Name: Chinmay A. Kulkarni

SRN: PES1UG19CS127

Roll number:46

**WEEK 11:**

1)

**client.c:**

```
#include <stdio.h>

#include "week_11_1_server.h"
int main()
{
    struct date d_1, d_2;
    date_compare(d_1, d_2);
    return 0;
}
```

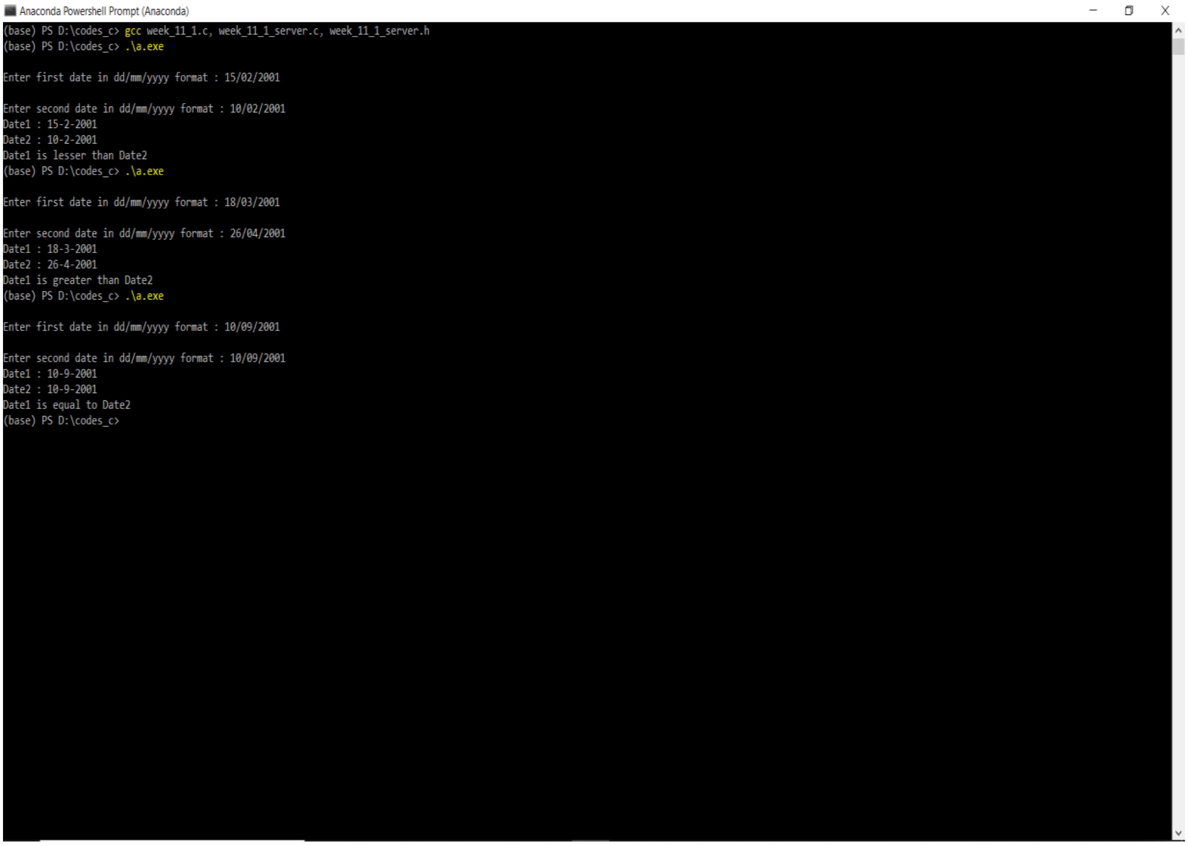
**server.h:**

```
struct date

{
    int dd, month, year;
};
void date_compare(struct date d_1, struct date d_2);
```

**server.c:**

```
#include <stdio.h>

#include <stdlib.h>
#include "week_11_1_server.h"
void date_compare(struct date d_1, struct date d_2)
{
    printf("\nEnter first date in dd/mm/yyyy format : ");
    scanf("%d/%d/%d", &d_1.dd, &d_1.month, &d_1.year);
    printf("\nEnter second date in dd/mm/yyyy format : ");
    scanf("%d/%d/%d", &d_2.dd, &d_2.month, &d_2.year);
    printf("Date1 : %d-%d-%d", d_1.dd, d_1.month, d_1.year);
    printf("\nDate2 : %d-%d-%d", d_2.dd, d_2.month, d_2.year);
    if (d_1.year == d_2.year && d_1.month == d_2.month && d_1.dd == d_2.dd)
        printf("\nDate1 is equal to Date2");
    else if ((d_1.year < d_2.year) || (d_1.year == d_2.year && d_1.month < d_2.month) || (d_1.year == d_2.year && d_1.month == d_2.month && d_1.dd < d_2.dd))
        printf("\nDate1 is greater than Date2");
    else if ((d_1.year > d_2.year) || (d_1.year == d_2.year && d_1.month > d_2.month) || (d_1.year == d_2.year && d_1.month == d_2.month && d_1.dd > d_2.dd))
        printf("\nDate1 is lesser than Date2");
}
```

**Output:**



```
(base) PS D:\codes_c> gcc week_11_1.c, week_11_1_server.c, week_11_1_server.h
(base) PS D:\codes_c> .\a.exe

Enter first date in dd/mm/yyyy format : 15/02/2001

Enter second date in dd/mm/yyyy format : 10/02/2001
Date1 : 15-2-2001
Date2 : 10-2-2001
Date1 is lesser than Date2
(base) PS D:\codes_c> .\a.exe

Enter first date in dd/mm/yyyy format : 18/03/2001

Enter second date in dd/mm/yyyy format : 26/04/2001
Date1 : 18-3-2001
Date2 : 26-4-2001
Date1 is greater than Date2
(base) PS D:\codes_c> .\a.exe

Enter first date in dd/mm/yyyy format : 10/09/2001

Enter second date in dd/mm/yyyy format : 10/09/2001
Date1 : 10-9-2001
Date2 : 10-9-2001
Date1 is equal to Date2
(base) PS D:\codes_c>
```

2)

**client.c:**

```
#include <stdio.h>

#include <stdlib.h>
#include "week_11_2_server.h"
int main()
{
    int n;
    printf("\nEnter the number of students : ");
    scanf("%d", &n);
    struct student *s = (struct student *)malloc(n * sizeof(struct student));
    for (int i = 0; i < n; i++)
    {
        printf("\nEnter Student %d details : ", i + 1);
        read_student_details(s + i);
    }
    for (int i = 0; i < n; i++)
    {
        printf("\n\nStudent details : ");
        print_student_details(s + i);
    }
    free(s);
    free(s->name);
    return 0;
}
```

**server.h:**

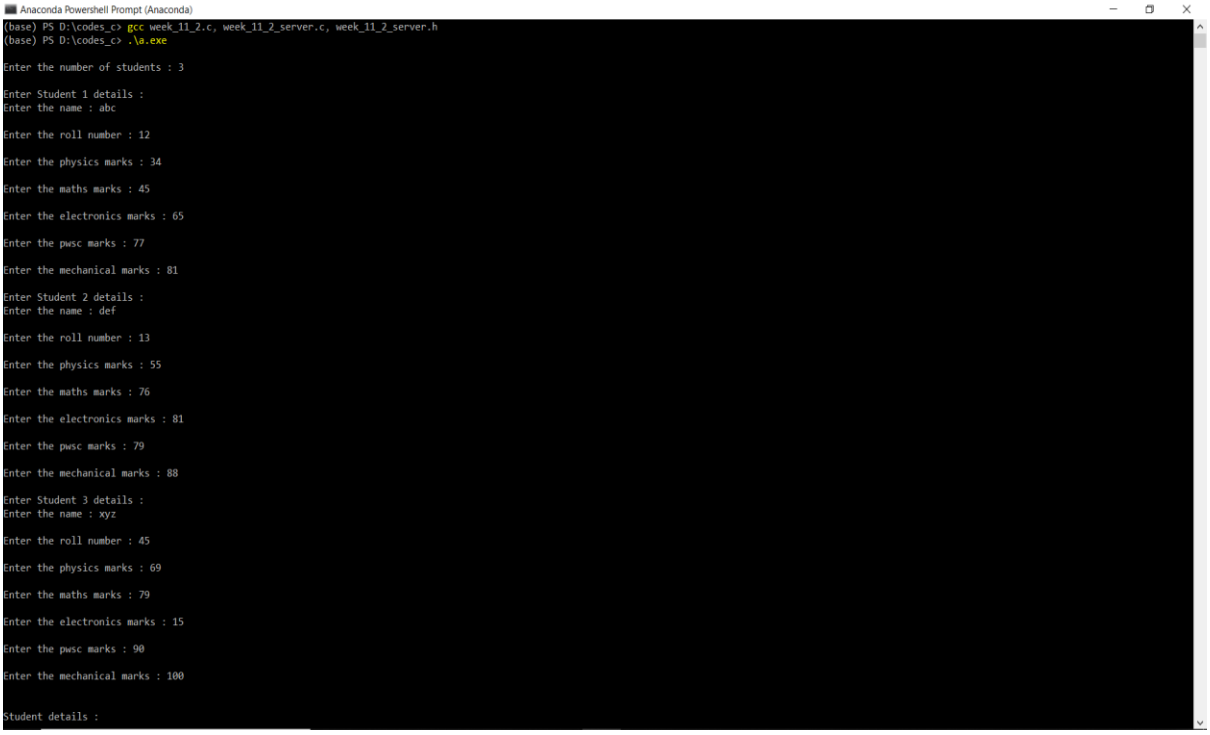
```
struct student

{
    char *name;
    int roll_num;
    float phy_marks, math_marks, ec_marks, pswc_marks, mech_marks, tota
l, average;
};
void read_student_details(struct student *s);
void print_student_details(struct student *s);
```

**server.c:**

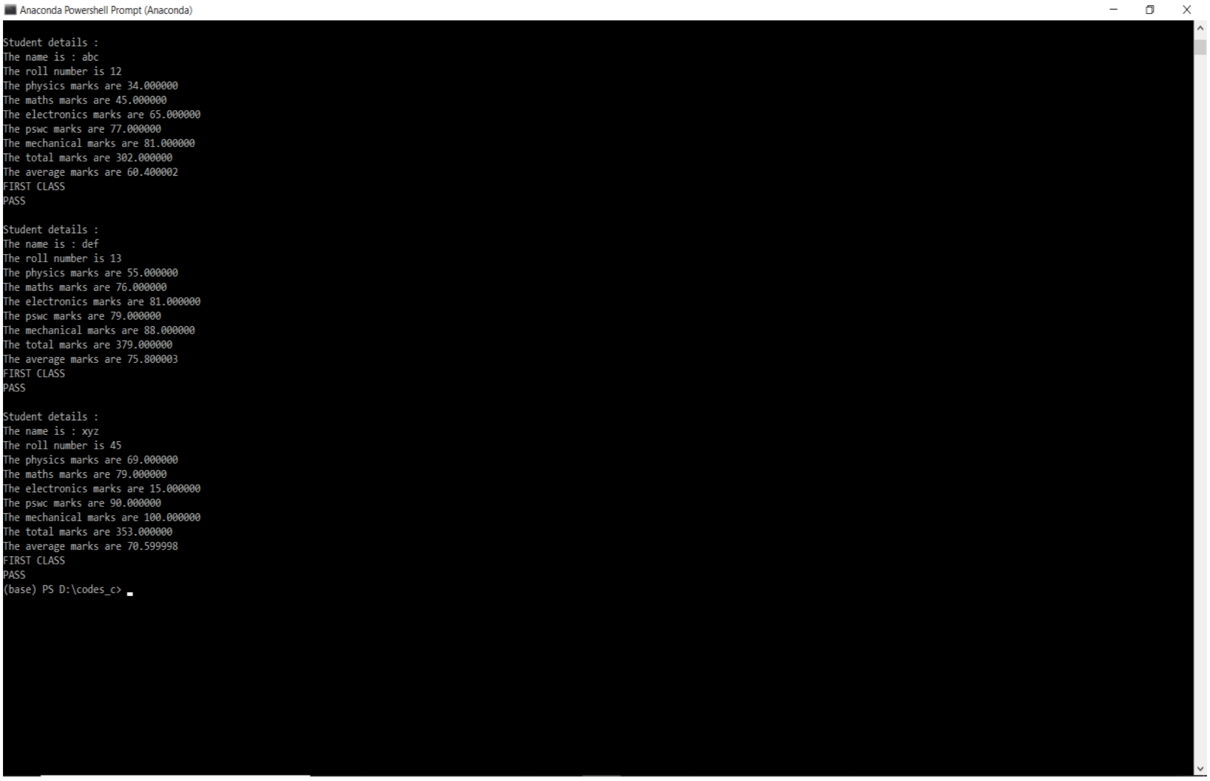
```
#include <stdio.h>

#include <stdlib.h>
#include "week_11_2_server.h"
void read_student_details(struct student *s)
{
    s->name = (char *)malloc(25 * sizeof(char *));

    printf("\nEnter the name : ");
    scanf("%s", s->name);
    printf("\nEnter the roll number : ");
    scanf("%d", &s->roll_num);
    printf("\nEnter the physics marks : ");
    scanf("%f", &s->phy_marks);
    printf("\nEnter the maths marks : ");
    scanf("%f", &s->math_marks);
    printf("\nEnter the electronics marks : ");
    scanf("%f", &s->ec_marks);
    printf("\nEnter the pwsc marks : ");
    scanf("%f", &s->pswc_marks);
    printf("\nEnter the mechanical marks : ");
    scanf("%f", &s->mech_marks);
```

```c
}
void print_student_details(struct student *s)
{
    printf("\nThe name is : %s", s->name);
    printf("\nThe roll number is %d", s->roll_num);
    printf("\nThe physics marks are %f", s->phy_marks);
    printf("\nThe maths marks are %f", s->math_marks);
    printf("\nThe electronics marks are %f", s->ec_marks);
    printf("\nThe pswc marks are %f", s->pswc_marks);
    printf("\nThe mechanical marks are %f", s->mech_marks);
    (s->total) = (s->phy_marks) + (s->math_marks) + (s->ec_marks) + (s-
>pswc_marks) + (s->mech_marks);
    (s->average) = (s->total) / 5;
    printf("\nThe total marks are %f", (s->total));
    printf("\nThe average marks are %f", (s->average));
    if (85 <= (s->average) && (s->average) <= 100)
        printf("\nFIRST CLASS WITH DISTINCTION\nPASS");
    else if (60 <= (s->average) && (s->average) <= 84)
        printf("\nFIRST CLASS\nPASS");
    else if (50 <= (s->average) && (s->average) <= 59)
        printf("\nSECOND CLASS\nPASS");
    else if (40 <= (s->average) && (s->average) <= 49)
        printf("\nTHIRD CLASS\nPASS");
    if ((s->average) < 49)
        printf("\nF GRADE\nPASS");
}
```

