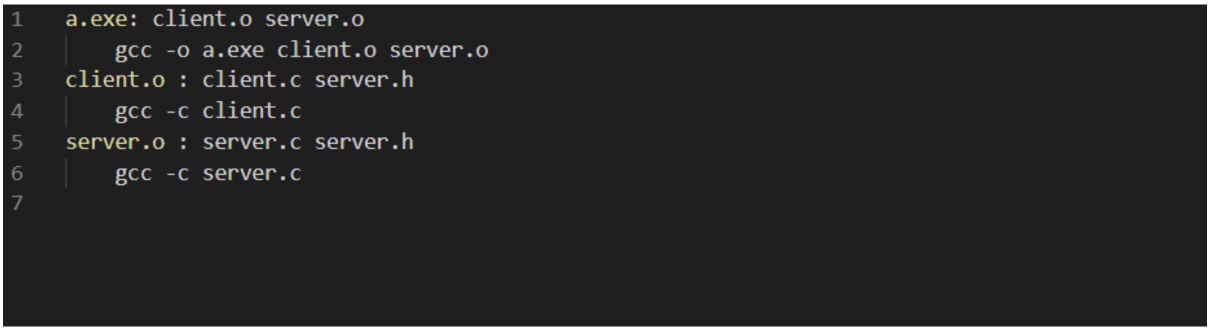
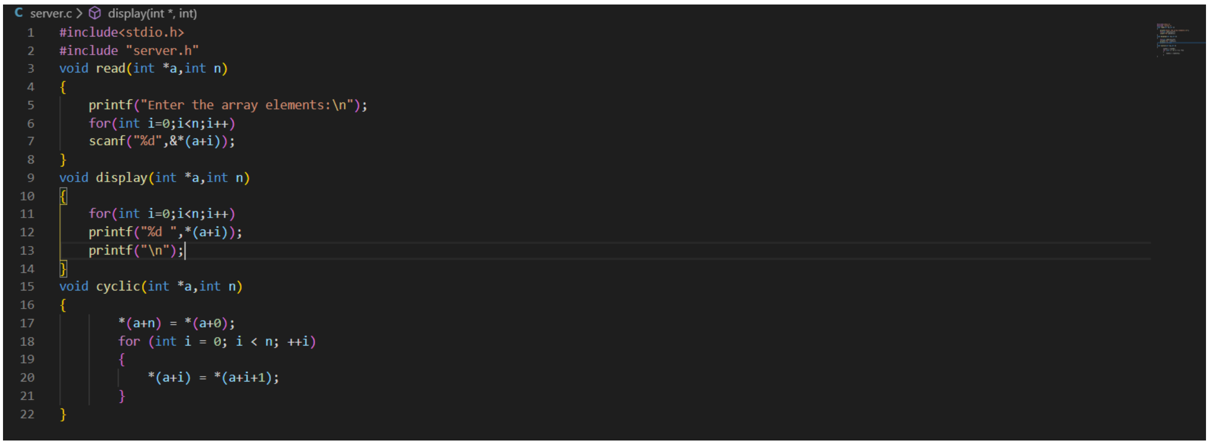
1)Problem 1:

Makefile.mk



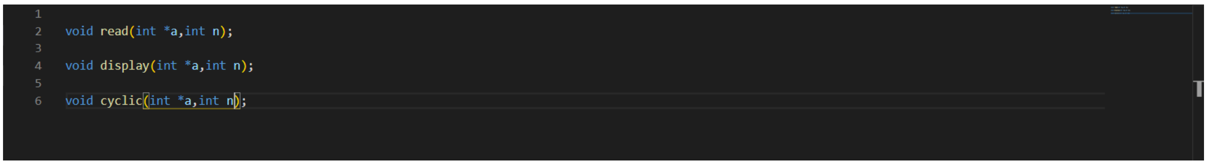
```
a.exe: client.o server.o
	gcc -o a.exe client.o server.o
client.o : client.c server.h
	gcc -c client.c
server.o : server.c server.h
	gcc -c server.c
```

Server.c



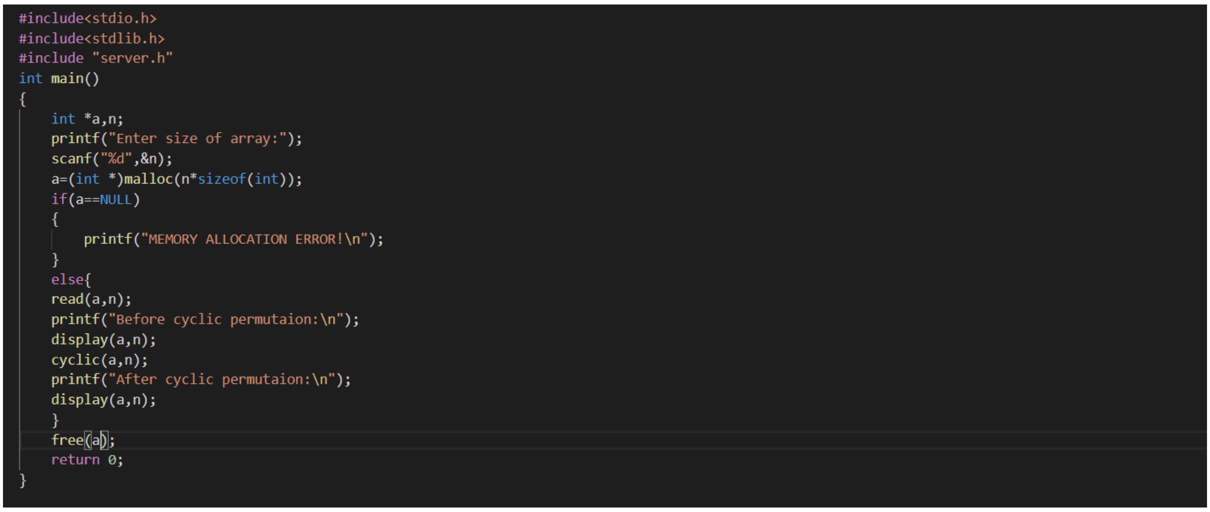
```c
#include<stdio.h>
#include "server.h"
void read(int *a,int n)
{
    printf("Enter the array elements:\n");
    for(int i=0;i<n;i++)
    scanf("%d",&*(a+i));
}
void display(int *a,int n)
{
    for(int i=0;i<n;i++)
    printf("%d ",*(a+i));
    printf("\n");
}
void cyclic(int *a,int n)
{
        *(a+n) = *(a+0);
        for (int i = 0; i < n; ++i)
        {
            *(a+i) = *(a+i+1);
        }
}
```

