

Record modified successfully.

Employee Management System:

1. Add a new record.
2. Delete a record.
3. Modify an existing record.
4. Retrieve and display a record.
5. Generate a complete list of employees.
6. Exit.

Enter your choice: 4

Enter ID of the employee to retrieve: 004

ID: 4

Name: Aayushi

Address: Vadodra

Phone: 2143658709

Employee Management System:

1. Add a new record.
2. Delete a record.
3. Modify an existing record.
4. Retrieve and display a record.
5. Generate a complete list of employees.
6. Exit.

Enter your choice: 5

List of Employees:

ID: 1, Name: Aruna, Address: Ahmedabad, Phone: 1234567890

ID: 2, Name: Ankush, Address: Gandhinagar, Phone: 1133557799

ID: 3, Name: Annu, Address: Surat, Phone: 2244668800

Employee Management System:

1. Add a new record.
2. Delete a record.
3. Modify an existing record.
4. Retrieve and display a record.
5. Generate a complete list of employees.
6. Exit.

Enter your choice: 6