**Output:**



```
(base) PS D:\codes_c> gcc week_11_2.c, week_11_2_server.c, week_11_2_server.h
(base) PS D:\codes_c> .\a.exe

Enter the number of students : 3

Enter Student 1 details :
Enter the name : abc

Enter the roll number : 12

Enter the physics marks : 34

Enter the maths marks : 45

Enter the electronics marks : 65

Enter the pwsc marks : 77

Enter the mechanical marks : 81

Enter Student 2 details :
Enter the name : def

Enter the roll number : 13

Enter the physics marks : 55

Enter the maths marks : 76

Enter the electronics marks : 81

Enter the pwsc marks : 79

Enter the mechanical marks : 88

Enter Student 3 details :
Enter the name : xyz

Enter the roll number : 45

Enter the physics marks : 69

Enter the maths marks : 79

Enter the electronics marks : 15

Enter the pwsc marks : 90

Enter the mechanical marks : 100


Student details :
```



```
Student details :
The name is : abc
The roll number is 12
The physics marks are 34.000000
The maths marks are 45.000000
The electronics marks are 65.000000
The pswc marks are 77.000000
The mechanical marks are 81.000000
The total marks are 302.000000
The average marks are 60.400002
FIRST CLASS
PASS

Student details :
The name is : def
The roll number is 13
The physics marks are 55.000000
The maths marks are 76.000000
The electronics marks are 81.000000
The pswc marks are 79.000000
The mechanical marks are 88.000000
The total marks are 379.000000
The average marks are 75.800003
FIRST CLASS
PASS

Student details :
The name is : xyz
The roll number is 45
The physics marks are 69.000000
The maths marks are 79.000000
The electronics marks are 15.000000
The pswc marks are 90.000000
The mechanical marks are 100.000000
The total marks are 353.000000
The average marks are 70.599998
FIRST CLASS
PASS
(base) PS D:\codes_c>
```

3)

**client.c:**

```
#include <stdio.h>

#include "week_11_3_server.h"
int main()
{
    struct feet_inch s_1, s_2;
    distance_add(s_1, s_2);
    return 0;
}
```

**server.h:**

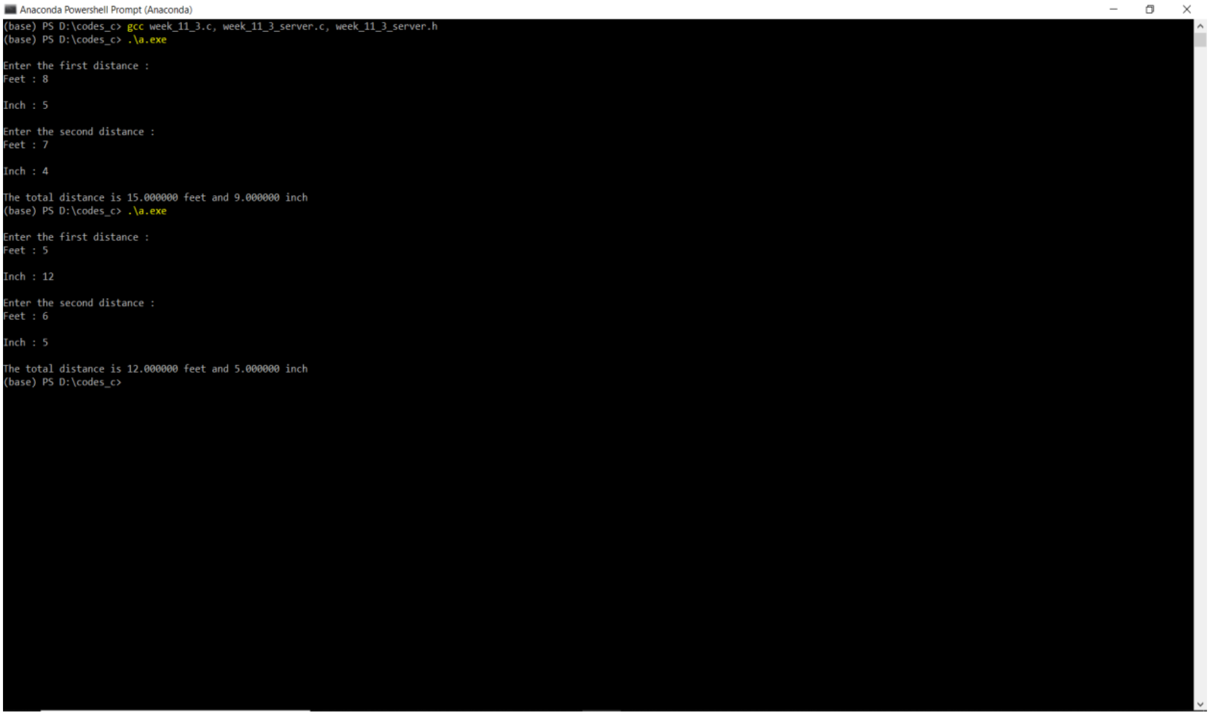
```
struct feet_inch

{
    float feet;
    float inch;
};
void distance_add(struct feet_inch s_1, struct feet_inch s_2);
```

**server.c:**

```
#include <stdio.h>

#include "week_11_3_server.h"
```

```c
void distance_add(struct feet_inch s_1, struct feet_inch s_2)
{
    printf("\nEnter the first distance : ");
    printf("\nFeet : ");
    scanf("%f", &s_1.feet);
    printf("\nInch : ");
    scanf("%f", &s_1.inch);
    printf("\nEnter the second distance : ");
    printf("\nFeet : ");
    scanf("%f", &s_2.feet);
    printf("\nInch : ");
    scanf("%f", &s_2.inch);
    if (s_1.inch + s_2.inch >= 12)
        printf("\nThe total distance is %f feet and %f inch", s_1.feet
+ s_2.feet + 1, s_1.inch + s_2.inch - 12);
    else
        printf("\nThe total distance is %f feet and %f inch", s_1.feet
+ s_2.feet, s_1.inch + s_2.inch);
}
```

**Output:**



```
(base) PS D:\codes_c> gcc week_11_3.c, week_11_3_server.c, week_11_3_server.h
(base) PS D:\codes_c> .\a.exe

Enter the first distance :
Feet : 8

Inch : 5

Enter the second distance :
Feet : 7

Inch : 4

The total distance is 15.000000 feet and 9.000000 inch
(base) PS D:\codes_c> .\a.exe

Enter the first distance :
Feet : 5

Inch : 12

Enter the second distance :
Feet : 6

Inch : 5

The total distance is 12.000000 feet and 5.000000 inch
(base) PS D:\codes_c>
```

4)

**client.c:**

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

```
#include "week_11_4_server.h"
int main()
{
    struct complex s_1, s_2;
    complex_add(s_1, s_2);
    complex_subtract(s_1, s_2);
    complex_multiply(s_1, s_2);
    return 0;
}
```

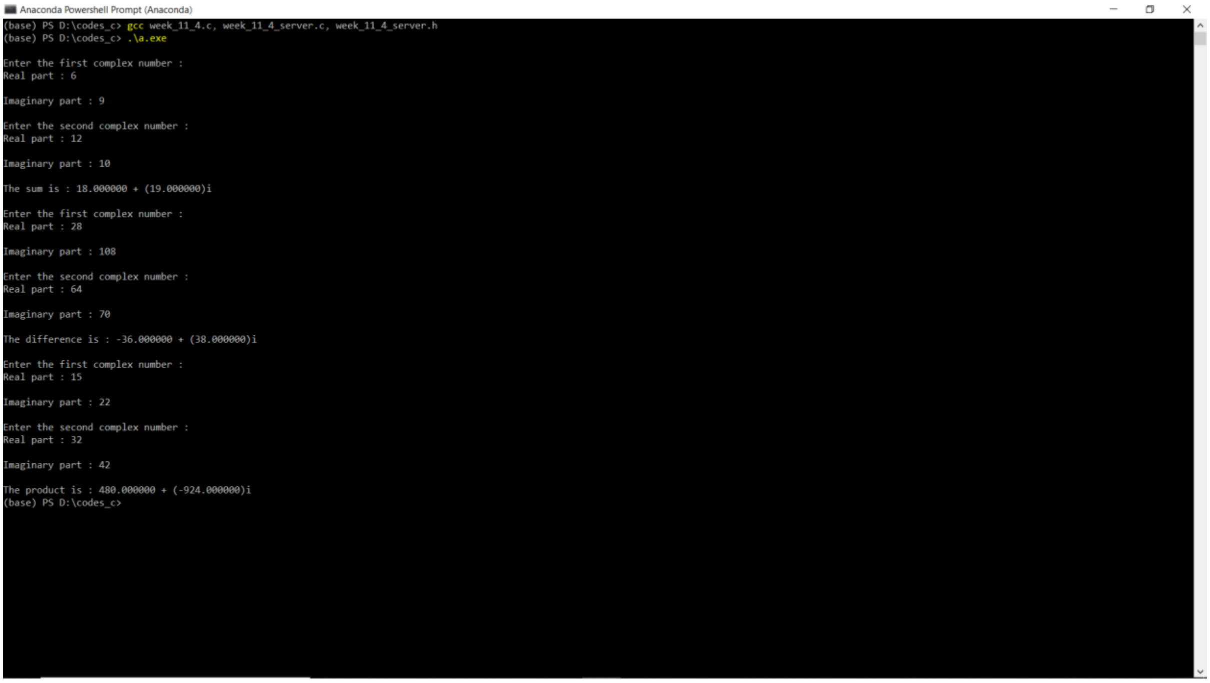
**server.h:**

```
struct complex

{
    float real;
    float im;
};
void complex_add(struct complex s_1, struct complex s_2);
void complex_subtract(struct complex s_1, struct complex s_2);
void complex_multiply(struct complex s_1, struct complex s_2);
```

**server.c:**

```
#include <stdio.h>

#include "week_11_4_server.h"
void complex_add(struct complex s_1, struct complex s_2)
{
    printf("\nEnter the first complex number : ");
    printf("\nReal part : ");
    scanf("%f", &s_1.real);
    printf("\nImaginary part : ");
    scanf("%f", &s_1.im);
    printf("\nEnter the second complex number : ");
    printf("\nReal part : ");
    scanf("%f", &s_2.real);
    printf("\nImaginary part : ");
    scanf("%f", &s_2.im);
    printf("\nThe sum is : %f + (%f)i", s_1.real + s_2.real, s_1.im + s
_2.im);
}
void complex_subtract(struct complex s_1, struct complex s_2)
{
    printf("\n\nEnter the first complex number : ");
    printf("\nReal part : ");
    scanf("%f", &s_1.real);
    printf("\nImaginary part : ");
    scanf("%f", &s_1.im);
    printf("\nEnter the second complex number : ");
    printf("\nReal part : ");
    scanf("%f", &s_2.real);
    printf("\nImaginary part : ");
    scanf("%f", &s_2.im);
    printf("\nThe difference is : %f + (%f)i", s_1.real - s_2.real, s_1
.im - s_2.im);
}
void complex_multiply(struct complex s_1, struct complex s_2)
{
    printf("\n\nEnter the first complex number : ");
    printf("\nReal part : ");
    scanf("%f", &s_1.real);
    printf("\nImaginary part : ");
    scanf("%f", &s_1.im);
    printf("\nEnter the second complex number : ");
    printf("\nReal part : ");
    scanf("%f", &s_2.real);
    printf("\nImaginary part : ");
    scanf("%f", &s_2.im);
    printf("\nThe product is : %f + (%f)i", s_1.real * s_2.real, s_1.im
 * s_2.im * -1);
}
```

**Output:**



```
(base) PS D:\codes_c> gcc week_11_4.c, week_11_4_server.c, week_11_4_server.h
(base) PS D:\codes_c> .\a.exe

Enter the first complex number :
Real part : 6

Imaginary part : 9

Enter the second complex number :
Real part : 12

Imaginary part : 10

The sum is : 18.000000 + (19.000000)i

Enter the first complex number :
Real part : 28

Imaginary part : 108

Enter the second complex number :
Real part : 64

Imaginary part : 70

The difference is : -36.000000 + (38.000000)i

Enter the first complex number :
Real part : 15

Imaginary part : 22

Enter the second complex number :
Real part : 32

Imaginary part : 42

The product is : 480.000000 + (-924.000000)i
(base) PS D:\codes_c>
```

5)

**client.c:**

```
#include <stdio.h>

#include "week_11_5_server.h"
#include <stdlib.h>
int main()
{
    int n, total_bill = 0;
    printf("\nEnter the number of products : ");
    scanf("%d", &n);
    struct product *s = (struct product *)malloc(n * sizeof(struct prod
uct));
    for (int i = 0; i < n; i++)
    {
        printf("\nEnter product %d details : ", i + 1);
        bill(s + i);
    }
    for (int i = 0; i < n; i++)
    {
        printf("\n\nProduct details : ");
        print_bill(s + i);
    }
    for (int i = 0; i < n; i++)
        total_bill += ((s + i)->cost) * ((s + i)->quantity);
    printf("\nThe total bill is %d", total_bill);
    free(s);
    free(s->name);
    return 0;
}
```

**server.h:**

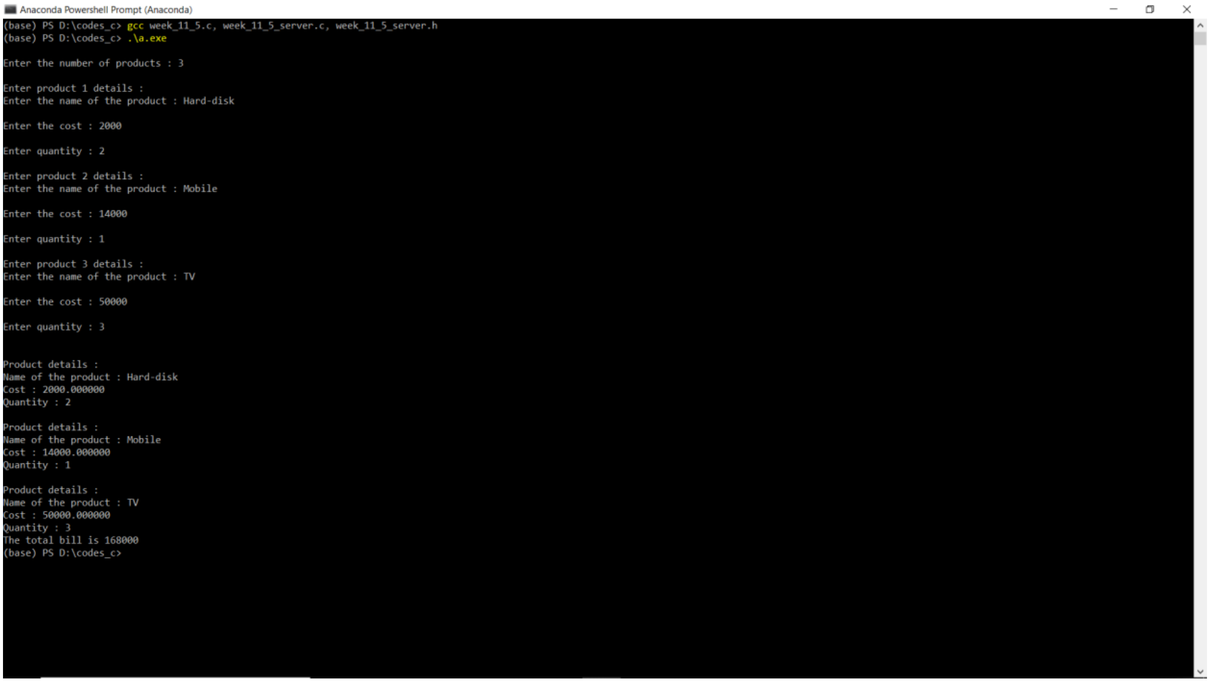
```
struct product

{
    char *name;
    float cost;
    int quantity;
};
```