Server.h



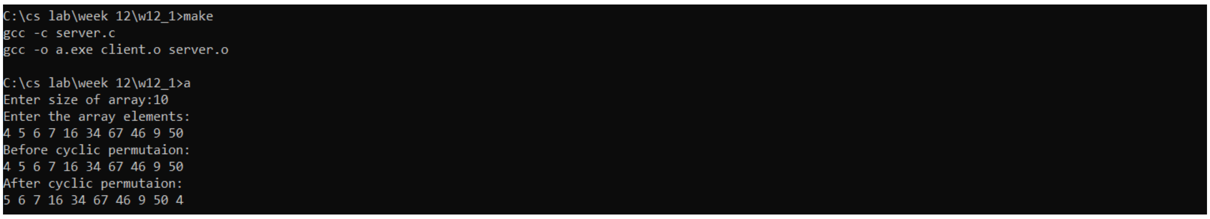
```c

void read(int *a,int n);

void display(int *a,int n);

void cyclic(int *a,int n);
```

Client.c



```c
#include<stdio.h>
#include<stdlib.h>
#include "server.h"
int main()
{
    int *a,n;
    printf("Enter size of array:");
    scanf("%d",&n);
    a=(int *)malloc(n*sizeof(int));
    if(a==NULL)
    {
        printf("MEMORY ALLOCATION ERROR!\n");
    }
    else{
    read(a,n);
    printf("Before cyclic permutaion:\n");
    display(a,n);
    cyclic(a,n);
    printf("After cyclic permutaion:\n");
    display(a,n);
    }
    free(a);
    return 0;
}
```

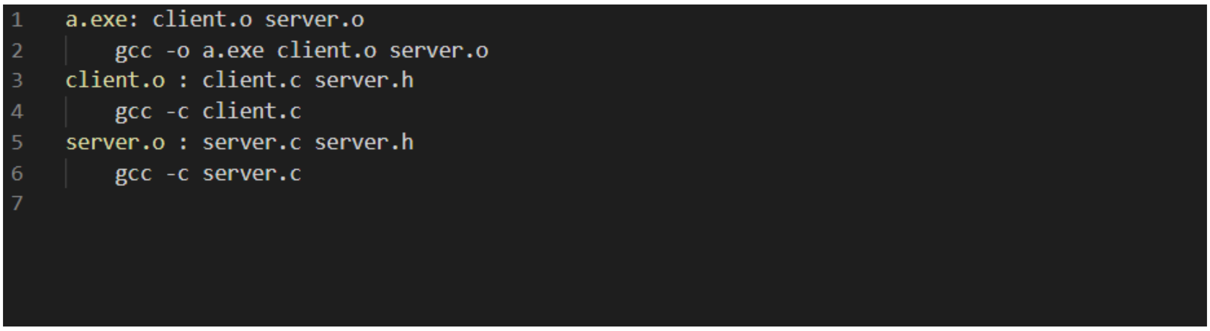
Output



```
C:\cs lab\week 12\w12_1>make
gcc -c server.c
gcc -o a.exe client.o server.o

C:\cs lab\week 12\w12_1>a
Enter size of array:10
Enter the array elements:
4 5 6 7 16 34 67 46 9 50
Before cyclic permutaion:
4 5 6 7 16 34 67 46 9 50
After cyclic permutaion:
5 6 7 16 34 67 46 9 50 4
```

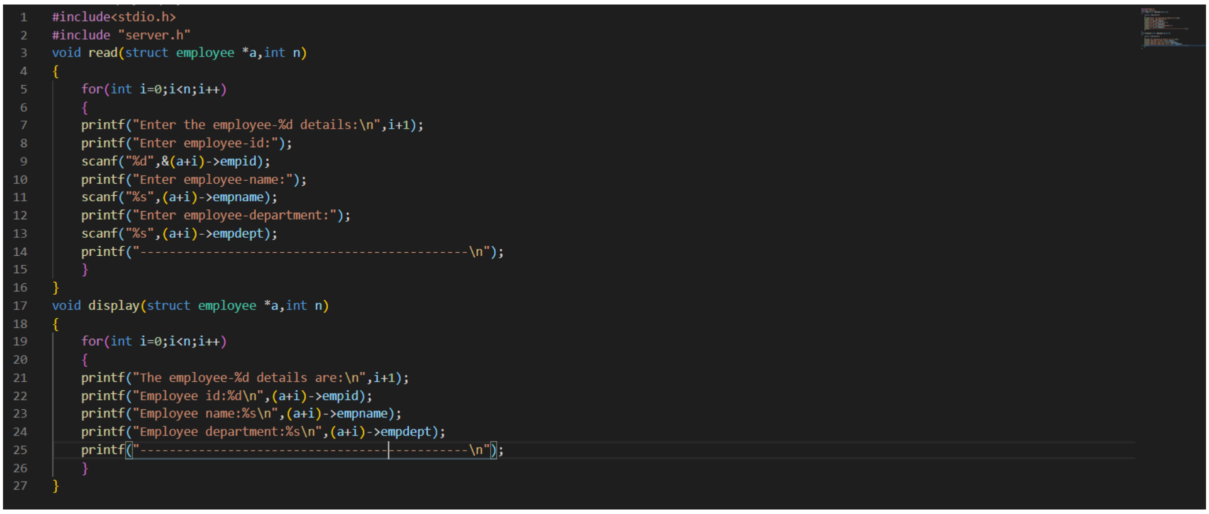
2)Problem 2:

Makefile.mk



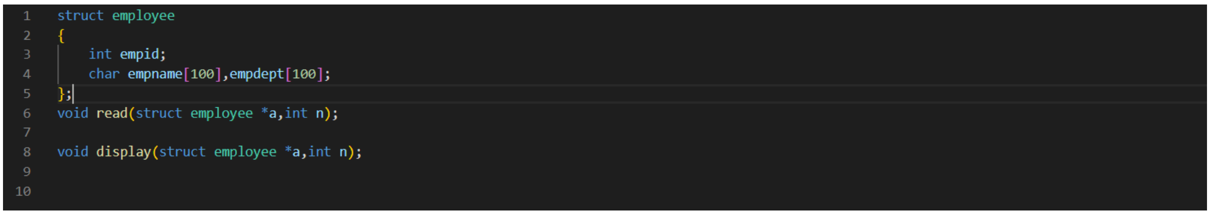
```
a.exe: client.o server.o
    gcc -o a.exe client.o server.o
client.o : client.c server.h
    gcc -c client.c
server.o : server.c server.h
    gcc -c server.c
```

Server.c



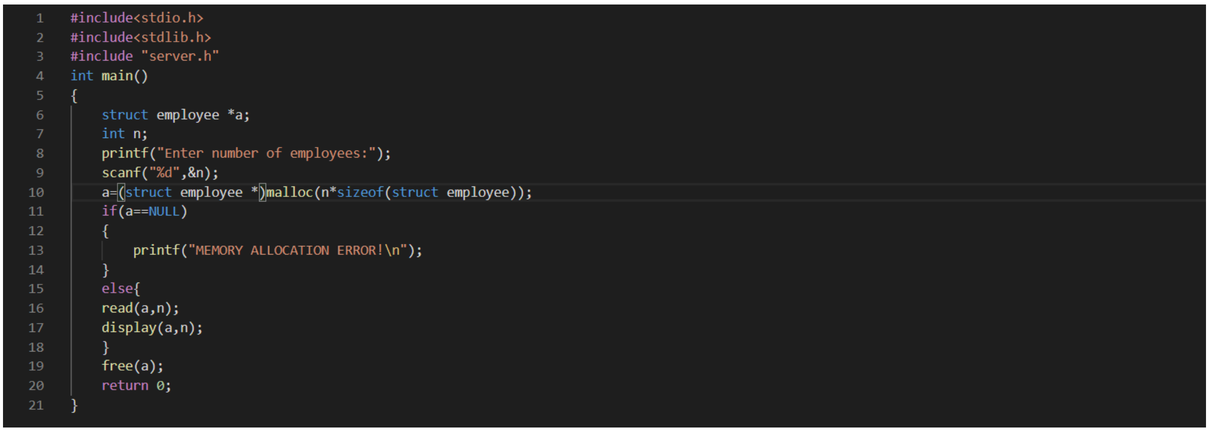
```
#include<stdio.h>
#include "server.h"
void read(struct employee *a,int n)
{
    for(int i=0;i<n;i++)
    {
    printf("Enter the employee-%d details:\n",i+1);
    printf("Enter employee-id:");
    scanf("%d",&(a+i)->empid);
    printf("Enter employee-name:");
    scanf("%s",(a+i)->empname);
    printf("Enter employee-department:");
    scanf("%s",(a+i)->empdept);
    printf("--------------------------------------------\n");
    }
}
void display(struct employee *a,int n)
{
    for(int i=0;i<n;i++)
    {
    printf("The employee-%d details are:\n",i+1);
    printf("Employee id:%d\n",(a+i)->empid);
    printf("Employee name:%s\n",(a+i)->empname);
    printf("Employee department:%s\n",(a+i)->empdept);
    printf("--------------------------------------------\n");
    }
}
```

Server.h



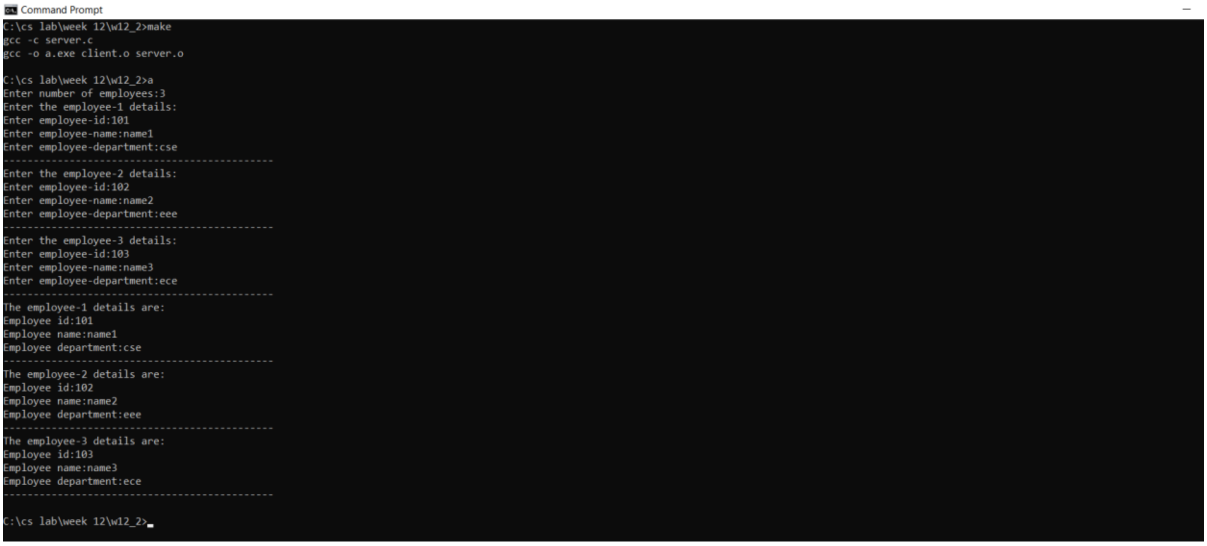
```
struct employee
{
    int empid;
    char empname[100],empdept[100];
};
void read(struct employee *a,int n);

void display(struct employee *a,int n);


```

Client.c



```c
#include<stdio.h>
#include<stdlib.h>
#include "server.h"
int main()
{
    struct employee *a;
    int n;
    printf("Enter number of employees:");
    scanf("%d",&n);
    a=(struct employee *)malloc(n*sizeof(struct employee));
    if(a==NULL)
    {
        printf("MEMORY ALLOCATION ERROR!\n");
    }
    else{
    read(a,n);
    display(a,n);
    }
    free(a);
    return 0;
}
```