**server.c:**

```
void bill(struct product *s);
void print_bill(struct product *s);
#include <stdio.h>

#include <stdlib.h>
```

```c
#include "week_11_5_server.h"
void bill(struct product *s)
{
    s->name = (char *)malloc(30 * sizeof(char));
    printf("\nEnter the name of the product : ");
    scanf("%s", s->name);
    printf("\nEnter the cost : ");
    scanf("%f", &s->cost);
    printf("\nEnter quantity : ");
    scanf("%d", &s->quantity);
}
void print_bill(struct product *s)
{
    printf("\nName of the product : %s", s->name);
    printf("\nCost : %f", s->cost);
    printf("\nQuantity : %d", s->quantity);
}
```

**Output:**



```
(base) PS D:\codes_c> gcc week_11_5.c, week_11_5_server.c, week_11_5_server.h
(base) PS D:\codes_c> .\a.exe

Enter the number of products : 3

Enter product 1 details :
Enter the name of the product : Hard-disk

Enter the cost : 2000

Enter quantity : 2

Enter product 2 details :
Enter the name of the product : Mobile

Enter the cost : 14000

Enter quantity : 1

Enter product 3 details :
Enter the name of the product : TV

Enter the cost : 50000

Enter quantity : 3


Product details :
Name of the product : Hard-disk
Cost : 2000.000000
Quantity : 2

Product details :
Name of the product : Mobile
Cost : 14000.000000
Quantity : 1

Product details :
Name of the product : TV
Cost : 50000.000000
Quantity : 3
The total bill is 168000
(base) PS D:\codes_c>
```

**WEEK 12:**

1)

**client.c:**

```
#include <stdio.h>

#include "week_12_1_server.h"
#include <stdlib.h>
int main()
{
    int n, *p;
    printf("\nEnter the number of elements : ");
    scanf("%d", &n);
    p = (int *)malloc(n * sizeof(n));
    array(p, n);
    return 0;
}
```

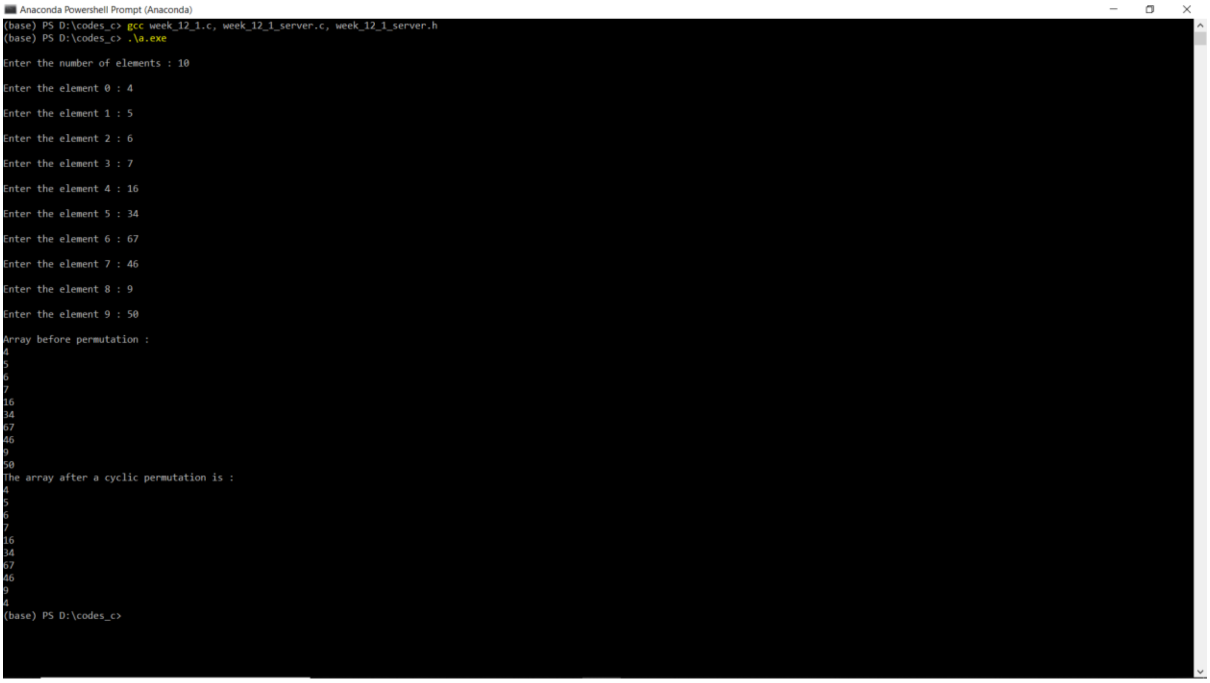
**server.h:**

```
void array(int *s, int n);
```

**server.c:**

```
#include <stdio.h>

#include "week_12_1_server.h"
void array(int *s, int n)
{
    for (int i = 0; i < n; i++)
    {
        printf("\nEnter the element %d : ", i);
        scanf("%d", s + i);
    }
    printf("\nArray before permutation : ");
    for (int i = 0; i < n; i++)
        printf("\n%d", *(s + i));
    printf("\nThe array after a cyclic permutation is : ");
    *(s + n - 1) = *s;
    for (int i = 0; i < n - 1; i++)
        *s = *(s + i + 1);
    for (int i = 0; i < n; i++)
        printf("\n%d", *(s + i));
}
```

**Output:**



Anaconda Powershell Prompt (Anaconda)

```
(base) PS D:\codes_c> gcc week_12_1.c, week_12_1_server.c, week_12_1_server.h
(base) PS D:\codes_c> .\a.exe

Enter the number of elements : 10

Enter the element 0 : 4

Enter the element 1 : 5

Enter the element 2 : 6

Enter the element 3 : 7

Enter the element 4 : 16

Enter the element 5 : 34

Enter the element 6 : 67

Enter the element 7 : 46

Enter the element 8 : 9

Enter the element 9 : 50

Array before permutation :
4
5
6
7
16
34
67
46
9
50
The array after a cyclic permutation is :
4
5
6
7
16
34
67
46
9
4
(base) PS D:\codes_c>
```

2)

**client.c:**

```c
#include <stdio.h>

#include <stdlib.h>
#include "week_12_2_server.h"
int main()
{
    int n;
    printf("\nEnter the number of employees : ");
    scanf("%d", &n);
    struct employee *p = (struct employee *)malloc(n * sizeof(struct employee));
    for (int i = 0; i < n; i++)
    {
        printf("\nEnter the details of employee %d : ", i + 1);
        read_details(p + i);
    }
    for (int i = 0; i < n; i++)
    {
        printf("\nThe details of employee %d : ", i + 1);
        print_details(p + i);
        printf("\n");
    }
    free(p->name);
    free(p->dep);
    free(p->emp_id);
    return 0;
}
```

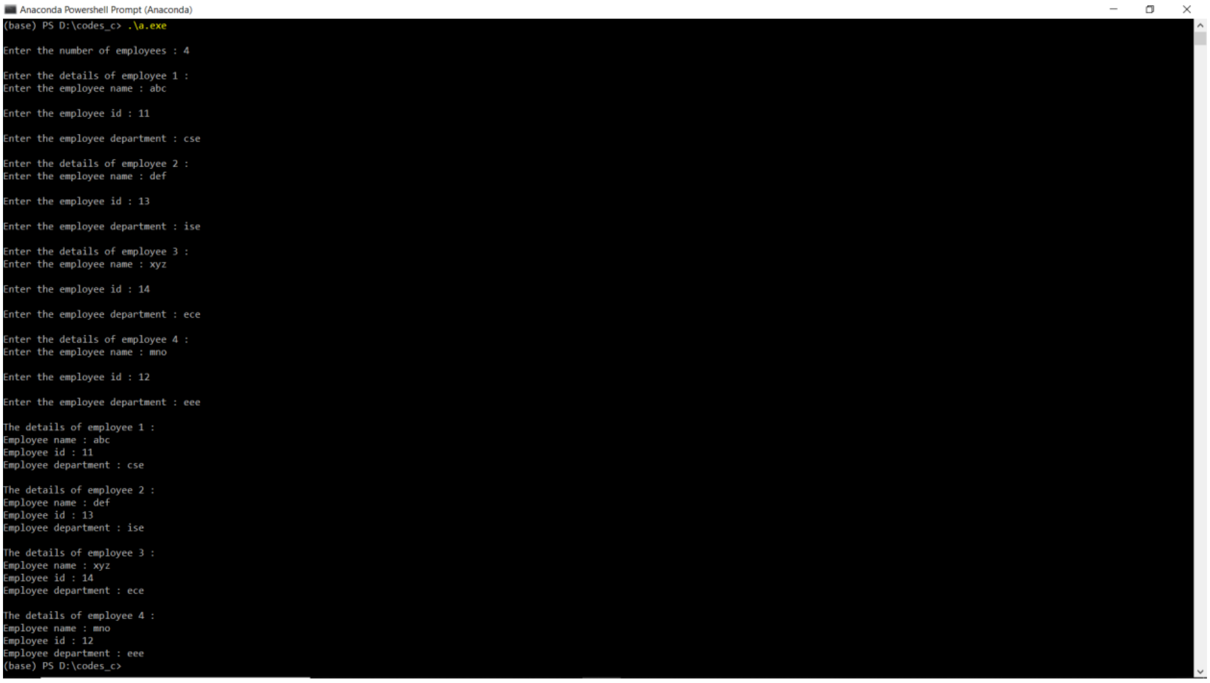
**server.h:**

```
struct employee

{
    char *emp_id, *name, *dep;
};
void read_details(struct employee *s);
void print_details(struct employee *s);
```

**server.c:**

```
#include <stdio.h>

#include <stdlib.h>
#include "week_12_2_server.h"
void read_details(struct employee *s)
{
    s->name = (char *)malloc(20*sizeof(char));
    s->emp_id = (char *)malloc(10*sizeof(char));
    s->dep = (char *)malloc(30 * sizeof(char));
    printf("\nEnter the employee name : ");
    scanf("%s", s->name);
    printf("\nEnter the employee id : ");
    scanf("%s", s->emp_id);
    printf("\nEnter the employee department : ");
    scanf("%s", s->dep);
}
void print_details(struct employee *s)
{
    printf("\nEmployee name : %s", s->name);
    printf("\nEmployee id : %s", s->emp_id);
    printf("\nEmployee department : %s", s->dep);
}
```

## Output:



```
(base) PS D:\codes_c> .\a.exe

Enter the number of employees : 4

Enter the details of employee 1 :
Enter the employee name : abc

Enter the employee id : 11

Enter the employee department : cse

Enter the details of employee 2 :
Enter the employee name : def

Enter the employee id : 13

Enter the employee department : ise

Enter the details of employee 3 :
Enter the employee name : xyz

Enter the employee id : 14

Enter the employee department : ece

Enter the details of employee 4 :
Enter the employee name : mno

Enter the employee id : 12

Enter the employee department : eee

The details of employee 1 :
Employee name : abc
Employee id : 11
Employee department : cse

The details of employee 2 :
Employee name : def
Employee id : 13
Employee department : ise

The details of employee 3 :
Employee name : xyz
Employee id : 14
Employee department : ece

The details of employee 4 :
Employee name : mno
Employee id : 12
Employee department : eee
(base) PS D:\codes_c>
```

3)

## client.c:

```c
#include <stdio.h>

#include <stdlib.h>
#include "week_12_3_server.h"
int main()
{
    int n;
    printf("\nEnter the number of students : ");
    scanf("%d", &n);
    struct student *p = (struct student *)malloc(n * sizeof(struct stud
ent));
    for (int i = 0; i < n; i++)
    {
        printf("\nEnter the details of student %d : ", i + 1);
        read_details(p + i);
    }
    printf("\nThe details before sorting : ");
    for (int i = 0; i < n; i++)
    {
        printf("\nThe details of student %d : ", i + 1);
        print_details(p + i);
        printf("\n");
    }
    sort_details(p, n);
    printf("\nThe details after sorting : ");
    for (int i = 0; i < n; i++)
    {
        printf("\nThe details of student %d : ", i + 1);
        print_details(p + i);
        printf("\n");
    }
    free(p);
    return 0;
}
```

**server.h:**

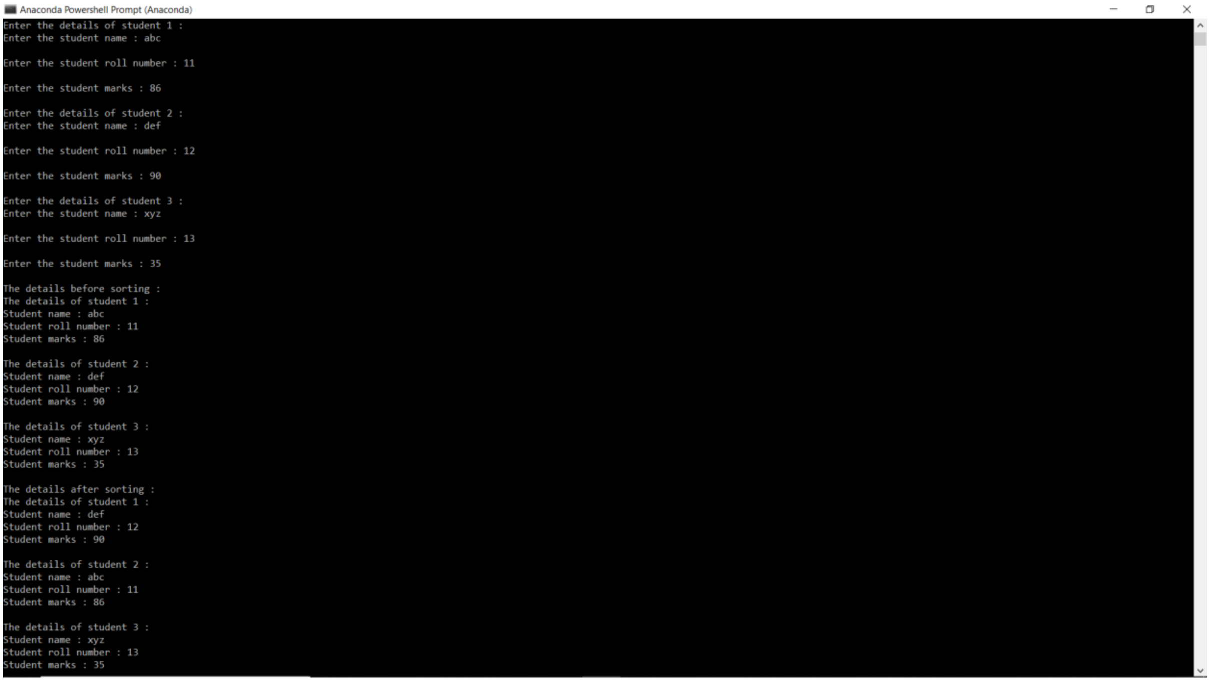
```
struct student

{
    int roll_num, marks;
    char *name;
};
void read_details(struct student *s);
void print_details(struct student *s);
void sort_details(struct student *s, int n);
```

**server.c:**

```
#include <stdio.h>

#include <stdlib.h>
#include "week_12_3_server.h"
void read_details(struct student *s)
{
    s->name = (char *)malloc(20 * sizeof(char));
    printf("\nEnter the student name : ");
    scanf("%s", s->name);
    printf("\nEnter the student roll number : ");
    scanf("%d", &s->roll_num);
    printf("\nEnter the student marks : ");
    scanf("%d", &s->marks);
}
void print_details(struct student *s)
{
    printf("\nStudent name : %s", s->name);
    printf("\nStudent roll number : %d", s->roll_num);
    printf("\nStudent marks : %d", s->marks);
}
void sort_details(struct student *s, int n)
{
    struct student temp;
    for (int i = 0; i < n; i++)
        for (int j = i + 1; j < n; j++)
```

```
            if ((s + i)->marks < (s + j)->marks)
            {
                temp = *(s + i);
                *(s + i) = *(s + j);
                *(s + j) = temp;
            }
}
```

**Output:**



```
Enter the details of student 1 :
Enter the student name : abc

Enter the student roll number : 11

Enter the student marks : 86

Enter the details of student 2 :
Enter the student name : def

Enter the student roll number : 12

Enter the student marks : 90

Enter the details of student 3 :
Enter the student name : xyz

Enter the student roll number : 13

Enter the student marks : 35

The details before sorting :
The details of student 1 :
Student name : abc
Student roll number : 11
Student marks : 86

The details of student 2 :
Student name : def
Student roll number : 12
Student marks : 90

The details of student 3 :
Student name : xyz
Student roll number : 13
Student marks : 35

The details after sorting :
The details of student 1 :
Student name : def
Student roll number : 12
Student marks : 90

The details of student 2 :
Student name : abc
Student roll number : 11
Student marks : 86

The details of student 3 :
Student name : xyz
Student roll number : 13
Student marks : 35
```

4)

**client.c:**

```
#include <stdio.h>

#include <stdlib.h>
#include "week_12_4_server.h"
int main()
{
    int n;
    struct course *c, *cd;
    c = (struct course *)malloc(6 * sizeof(struct course));
    for (int i = 0; i < 6; i++)
        read_courses(c + i);
    printf("\nThe courses registered are : ");
    for (int i = 0; i < 6; i++)
        print_courses(c + i);
    printf("\nEnter the number of courses you wish to drop : ");
    scanf("%d", &n);
    realloc(c, 6 - n);
    for (int i = 0; i < 6 - n; i++)
        read_courses(c + i);
    printf("\nThe courses registered after dropping are : ");
    for (int i = 0; i < n; i++)
        print_courses(c + i);
    return 0;
}
```

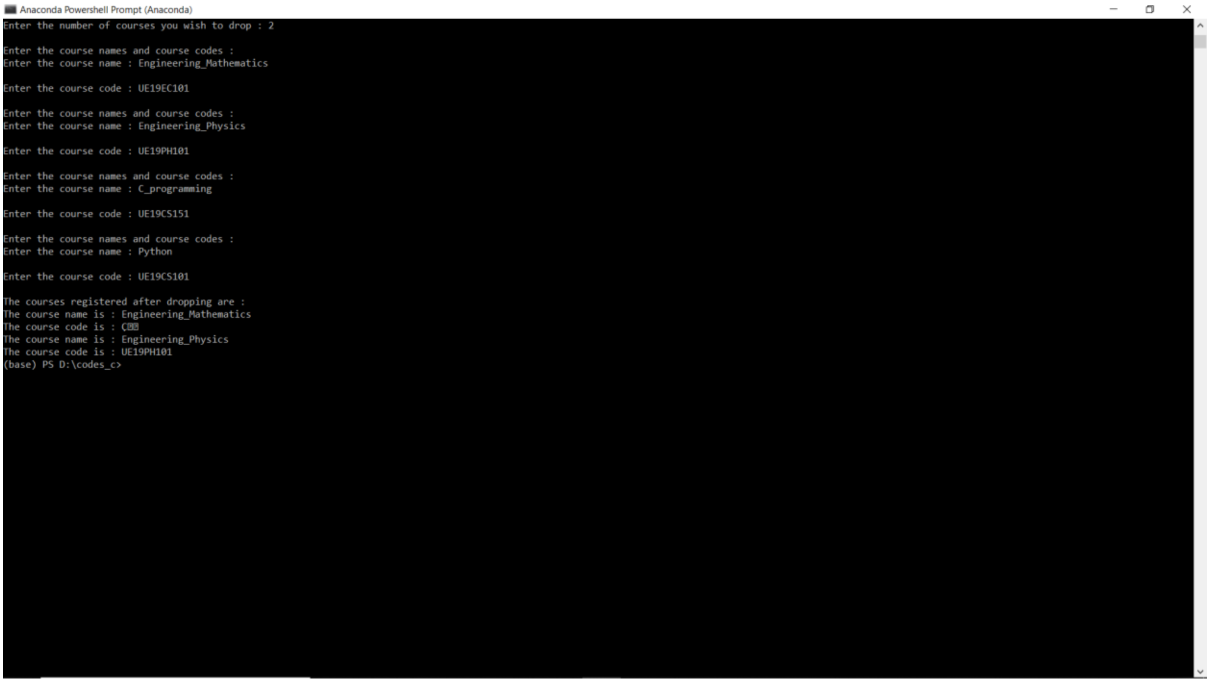
**server.h:**

```
struct course

{
    char *course_code, *course_name;
};
void read_courses(struct course *c);
void print_courses(struct course *c);
```

**server.c:**

```
#include <stdio.h>

#include <stdlib.h>
#include "week_12_4_server.h"
void read_courses(struct course *c)
{
    c->course_code = (char *)malloc(9 * sizeof(char));
    c->course_name = (char *)malloc(30 * sizeof(char));
    printf("\nEnter the course names and course codes : ");
    printf("\nEnter the course name : ");
    scanf("%s", c->course_name);
    printf("\nEnter the course code : ");
    scanf("%s", c->course_code);
}
void print_courses(struct course *c)
{
    printf("\nThe course name is : %s", c->course_name);
    printf("\nThe course code is : %s", c->course_code);
}
```

**Output:**



```
Enter the number of courses you wish to drop : 2

Enter the course names and course codes :
Enter the course name : Engineering_Mathematics

Enter the course code : UE19EC101

Enter the course names and course codes :
Enter the course name : Engineering_Physics

Enter the course code : UE19PH101

Enter the course names and course codes :
Enter the course name : C_programming

Enter the course code : UE19CS151

Enter the course names and course codes :
Enter the course name : Python

Enter the course code : UE19CS101

The courses registered after dropping are :
The course name is : Engineering_Mathematics
The course code is : Ç▯▯
The course name is : Engineering_Physics
The course code is : UE19PH101
(base) PS D:\codes_c>
```

5)

**client.c:**

```
#include <stdio.h>

#include <stdlib.h>
#include "week_12_5_server.h"
int main()
{
    int *a, n;
    printf("\nEnter the number of elements to be entered : ");
    scanf("%d", &n);
    a = (int *)malloc(n * sizeof(int));
    for (int i = 0; i < n; i++)
    {
        printf("\nEnter the element %d of the array : ", i);
        scanf("%d", (a + i));
    }
    populate(a, n);
    printf("\nThe squares of the entered elements are : ");
    for (int i = 0; i < n; i++)
        printf("\n%d", *(a + i));
    free(a);
    return 0;
}
```

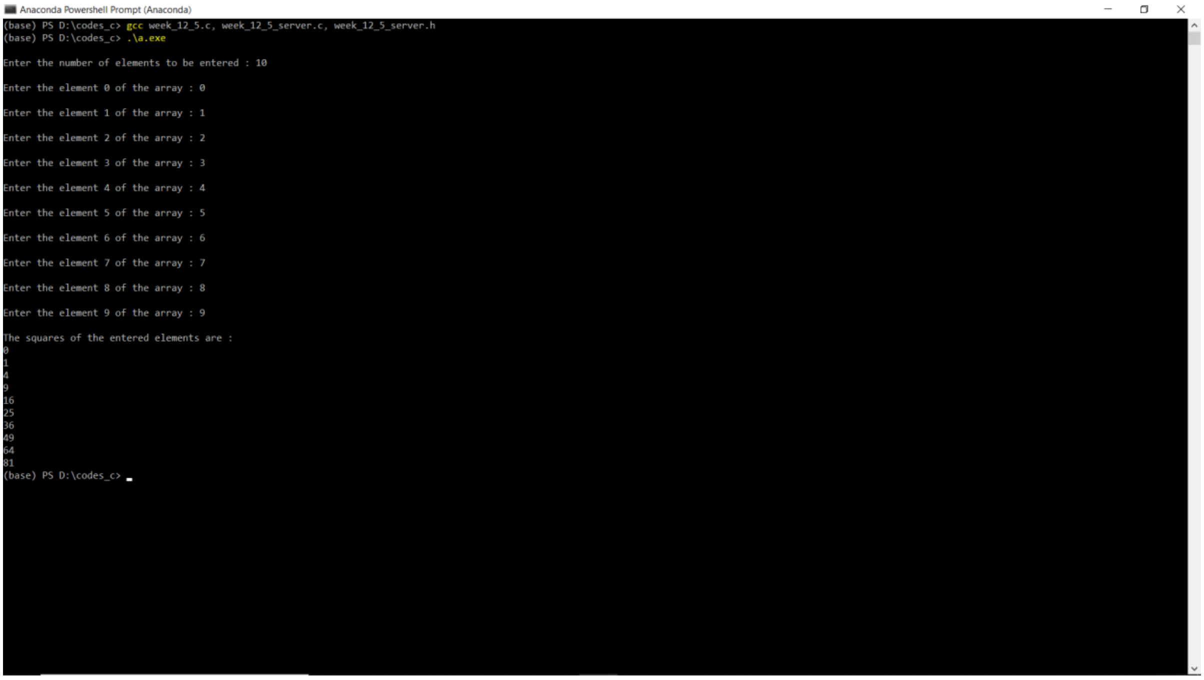
**server.h:**

```c
    void populate(int *a, int n);
```

**server.c:**

```c
#include <stdio.h>

#include <stdlib.h>
#include "week_12_5_server.h"
void populate(int *a, int n)
{
    for (int i = 0; i < n; i++)
        *(a + i) = *(a + i) * *(a + i);
}
```

**Output:**



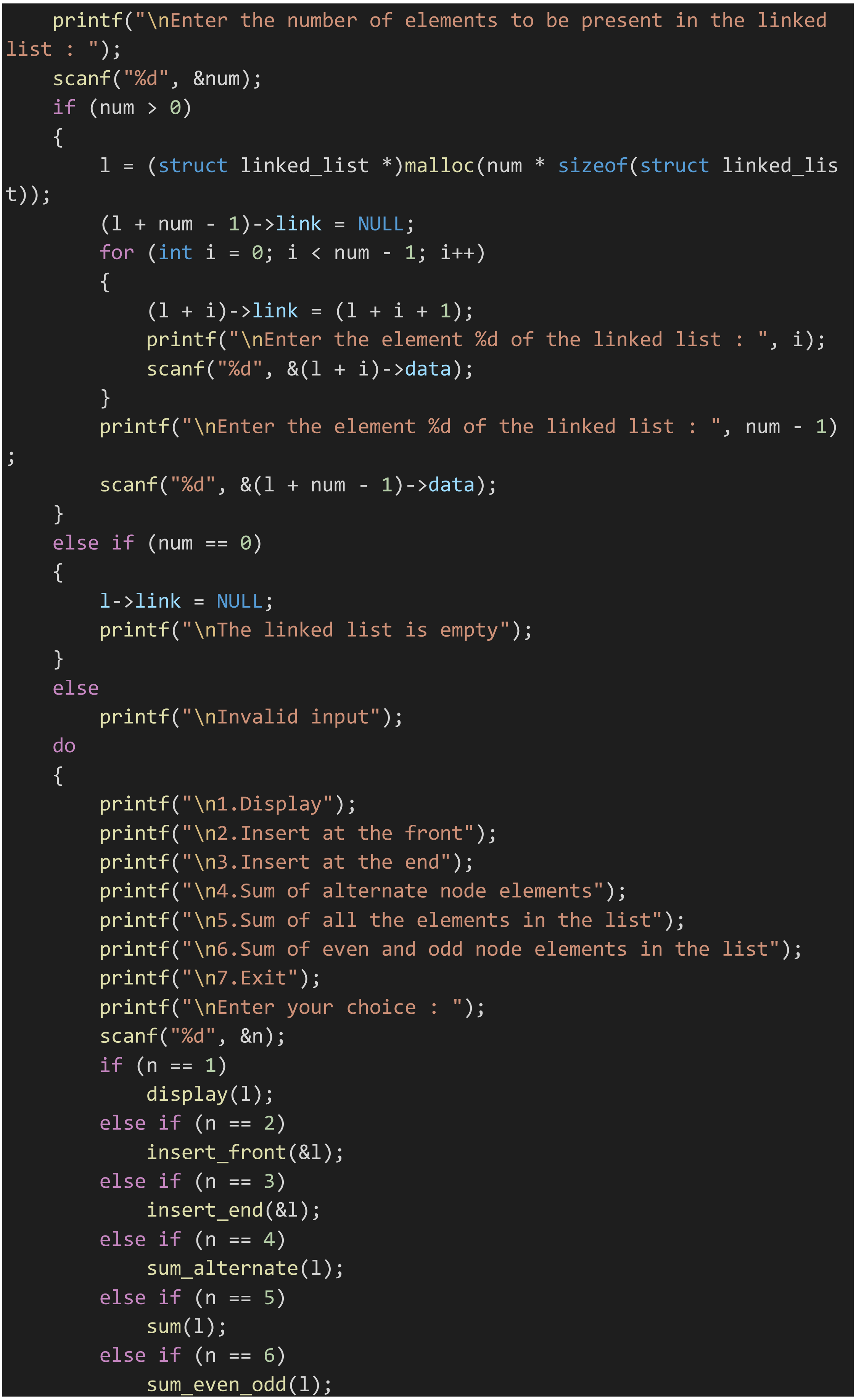
```
(base) PS D:\codes_c> gcc week_12_5.c, week_12_5_server.c, week_12_5_server.h
(base) PS D:\codes_c> .\a.exe

Enter the number of elements to be entered : 10

Enter the element 0 of the array : 0

Enter the element 1 of the array : 1

Enter the element 2 of the array : 2

Enter the element 3 of the array : 3

Enter the element 4 of the array : 4

Enter the element 5 of the array : 5

Enter the element 6 of the array : 6

Enter the element 7 of the array : 7

Enter the element 8 of the array : 8

Enter the element 9 of the array : 9

The squares of the entered elements are :
0
1
4
9
16
25
36
49
64
81
(base) PS D:\codes_c>
```

**WEEK 14:**

1)

**client.c:**

```c
#include <stdio.h>

#include <stdlib.h>
#include "week_14_1_server.h"
int main()
{
    struct linked_list *l;
    int n, num = 0;
```