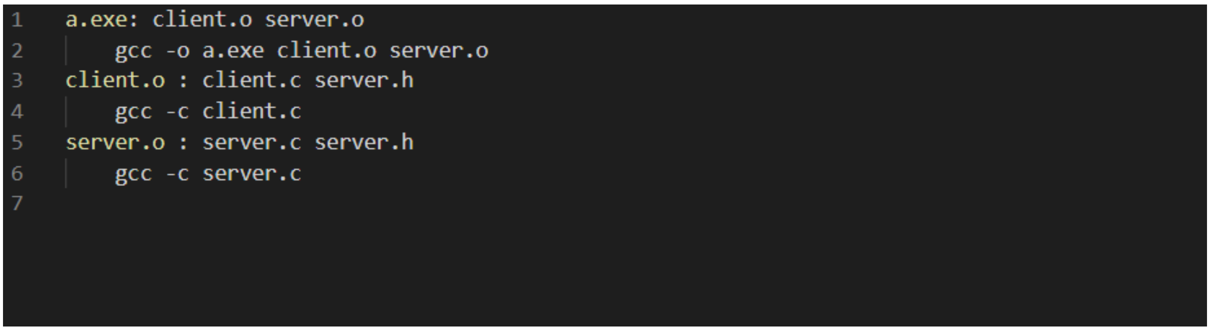
Output



```
C:\cs lab\week 12\w12_2>make
gcc -c server.c
gcc -o a.exe client.o server.o

C:\cs lab\week 12\w12_2>a
Enter number of employees:3
Enter the employee-1 details:
Enter employee-id:101
Enter employee-name:name1
Enter employee-department:cse
---------------------------------------------
Enter the employee-2 details:
Enter employee-id:102
Enter employee-name:name2
Enter employee-department:eee
---------------------------------------------
Enter the employee-3 details:
Enter employee-id:103
Enter employee-name:name3
Enter employee-department:ece
---------------------------------------------
The employee-1 details are:
Employee id:101
Employee name:name1
Employee department:cse
---------------------------------------------
The employee-2 details are:
Employee id:102
Employee name:name2
Employee department:eee
---------------------------------------------
The employee-3 details are:
Employee id:103
Employee name:name3
Employee department:ece
---------------------------------------------

C:\cs lab\week 12\w12_2>_
```

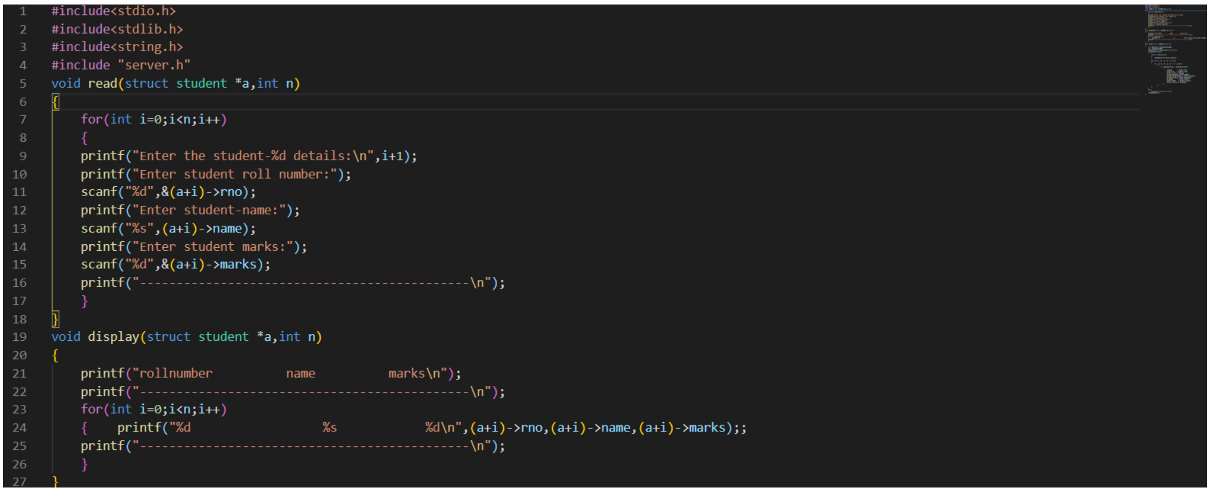
3)Problem 3:

Makefile.mk

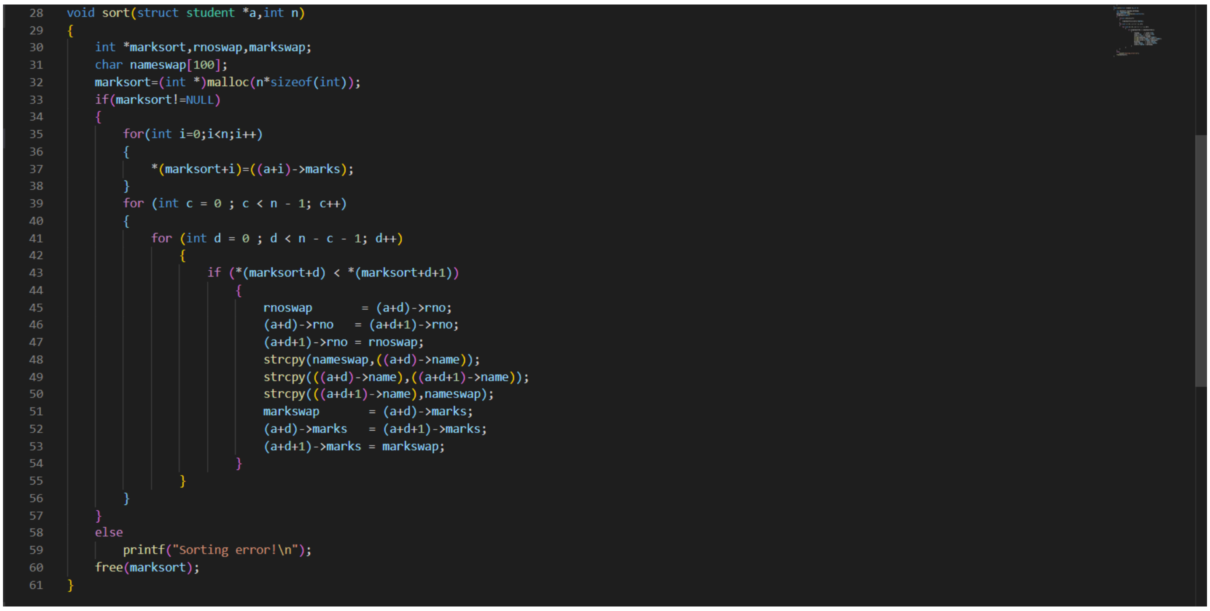


```makefile
a.exe: client.o server.o
	gcc -o a.exe client.o server.o
client.o : client.c server.h
	gcc -c client.c
server.o : server.c server.h
	gcc -c server.c
```