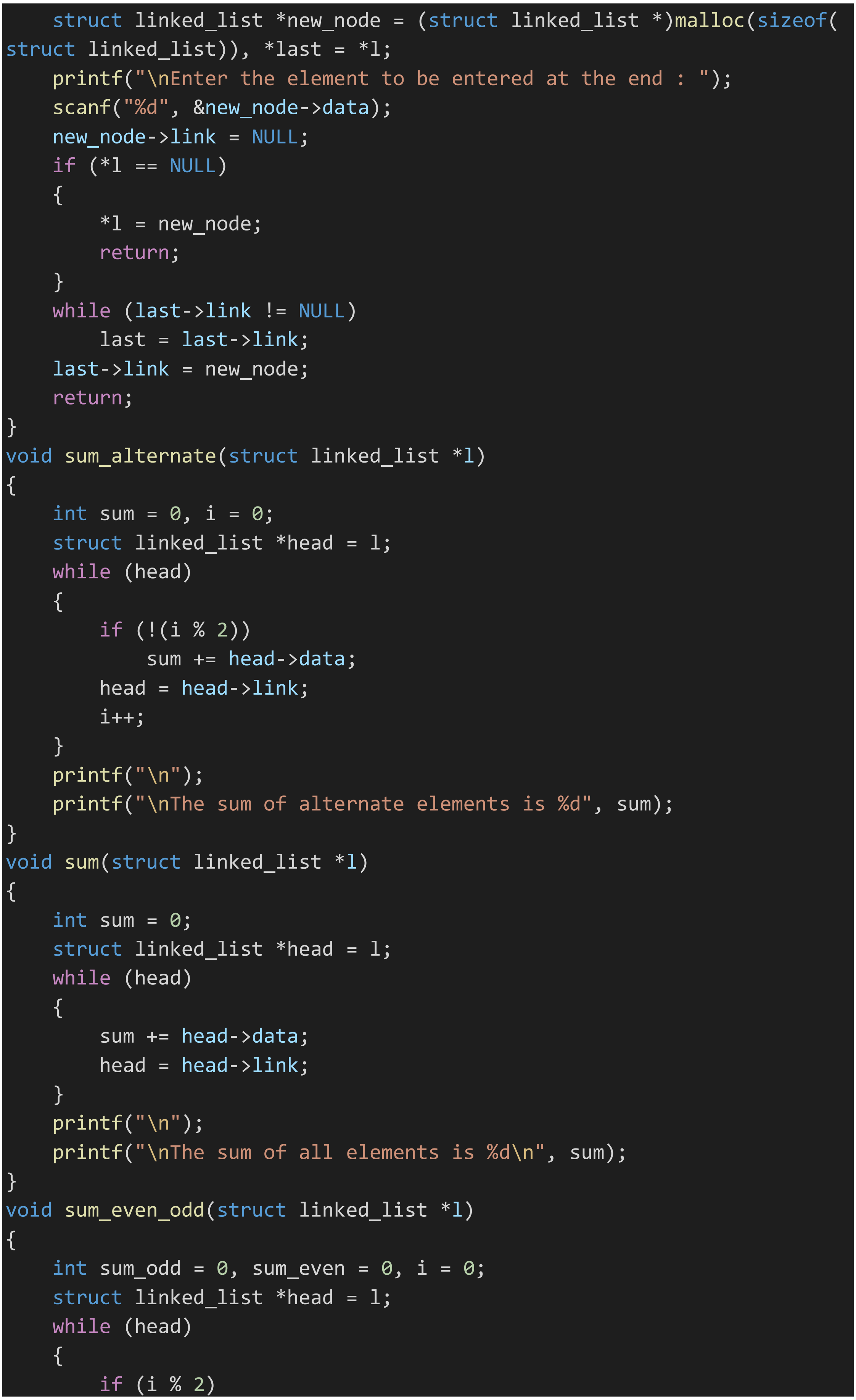
```c
    printf("\nEnter the number of elements to be present in the linked list : ");
    scanf("%d", &num);
    if (num > 0)
    {
        l = (struct linked_list *)malloc(num * sizeof(struct linked_list));
        (l + num - 1)->link = NULL;
        for (int i = 0; i < num - 1; i++)
        {
            (l + i)->link = (l + i + 1);
            printf("\nEnter the element %d of the linked list : ", i);
            scanf("%d", &(l + i)->data);
        }
        printf("\nEnter the element %d of the linked list : ", num - 1);
        scanf("%d", &(l + num - 1)->data);
    }
    else if (num == 0)
    {
        l->link = NULL;
        printf("\nThe linked list is empty");
    }
    else
        printf("\nInvalid input");
    do
    {
        printf("\n1.Display");
        printf("\n2.Insert at the front");
        printf("\n3.Insert at the end");
        printf("\n4.Sum of alternate node elements");
        printf("\n5.Sum of all the elements in the list");
        printf("\n6.Sum of even and odd node elements in the list");
        printf("\n7.Exit");
        printf("\nEnter your choice : ");
        scanf("%d", &n);
        if (n == 1)
            display(l);
        else if (n == 2)
            insert_front(&l);
        else if (n == 3)
            insert_end(&l);
        else if (n == 4)
            sum_alternate(l);
        else if (n == 5)
            sum(l);
        else if (n == 6)
            sum_even_odd(l);
        else if (n != 7)
            printf("\nInvalid input");
    } while (n != 7);
    return 0;
}
```

**server.h:**

```
struct linked_list

{
    int data;
    struct linked_list *link;
};
void display(struct linked_list *l);
void insert_front(struct linked_list **l);
void insert_end(struct linked_list **l);
void sum_alternate(struct linked_list *l);
void sum(struct linked_list *l);
void sum_even_odd(struct linked_list *l);
```

**server.c:**

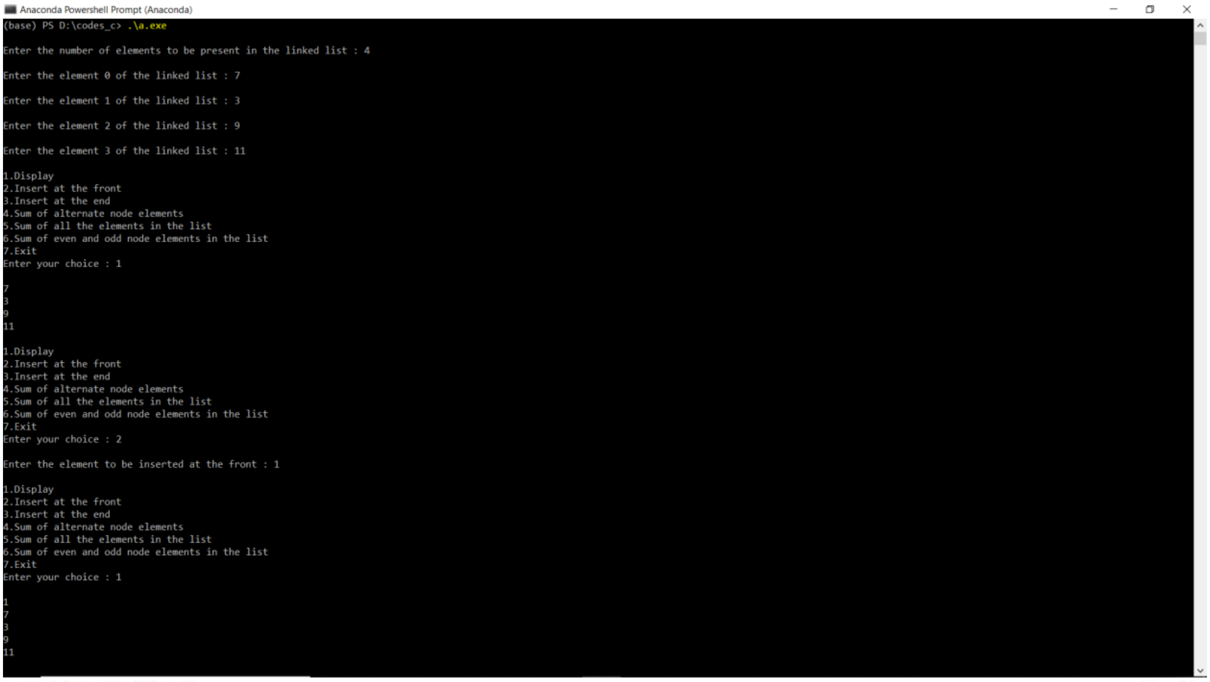
```
#include <stdio.h>

#include <stdlib.h>
#include "week_14_1_server.h"
void display(struct linked_list *l)
{
    struct linked_list *head = l;
    while (head)
    {
        printf("\n%d", head->data);
        head = head->link;
    }
    printf("\n");
}
void insert_front(struct linked_list **l)
{
    struct linked_list *new_node = (struct linked_list *)malloc(sizeof(
struct linked_list));
    printf("\nEnter the element to be inserted at the front : ");
    scanf("%d", &new_node->data);
    new_node->link = *l;
    *l = new_node;
}
void insert_end(struct linked_list **l)
{
```



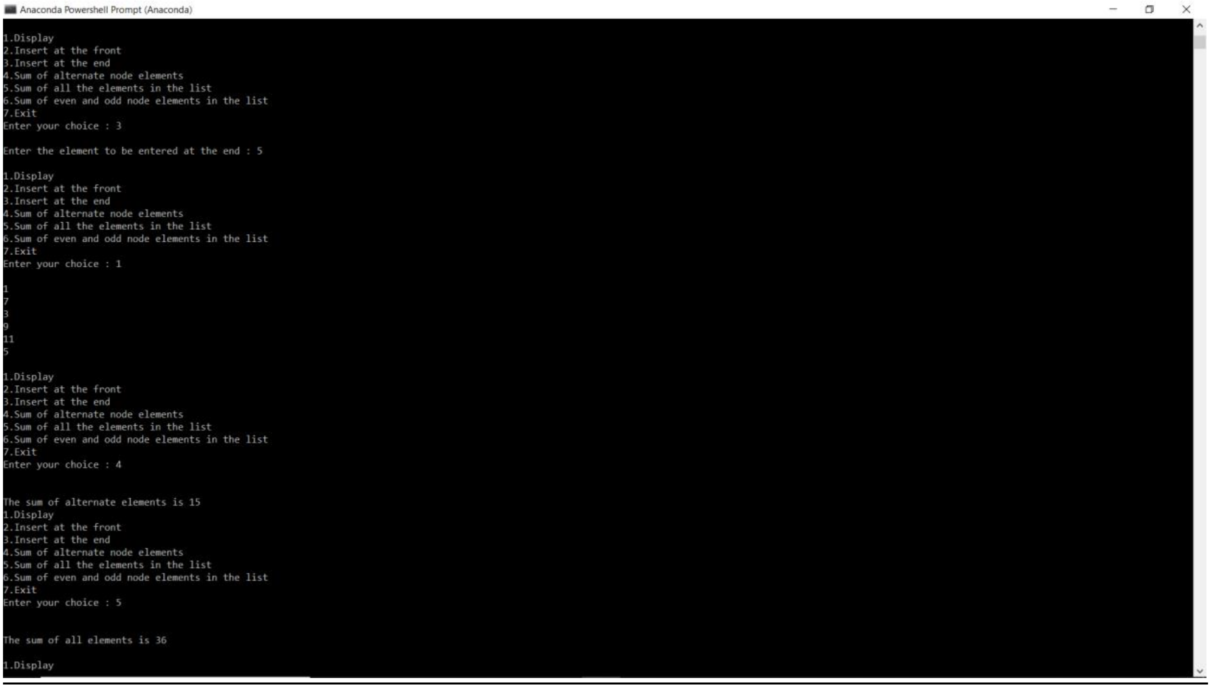
```
    struct linked_list *new_node = (struct linked_list *)malloc(sizeof(
struct linked_list)), *last = *l;
    printf("\nEnter the element to be entered at the end : ");
    scanf("%d", &new_node->data);
    new_node->link = NULL;
    if (*l == NULL)
    {
        *l = new_node;
        return;
    }
    while (last->link != NULL)
        last = last->link;
    last->link = new_node;
    return;
}
void sum_alternate(struct linked_list *l)
{
    int sum = 0, i = 0;
    struct linked_list *head = l;
    while (head)
    {
        if (!(i % 2))
            sum += head->data;
        head = head->link;
        i++;
    }
    printf("\n");
    printf("\nThe sum of alternate elements is %d", sum);
}
void sum(struct linked_list *l)
{
    int sum = 0;
    struct linked_list *head = l;
    while (head)
    {
        sum += head->data;
        head = head->link;
    }
    printf("\n");
    printf("\nThe sum of all elements is %d\n", sum);
}
void sum_even_odd(struct linked_list *l)
{
    int sum_odd = 0, sum_even = 0, i = 0;
    struct linked_list *head = l;
    while (head)
    {
        if (i % 2)
```

```
            sum_odd += head->data;
        else
            sum_even += head->data;
        head = head->link;
        i++;
    }
    printf("\n");
    ;
    printf("\nThe sum of elements at even nodes is %d", sum_even);
    printf("\nThe sum of elements at odd nodes is %d", sum_odd);
}
```

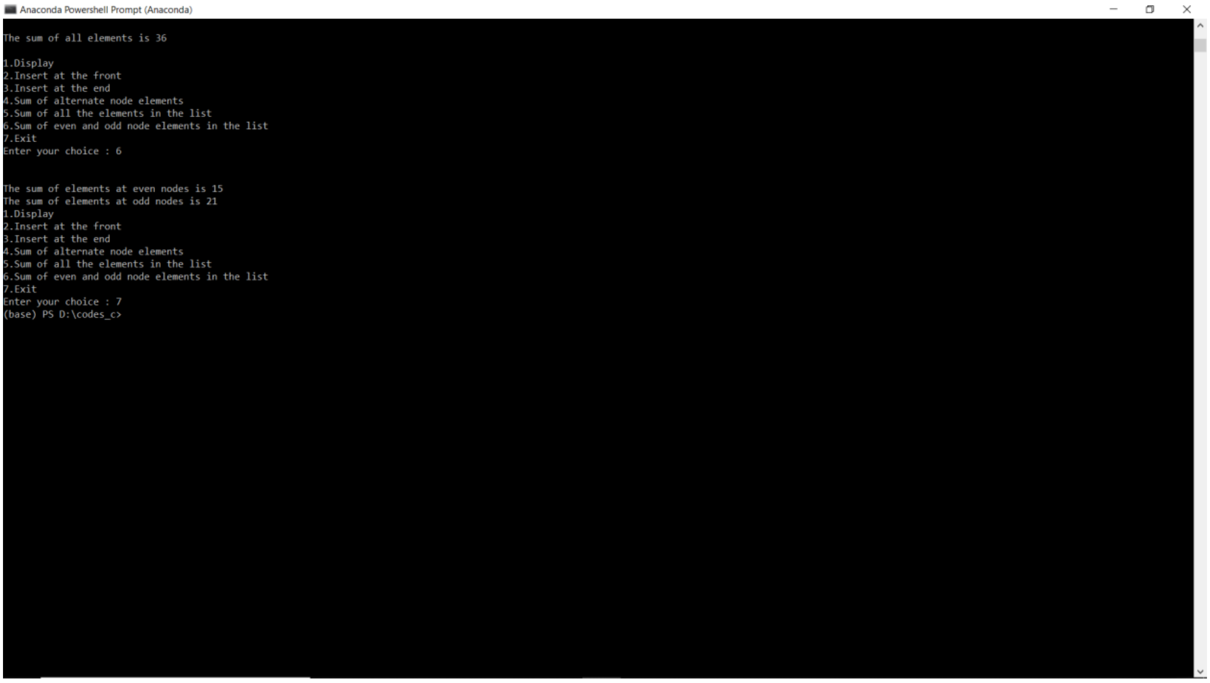
**Output:**



```
(base) PS D:\codes_c> .\a.exe

Enter the number of elements to be present in the linked list : 4

Enter the element 0 of the linked list : 7

Enter the element 1 of the linked list : 3

Enter the element 2 of the linked list : 9

Enter the element 3 of the linked list : 11

1.Display
2.Insert at the front
3.Insert at the end
4.Sum of alternate node elements
5.Sum of all the elements in the list
6.Sum of even and odd node elements in the list
7.Exit
Enter your choice : 1

7
3
9
11

1.Display
2.Insert at the front
3.Insert at the end
4.Sum of alternate node elements
5.Sum of all the elements in the list
6.Sum of even and odd node elements in the list
7.Exit
Enter your choice : 2

Enter the element to be inserted at the front : 1

1.Display
2.Insert at the front
3.Insert at the end
4.Sum of alternate node elements
5.Sum of all the elements in the list
6.Sum of even and odd node elements in the list
7.Exit
Enter your choice : 1

1
7
3
9
11
```



```

1.Display
2.Insert at the front
3.Insert at the end
4.Sum of alternate node elements
5.Sum of all the elements in the list
6.Sum of even and odd node elements in the list
7.Exit
Enter your choice : 3

Enter the element to be entered at the end : 5

1.Display
2.Insert at the front
3.Insert at the end
4.Sum of alternate node elements
5.Sum of all the elements in the list
6.Sum of even and odd node elements in the list
7.Exit
Enter your choice : 1

1
7
3
9
11
5

1.Display
2.Insert at the front
3.Insert at the end
4.Sum of alternate node elements
5.Sum of all the elements in the list
6.Sum of even and odd node elements in the list
7.Exit
Enter your choice : 4


The sum of alternate elements is 15
1.Display
2.Insert at the front
3.Insert at the end
4.Sum of alternate node elements
5.Sum of all the elements in the list
6.Sum of even and odd node elements in the list
7.Exit
Enter your choice : 5


The sum of all elements is 36

1.Display
```



```
The sum of all elements is 36

1.Display
2.Insert at the front
3.Insert at the end
4.Sum of alternate node elements
5.Sum of all the elements in the list
6.Sum of even and odd node elements in the list
7.Exit
Enter your choice : 6


The sum of elements at even nodes is 15
The sum of elements at odd nodes is 21
1.Display
2.Insert at the front
3.Insert at the end
4.Sum of alternate node elements
5.Sum of all the elements in the list
6.Sum of even and odd node elements in the list
7.Exit
Enter your choice : 7
(base) PS D:\codes_c>
```

2)

**client.c:**

```c
#include <stdio.h>

#include <stdlib.h>
#include "week_14_2_server.h"
int main()
{
    struct linked_list *l;
    int n, num = 0;
    printf("\nEnter the number of elements to be present in the linked
list : ");
    scanf("%d", &num);
    if (num > 0)
    {
        l = (struct linked_list *)malloc(num * sizeof(struct linked_lis
t));
        (l + num - 1)->link = NULL;
        for (int i = 0; i < num - 1; i++)
        {
            (l + i)->link = (l + i + 1);
            printf("\nEnter the element %d of the linked list : ", i);
            scanf("%d", &(l + i)->data);
        }
        printf("\nEnter the element %d of the linked list : ", num - 1)
;
        scanf("%d", &(l + num - 1)->data);
    }
    else if (num == 0)
    {
        l->link = NULL;
        printf("\nThe linked list is empty");
    }
    else
        printf("\nInvalid input");
    do
    {
        printf("\n1.Display");
        printf("\n2.Delete at the front");
        printf("\n3.Delete at the end");
        printf("\n4.Search the element");
        printf("\n5.Product of nodes of a linked list which are divisib
le by a given number");
        printf("\n6.Exit");
        printf("\nEnter your choice : ");
        scanf("%d", &n);
        if (n == 1)
            display(l);
        else if (n == 2)
            delete_front(&l);
        else if (n == 3)
            delete_end(&l);
        else if (n == 4)
            search_element(l);
        else if (n == 5)
            product_nodes(l);
        else if (n != 6)
            printf("\nInvalid input");
    } while (n != 6);
    free(l);
    return 0;
}
```

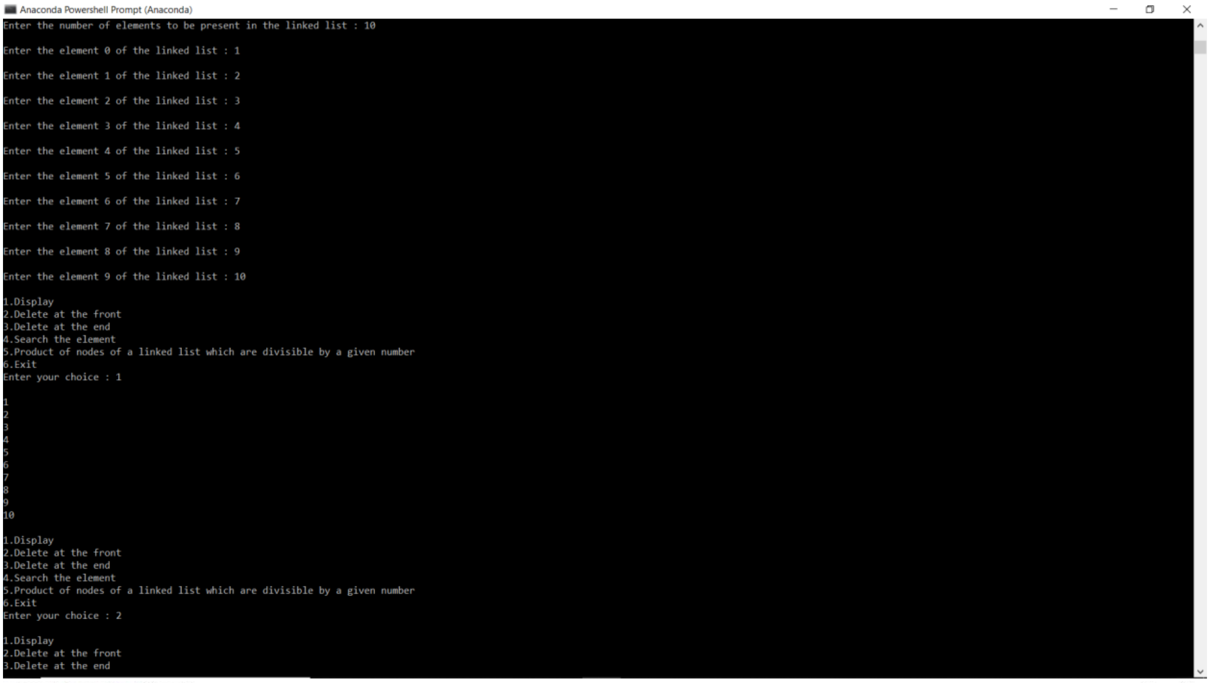
**server.h:**

```
struct linked_list

{
    int data;
    struct linked_list *link;
};
void display(struct linked_list *l);
void delete_front(struct linked_list **l);
void delete_end(struct linked_list **l);
void search_element(struct linked_list *l);
void product_nodes(struct linked_list *l);
```

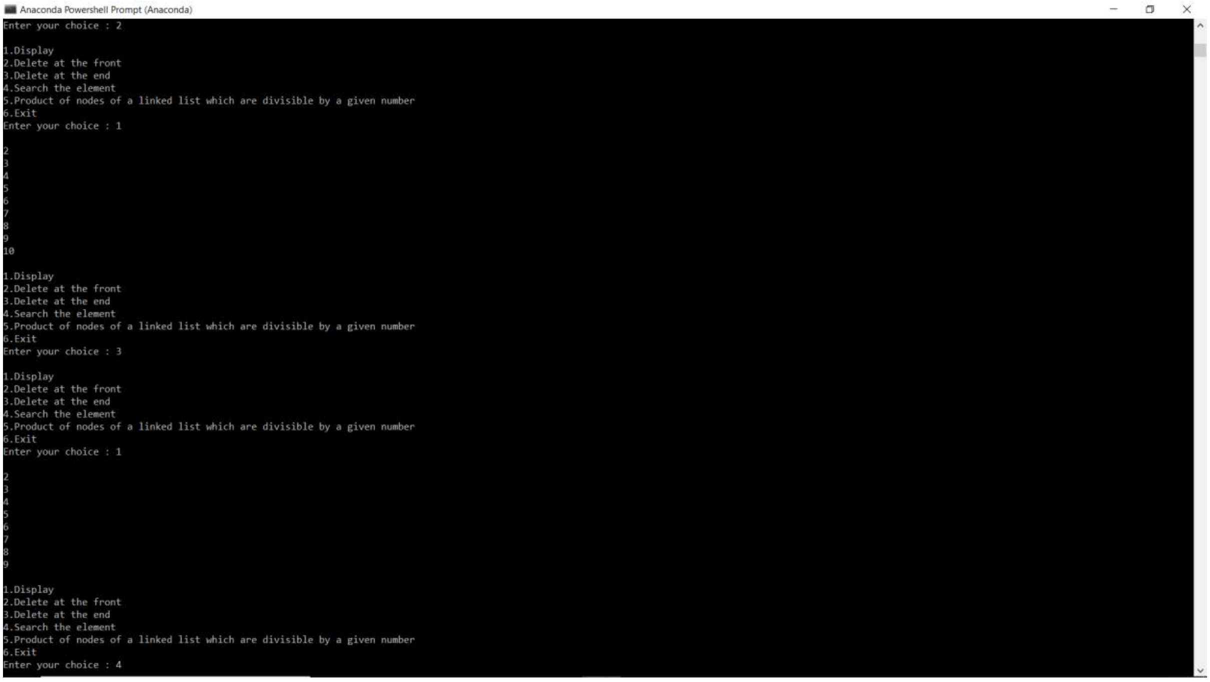
**server.c:**

```
#include <stdio.h>

#include <stdlib.h>
#include "week_14_2_server.h"
void display(struct linked_list *l)
{
    struct linked_list *head = l;
    while (head)
    {
        printf("\n%d", head->data);
        head = head->link;
    }
    printf("\n");
}
void delete_front(struct linked_list **l)
{
    if (!(*l))
        return;
    struct linked_list *ref = *l;
    *l = ref->link;
    free(ref);
    return;
}
void delete_end(struct linked_list **l)
{
    if (!(*l))
        return;
    struct linked_list *ref = *l;
    while ((ref->link)->link)
        ref = ref->link;
    free(ref->link);
    ref->link = NULL;
}
void search_element(struct linked_list *l)
{
    int n, i = 0;
    printf("\nEnter the element to be searched : ");
    scanf("%d", &n);
    struct linked_list *head = l;
    do
    {
        if (head->data == n)
        {
            printf("\nThe element %d is found at node %d", n, i);
            i = 0;
            break;
        }
        i++;
        head = head->link;
    } while (head);
    if (i)
        printf("\nThe element %d is not there in the linked list", n);
    printf("\n");
}
void product_nodes(struct linked_list *l)
{
    int product = 1, i = 0, n;
    printf("\nEnter the number : ");
    scanf("%d", &n);
    struct linked_list *head = l;
    do
    {
        if (!(i % n))
            product *= head->data;
        i++;
        head = head->link;
    } while (head);
    printf("\nProduct of nodes of a linked list which are divisible by
%d is %d", n, product);
}
```

## Output:



```
Enter the number of elements to be present in the linked list : 10

Enter the element 0 of the linked list : 1

Enter the element 1 of the linked list : 2

Enter the element 2 of the linked list : 3

Enter the element 3 of the linked list : 4

Enter the element 4 of the linked list : 5

Enter the element 5 of the linked list : 6

Enter the element 6 of the linked list : 7

Enter the element 7 of the linked list : 8

Enter the element 8 of the linked list : 9

Enter the element 9 of the linked list : 10

1.Display
2.Delete at the front
3.Delete at the end
4.Search the element
5.Product of nodes of a linked list which are divisible by a given number
6.Exit
Enter your choice : 1

1
2
3
4
5
6
7
8
9
10

1.Display
2.Delete at the front
3.Delete at the end
4.Search the element
5.Product of nodes of a linked list which are divisible by a given number
6.Exit
Enter your choice : 2

1.Display
2.Delete at the front
3.Delete at the end
```



```
Enter your choice : 2

1.Display
2.Delete at the front
3.Delete at the end
4.Search the element
5.Product of nodes of a linked list which are divisible by a given number
6.Exit
Enter your choice : 1

2
3
4
5
6
7
8
9
10

1.Display
2.Delete at the front
3.Delete at the end
4.Search the element
5.Product of nodes of a linked list which are divisible by a given number
6.Exit
Enter your choice : 3

1.Display
2.Delete at the front
3.Delete at the end
4.Search the element
5.Product of nodes of a linked list which are divisible by a given number
6.Exit
Enter your choice : 1

2
3
4
5
6
7
8
9

1.Display
2.Delete at the front
3.Delete at the end
4.Search the element
5.Product of nodes of a linked list which are divisible by a given number
6.Exit
Enter your choice : 4
```

**WEEK 15:**

1)

**client.c:**

```c
#include <stdio.h>

#include <stdlib.h>
#include "week_15_1_server.h"
int main()
{
    int n;
    printf("\nEnter number of elements to be sorted : ");
    scanf("%d", &n);
    int *a = (int *)malloc(sizeof(int));
    printf("\nEnter the desired values : ");
    for (int i = 0; i < n; i++)
        *(a + i) = read(*(a + i));
    printf("\nValues before sorting : ");
    display(a, n);
    sort(a, n);
    printf("\nValues after sorting : ");
    display(a, n);
    free(a);
}
```