Server.c

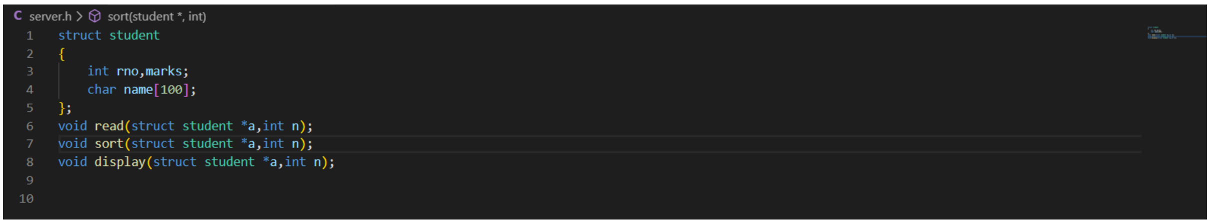


```c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include "server.h"
void read(struct student *a,int n)
{
    for(int i=0;i<n;i++)
    {
    printf("Enter the student-%d details:\n",i+1);
    printf("Enter student roll number:");
    scanf("%d",&(a+i)->rno);
    printf("Enter student-name:");
    scanf("%s",(a+i)->name);
    printf("Enter student marks:");
    scanf("%d",&(a+i)->marks);
    printf("--------------------------------------------\n");
    }
}
void display(struct student *a,int n)
{
    printf("rollnumber          name          marks\n");
    printf("--------------------------------------------\n");
    for(int i=0;i<n;i++)
    {    printf("%d                 %s            %d\n",(a+i)->rno,(a+i)->name,(a+i)->marks);;
    printf("--------------------------------------------\n");
    }
}
```



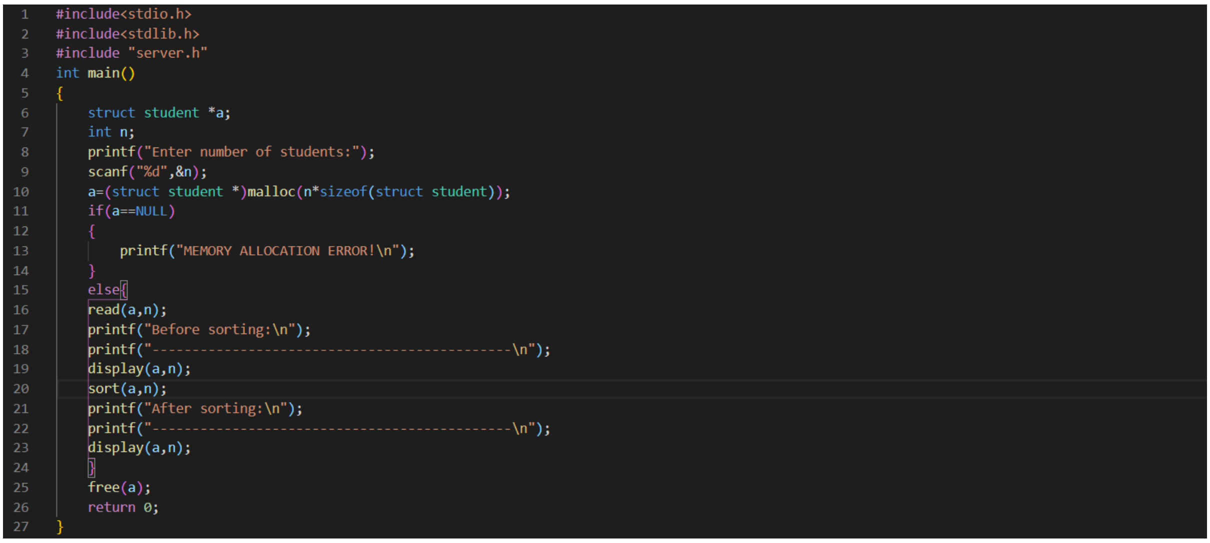
```c
void sort(struct student *a,int n)
{
    int *marksort,rnoswap,markswap;
    char nameswap[100];
    marksort=(int *)malloc(n*sizeof(int));
    if(marksort!=NULL)
    {
        for(int i=0;i<n;i++)
        {
            *(marksort+i)=((a+i)->marks);
        }
        for (int c = 0 ; c < n - 1; c++)
        {
            for (int d = 0 ; d < n - c - 1; d++)
                {
                    if (*(marksort+d) < *(marksort+d+1))
                        {
                            rnoswap       = (a+d)->rno;
                            (a+d)->rno    = (a+d+1)->rno;
                            (a+d+1)->rno = rnoswap;
                            strcpy(nameswap,((a+d)->name));
                            strcpy(((a+d)->name),((a+d+1)->name));
                            strcpy(((a+d+1)->name),nameswap);
                            markswap       = (a+d)->marks;
                            (a+d)->marks   = (a+d+1)->marks;
                            (a+d+1)->marks = markswap;
                        }
                }
        }
    }
    else
        printf("Sorting error!\n");
    free(marksort);
}
```