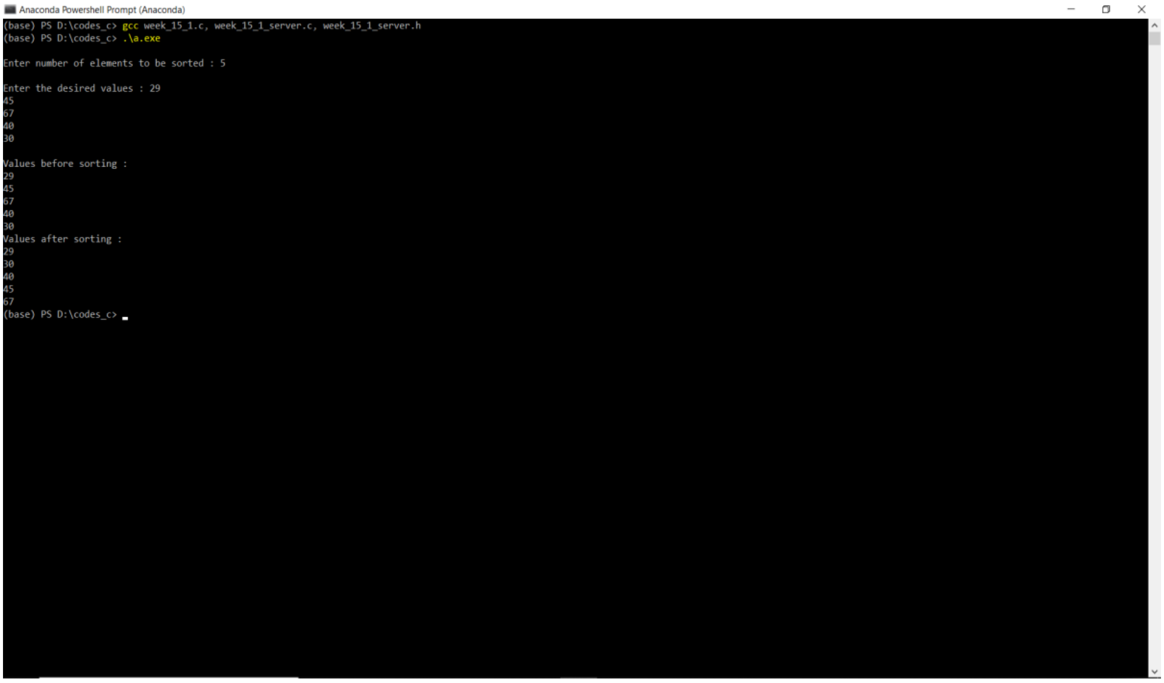
**server.h:**

```c
void display(int a[], int n);
void sort(int a[], int n);
int read(int a);
```

**server.c:**

```c
#include <stdio.h>
#include "week_15_1_server.h"
void display(int a[], int n)
{
    for (int i = 0; i < n; i++)
        printf("\n%d", a[i]);
}
int read(int a)
{
    scanf("%d", &a);
    return a;
}
void sort(int a[], int n)
{
    int swap, temp;
    for (int i = 0; i < n; i++)
    {
        swap = 0;
        for (int j = 0; j < n - 1; j++)
        {
            if (a[j] > a[j + 1])
            {
                temp = a[j + 1];
                a[j + 1] = a[j];
                a[j] = temp;
                swap++;
            }
        }
        if (swap == 0)
            break;
    }
}
```

**Output:**



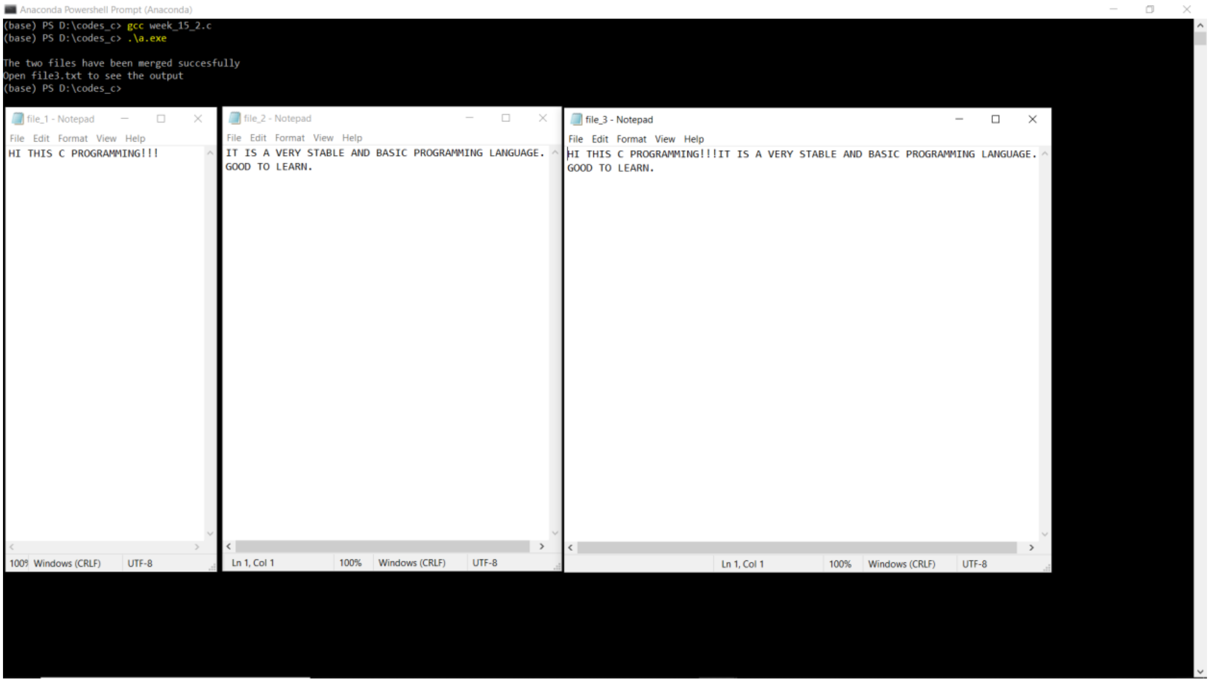
```
(base) PS D:\codes_c> gcc week_15_1.c, week_15_1_server.c, week_15_1_server.h
(base) PS D:\codes_c> .\a.exe

Enter number of elements to be sorted : 5

Enter the desired values : 29
45
67
40
30

Values before sorting :
29
45
67
40
30
Values after sorting :
29
30
40
45
67
(base) PS D:\codes_c> _
```

2)

**client.c:**

```
#include <stdio.h>

#include <stdlib.h>

int main()
{
    FILE *ptr_1 = fopen("file_1.txt", "r");
    char ch;
    FILE *ptr_2 = fopen("file_2.txt", "r");
    FILE *ptr_3 = fopen("file_3.txt", "w");
    while ((ch = fgetc(ptr_1)) != EOF)
        fputc(ch, ptr_3);
    while ((ch = fgetc(ptr_2)) != EOF)
        fputc(ch, ptr_3);
    free(ptr_1);
    free(ptr_2);
    free(ptr_3);
    fclose(ptr_1);
    fclose(ptr_2);
    fclose(ptr_3);
    printf("\nThe two files have been merged succesfully\nOpen file3.txt to see the output");
    return 0;
}
```

**Output:**



Anaconda Powershell Prompt (Anaconda)
file_1 - Notepad
File Edit Format View Help
100% Windows (CRLF) UTF-8
file_2 - Notepad
File Edit Format View Help
Ln 1, Col 1 100% Windows (CRLF) UTF-8
file_3 - Notepad
File Edit Format View Help
Ln 1, Col 1 100% Windows (CRLF) UTF-8

```
(base) PS D:\codes_c> gcc week_15_2.c
(base) PS D:\codes_c> .\a.exe

The two files have been merged succesfully
Open file3.txt to see the output
(base) PS D:\codes_c>
```

```
HI THIS C PROGRAMMING!!!
```

```
IT IS A VERY STABLE AND BASIC PROGRAMMING LANGUAGE.
GOOD TO LEARN.
```

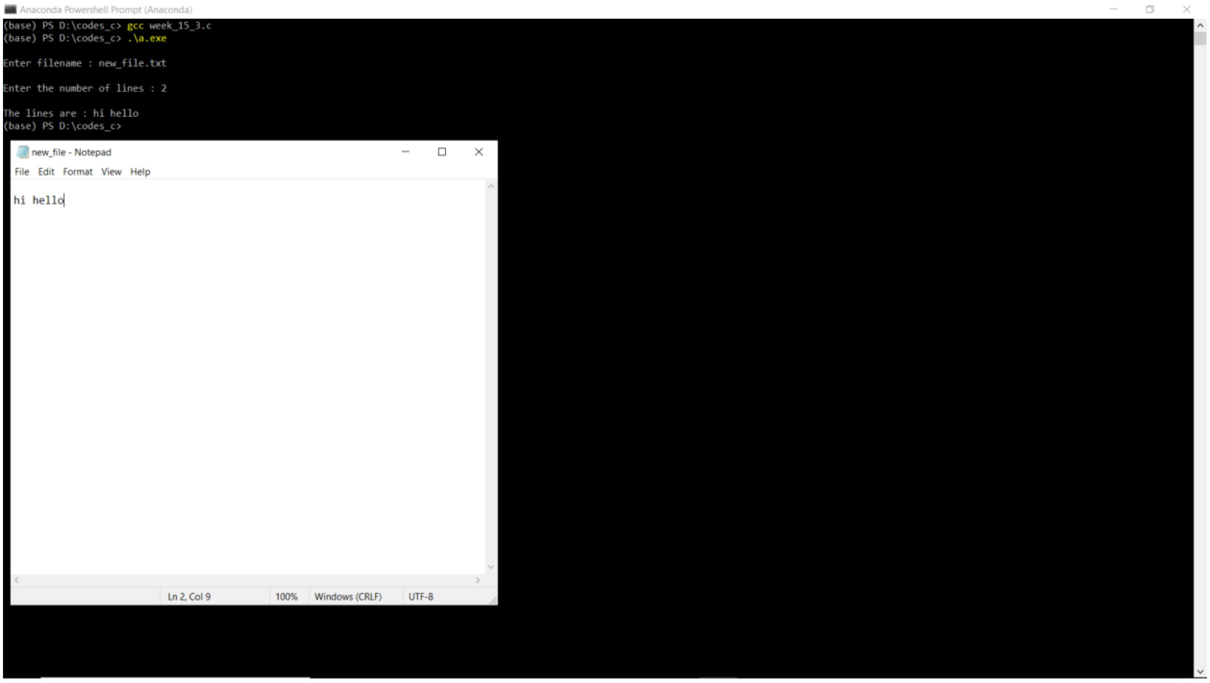
```
HI THIS C PROGRAMMING!!!IT IS A VERY STABLE AND BASIC PROGRAMMING LANGUAGE.
GOOD TO LEARN.
```

3)

**client.c:**

```
#include <stdio.h>

#include <stdlib.h>
int main()
{
    char *name = (char *)malloc(sizeof(char) * 256);
    char *str = (char *)malloc(sizeof(char) * 256);
    int n;
    printf("\nEnter filename : ");
    scanf("%s", name);
    FILE *ptr_1 = fopen(name, "w");
    printf("\nEnter the number of lines : ");
    scanf("%d", &n);
    printf("\nThe lines are : ");
    for (int i = 0; i < n + 1; i++)
    {
        fgets(str, sizeof(str), stdin);
        fputs(str, ptr_1);
    }
    fclose(ptr_1);
    free(name);
    free(str);
}
```

**Output:**


Anaconda Powershell Prompt (Anaconda)

```
(base) PS D:\codes_c> gcc week_15_3.c
(base) PS D:\codes_c> .\a.exe

Enter filename : new_file.txt

Enter the number of lines : 2

The lines are : hi hello
(base) PS D:\codes_c>
```

new_file - Notepad
File Edit Format View Help

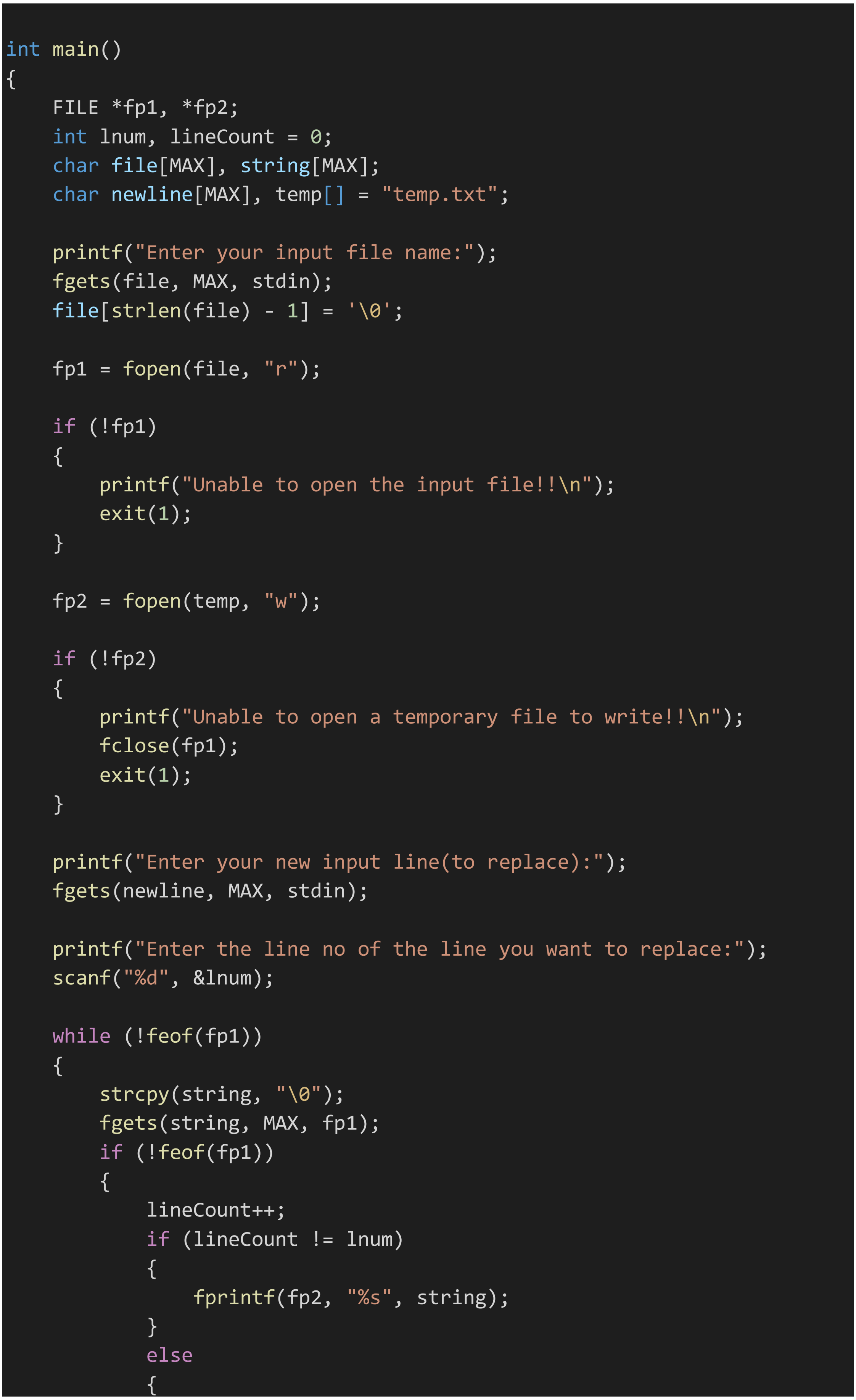
```
hi hello
```

Ln 2, Col 9
100%
Windows (CRLF)
UTF-8

4)