Server.h



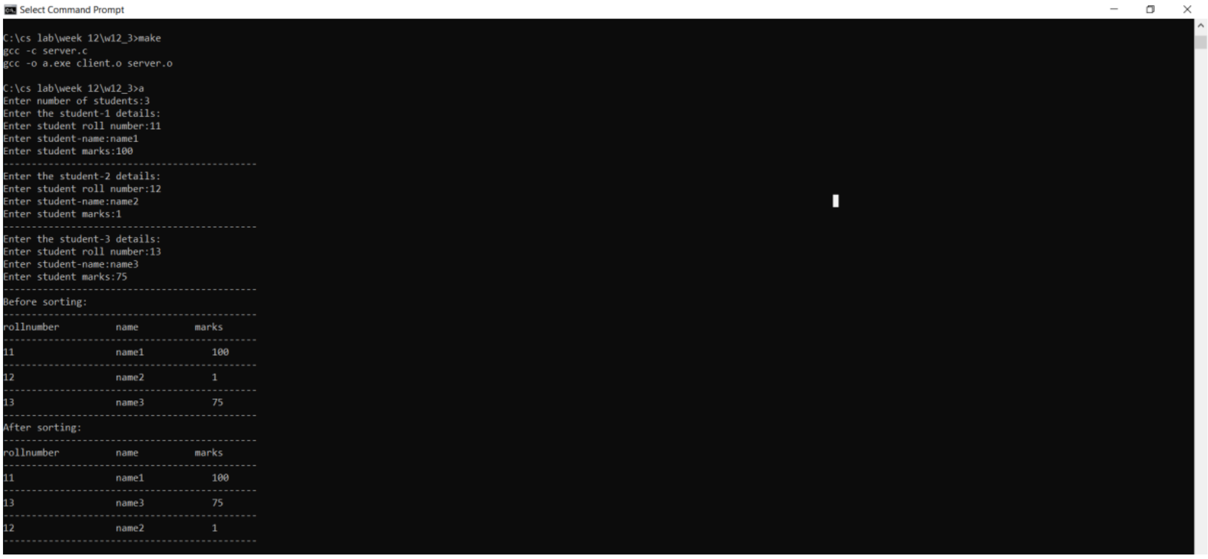
```c
struct student
{
    int rno,marks;
    char name[100];
};
void read(struct student *a,int n);
void sort(struct student *a,int n);
void display(struct student *a,int n);


```

Client.c



```c
#include<stdio.h>
#include<stdlib.h>
#include "server.h"
int main()
{
    struct student *a;
    int n;
    printf("Enter number of students:");
    scanf("%d",&n);
    a=(struct student *)malloc(n*sizeof(struct student));
    if(a==NULL)
    {
        printf("MEMORY ALLOCATION ERROR!\n");
    }
    else{
    read(a,n);
    printf("Before sorting:\n");
    printf("---------------------------------------------\n");
    display(a,n);
    sort(a,n);
    printf("After sorting:\n");
    printf("---------------------------------------------\n");
    display(a,n);
    }
    free(a);
    return 0;
}
```

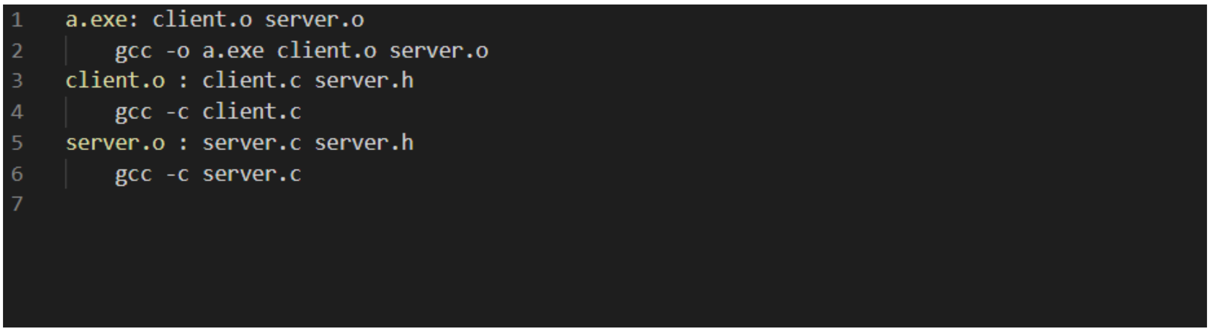
Output



```
C:\cs lab\week 12\w12_3>make
gcc -c server.c
gcc -o a.exe client.o server.o

C:\cs lab\week 12\w12_3>a
Enter number of students:3
Enter the student-1 details:
Enter student roll number:11
Enter student-name:name1
Enter student marks:100
---------------------------------------------
Enter the student-2 details:
Enter student roll number:12
Enter student-name:name2
Enter student marks:1
---------------------------------------------
Enter the student-3 details:
Enter student roll number:13
Enter student-name:name3
Enter student marks:75
---------------------------------------------
Before sorting:
---------------------------------------------
rollnumber          name          marks
---------------------------------------------
11                  name1            100
---------------------------------------------
12                  name2            1
---------------------------------------------
13                  name3            75
---------------------------------------------
After sorting:
---------------------------------------------
rollnumber          name          marks
---------------------------------------------
11                  name1            100
---------------------------------------------
13                  name3            75
---------------------------------------------
12                  name2            1
---------------------------------------------
```

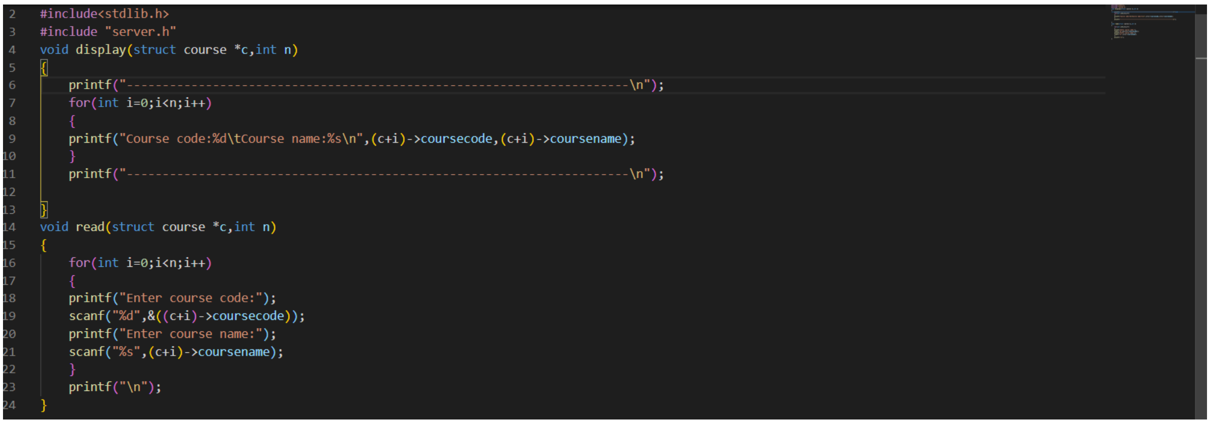
4)practice 1:

Makefile.mk



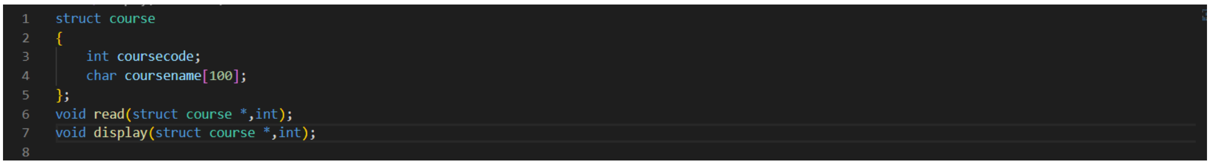
```make
a.exe: client.o server.o
	gcc -o a.exe client.o server.o
client.o : client.c server.h
	gcc -c client.c
server.o : server.c server.h
	gcc -c server.c

```

Server.c



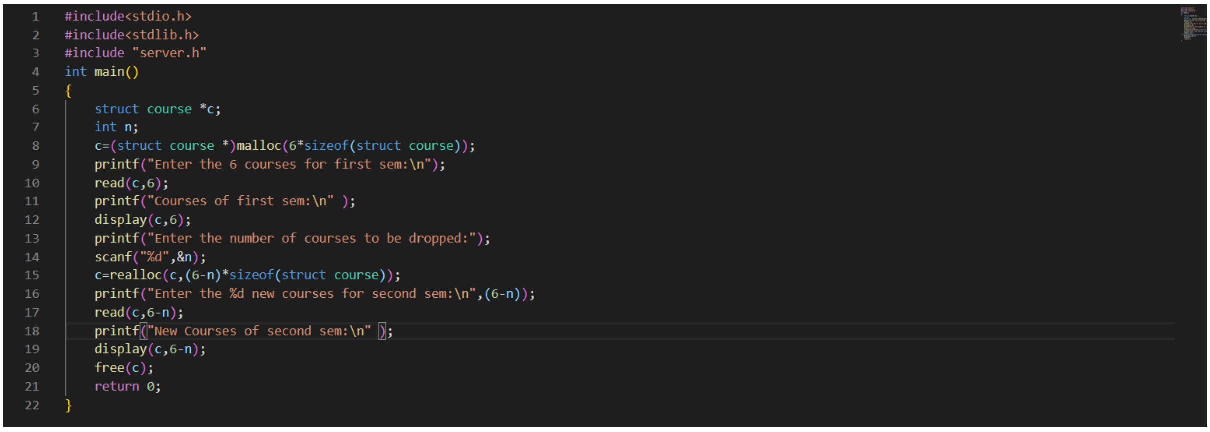
```c
#include<stdlib.h>
#include "server.h"
void display(struct course *c,int n)
{
    printf("----------------------------------------------------------------------\n");
    for(int i=0;i<n;i++)
    {
    printf("Course code:%d\tCourse name:%s\n",(c+i)->coursecode,(c+i)->coursename);
    }
    printf("----------------------------------------------------------------------\n");

}
void read(struct course *c,int n)
{
    for(int i=0;i<n;i++)
    {
    printf("Enter course code:");
    scanf("%d",&((c+i)->coursecode));
    printf("Enter course name:");
    scanf("%s",(c+i)->coursename);
    }
    printf("\n");
}
```

Server.h



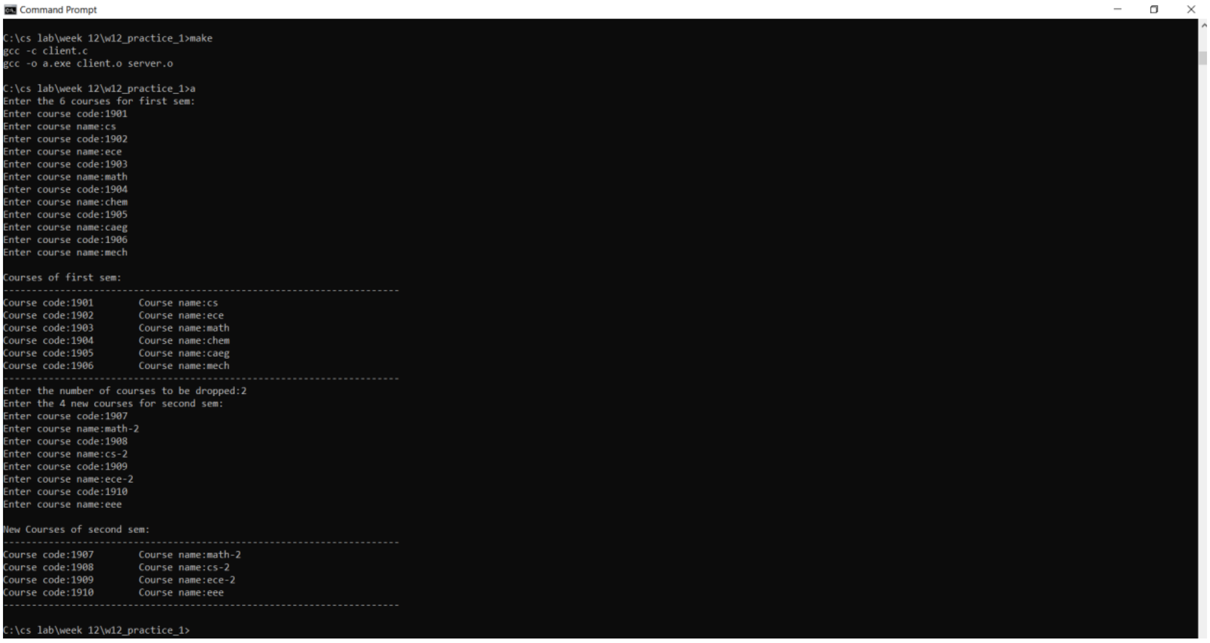
```c
struct course
{
    int coursecode;
    char coursename[100];
};
void read(struct course *,int);
void display(struct course *,int);