**client.c:**

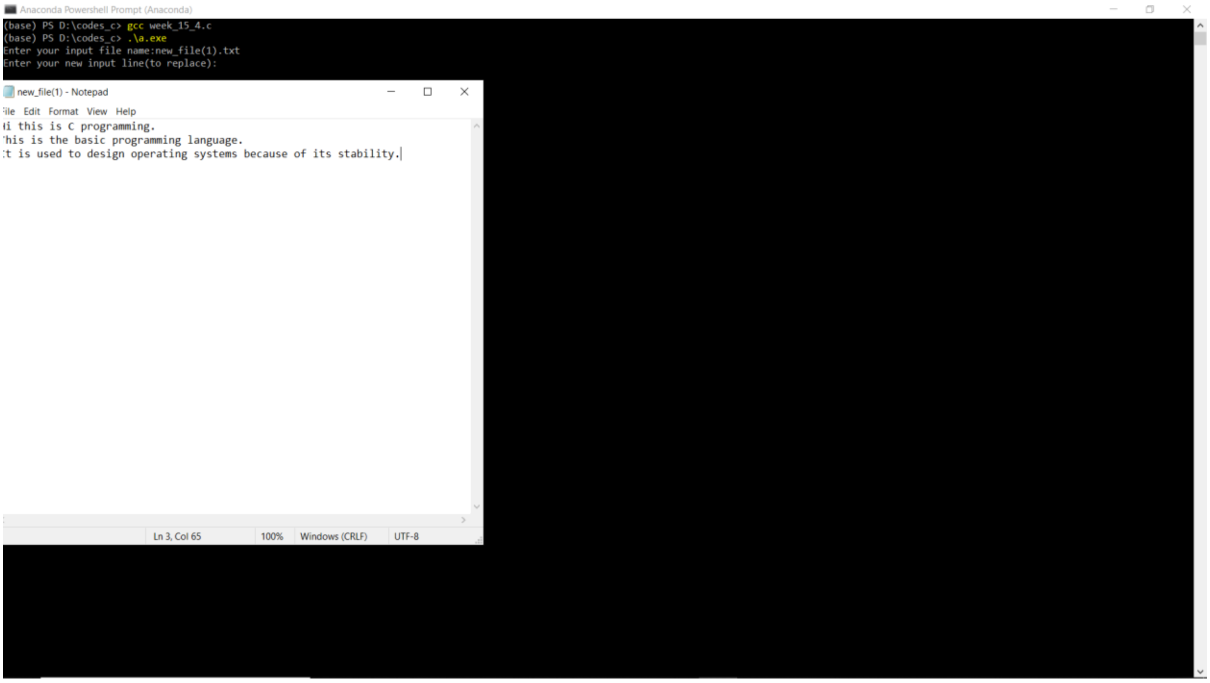
```
#include <stdio.h>

#include <stdlib.h>
#include <string.h>
#define MAX 256
```



```c
int main()
{
    FILE *fp1, *fp2;
    int lnum, lineCount = 0;
    char file[MAX], string[MAX];
    char newline[MAX], temp[] = "temp.txt";

    printf("Enter your input file name:");
    fgets(file, MAX, stdin);
    file[strlen(file) - 1] = '\0';

    fp1 = fopen(file, "r");

    if (!fp1)
    {
        printf("Unable to open the input file!!\n");
        exit(1);
    }

    fp2 = fopen(temp, "w");

    if (!fp2)
    {
        printf("Unable to open a temporary file to write!!\n");
        fclose(fp1);
        exit(1);
    }

    printf("Enter your new input line(to replace):");
    fgets(newline, MAX, stdin);

    printf("Enter the line no of the line you want to replace:");
    scanf("%d", &lnum);

    while (!feof(fp1))
    {
        strcpy(string, "\0");
        fgets(string, MAX, fp1);
        if (!feof(fp1))
        {
            lineCount++;
            if (lineCount != lnum)
            {
                fprintf(fp2, "%s", string);
            }
            else
            {
```

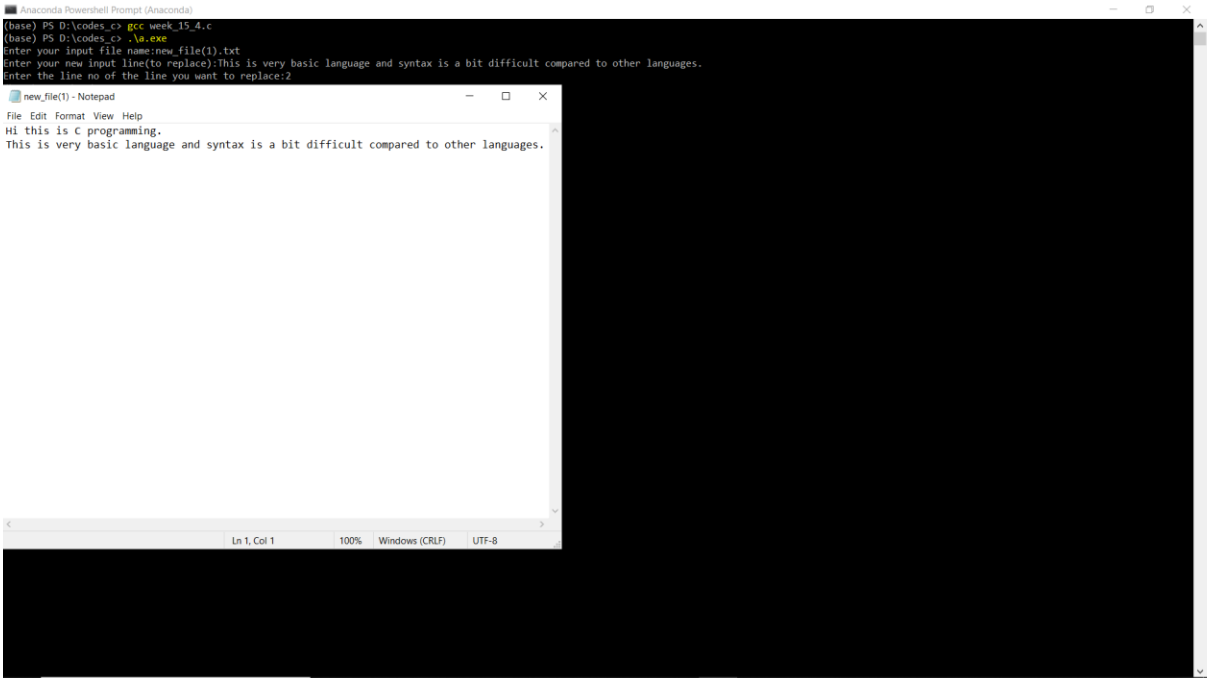
```
                        fprintf(fp2, "%s", newline);
                    }
                }
            }

    fclose(fp1);
    fclose(fp2);

    remove(file);

    rename(temp, file);
}
```

**Output:**


Anaconda Powershell Prompt (Anaconda)
new_file(1) - Notepad
File Edit Format View Help
Ln 3, Col 65
100%
Windows (CRLF)
UTF-8

```
(base) PS D:\codes_c> gcc week_15_4.c
(base) PS D:\codes_c> .\a.exe
Enter your input file name:new_file(1).txt
Enter your new input line(to replace):
```

```
Hi this is C programming.
This is the basic programming language.
It is used to design operating systems because of its stability.
```

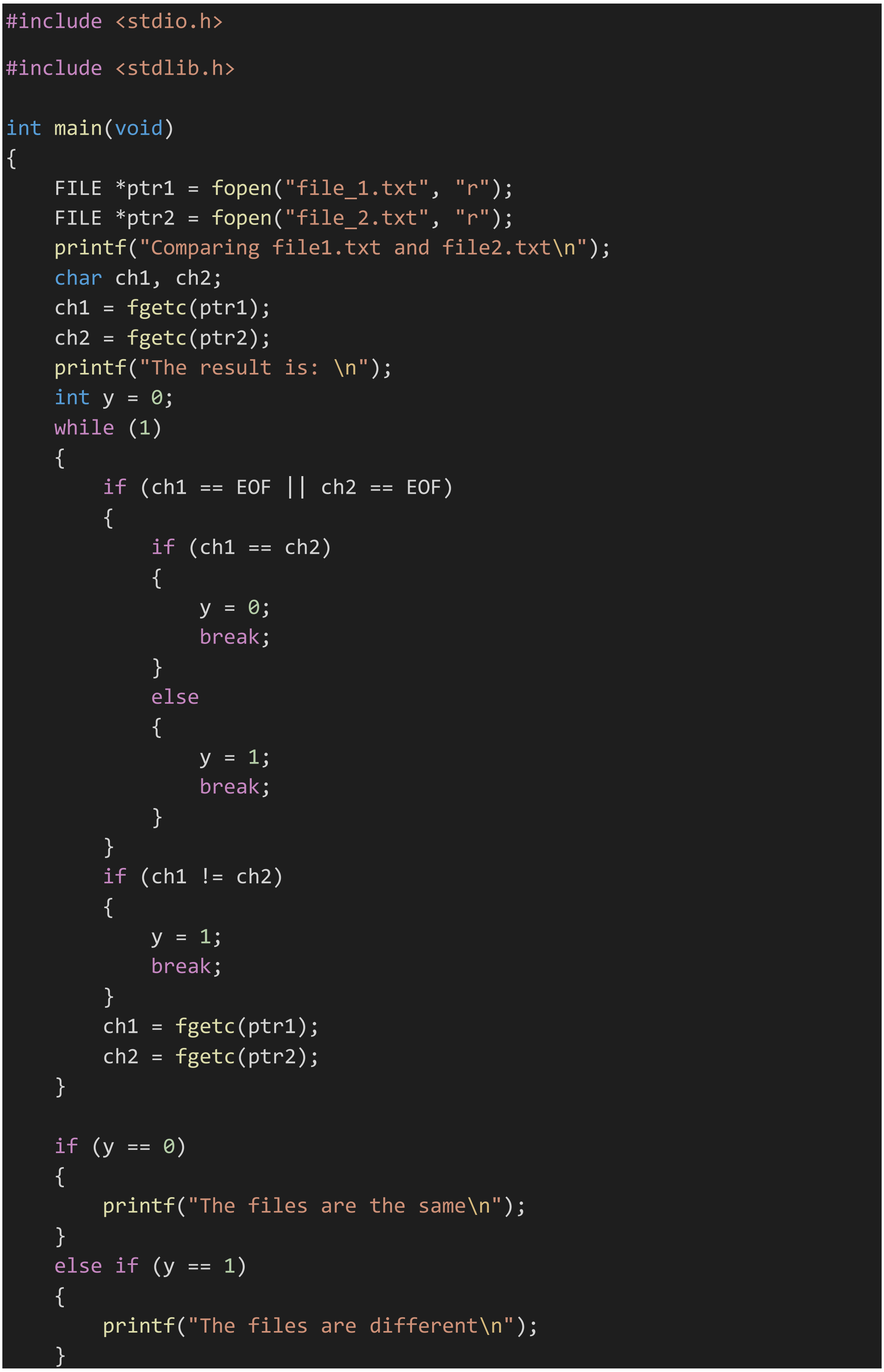

Anaconda Powershell Prompt (Anaconda)
new_file(1) - Notepad
File Edit Format View Help
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8

```
(base) PS D:\codes_c> gcc week_15_4.c
(base) PS D:\codes_c> .\a.exe
Enter your input file name:new_file(1).txt
Enter your new input line(to replace):This is very basic language and syntax is a bit difficult compared to other languages.
Enter the line no of the line you want to replace:2
```

```
Hi this is C programming.
This is very basic language and syntax is a bit difficult compared to other languages.
```

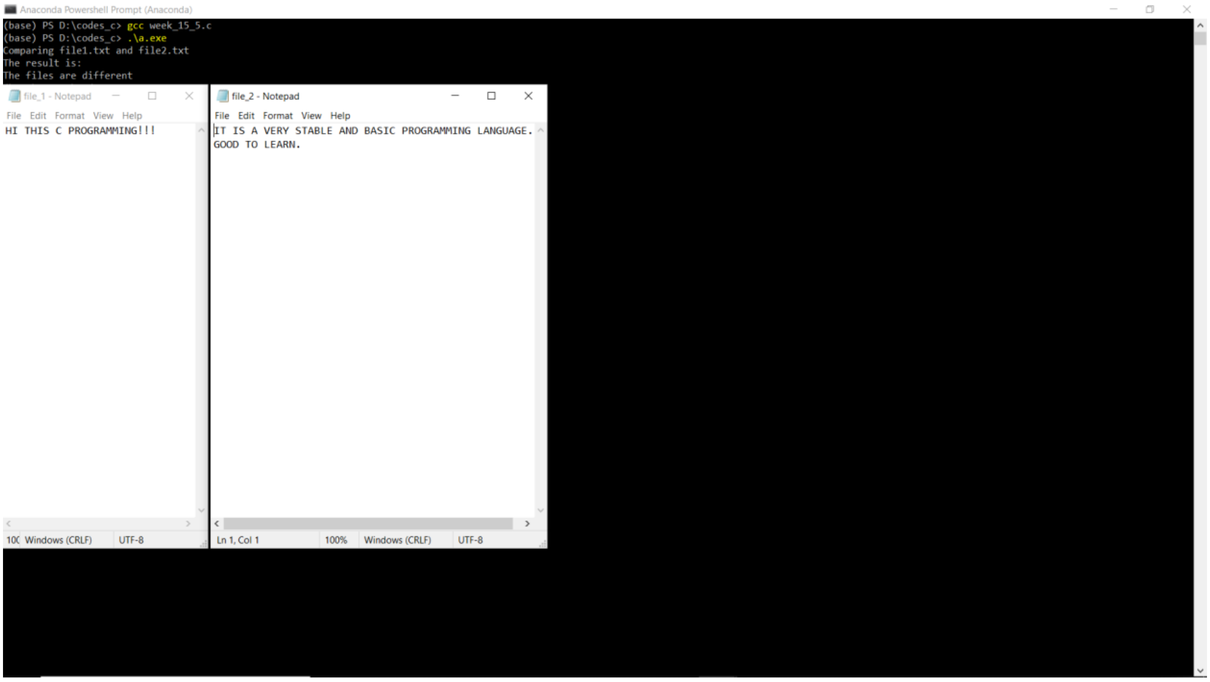
5)

**client.c:**



```
#include <stdio.h>

#include <stdlib.h>

int main(void)
{
    FILE *ptr1 = fopen("file_1.txt", "r");
    FILE *ptr2 = fopen("file_2.txt", "r");
    printf("Comparing file1.txt and file2.txt\n");
    char ch1, ch2;
    ch1 = fgetc(ptr1);
    ch2 = fgetc(ptr2);
    printf("The result is: \n");
    int y = 0;
    while (1)
    {
        if (ch1 == EOF || ch2 == EOF)
        {
            if (ch1 == ch2)
            {
                y = 0;
                break;
            }
            else
            {
                y = 1;
                break;
            }
        }
        if (ch1 != ch2)
        {
            y = 1;
            break;
        }
        ch1 = fgetc(ptr1);
        ch2 = fgetc(ptr2);
    }

    if (y == 0)
    {
        printf("The files are the same\n");
    }
    else if (y == 1)
    {
        printf("The files are different\n");
    }
    fclose(ptr1);
    fclose(ptr2);
}
```

## Output:



Anaconda Powershell Prompt (Anaconda)
file_1 - Notepad
File Edit Format View Help
10( Windows (CRLF) UTF-8
file_2 - Notepad
File Edit Format View Help
Ln 1, Col 1 100% Windows (CRLF) UTF-8

```
(base) PS D:\codes_c> gcc week_15_5.c
(base) PS D:\codes_c> .\a.exe
Comparing file1.txt and file2.txt
The result is:
The files are different
```

```
HI THIS C PROGRAMMING!!!
```

```
IT IS A VERY STABLE AND BASIC PROGRAMMING LANGUAGE.
GOOD TO LEARN.
```