```

Client.c



```c
#include<stdio.h>
#include<stdlib.h>
#include "server.h"
int main()
{
    struct course *c;
    int n;
    c=(struct course *)malloc(6*sizeof(struct course));
    printf("Enter the 6 courses for first sem:\n");
    read(c,6);
    printf("Courses of first sem:\n" );
    display(c,6);
    printf("Enter the number of courses to be dropped:");
    scanf("%d",&n);
    c=realloc(c,(6-n)*sizeof(struct course));
    printf("Enter the %d new courses for second sem:\n",(6-n));
    read(c,6-n);
    printf("New Courses of second sem:\n" );
    display(c,6-n);
    free(c);
    return 0;
}
```

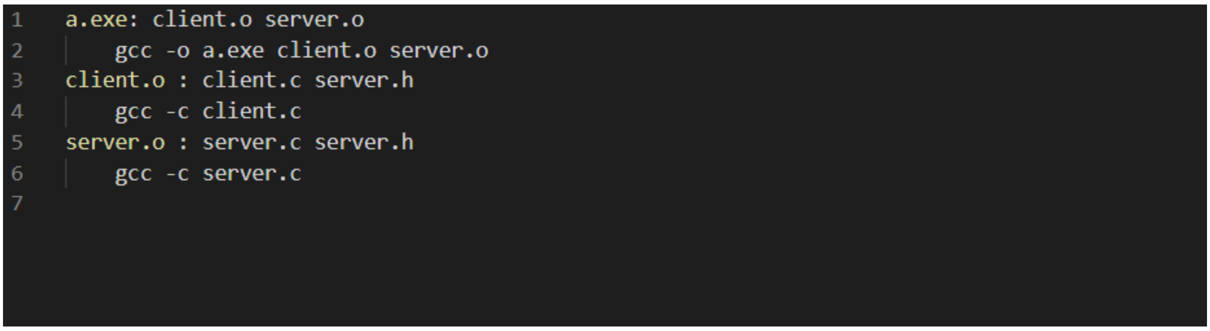
Output



```
C:\cs lab\week 12\w12_practice_1>make
gcc -c client.c
gcc -o a.exe client.o server.o

C:\cs lab\week 12\w12_practice_1>a
Enter the 6 courses for first sem:
Enter course code:1901
Enter course name:cs
Enter course code:1902
Enter course name:ece
Enter course code:1903
Enter course name:math
Enter course code:1904
Enter course name:chem
Enter course code:1905
Enter course name:caeg
Enter course code:1906
Enter course name:mech

Courses of first sem:
----------------------------------------------------------------------
Course code:1901        Course name:cs
Course code:1902        Course name:ece
Course code:1903        Course name:math
Course code:1904        Course name:chem
Course code:1905        Course name:caeg
Course code:1906        Course name:mech
----------------------------------------------------------------------
Enter the number of courses to be dropped:2
Enter the 4 new courses for second sem:
Enter course code:1907
Enter course name:math-2
Enter course code:1908
Enter course name:cs-2
Enter course code:1909
Enter course name:ece-2
Enter course code:1910
Enter course name:eee

New Courses of second sem:
----------------------------------------------------------------------
Course code:1907        Course name:math-2
Course code:1908        Course name:cs-2
Course code:1909        Course name:ece-2
Course code:1910        Course name:eee
----------------------------------------------------------------------

C:\cs lab\week 12\w12_practice_1>
```

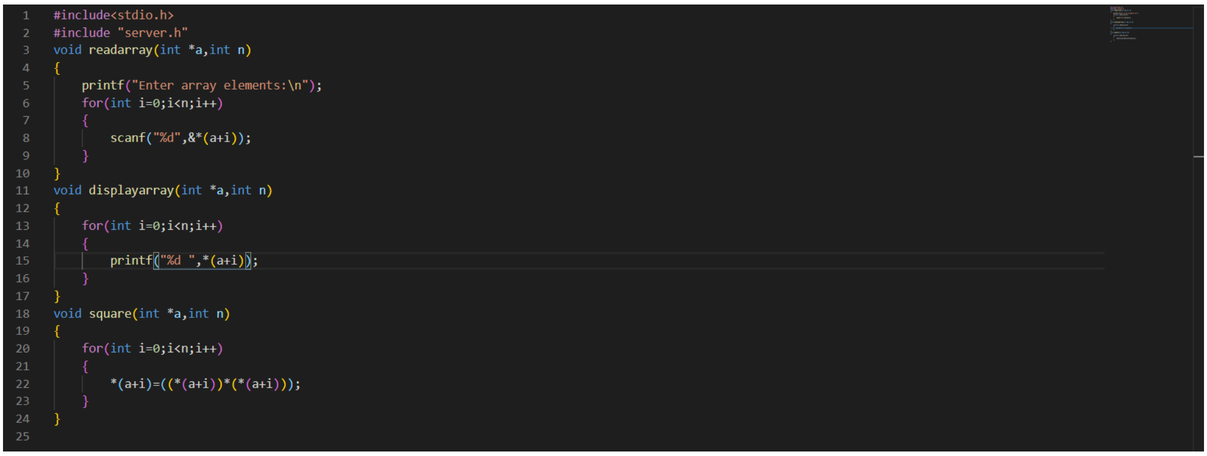
5)Practice 2:

Makefile.mk



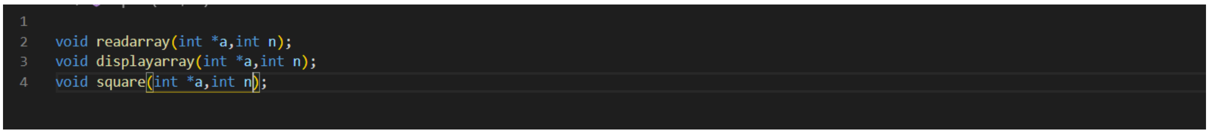
```
a.exe: client.o server.o
	gcc -o a.exe client.o server.o
client.o : client.c server.h
	gcc -c client.c
server.o : server.c server.h
	gcc -c server.c

```

Server.c



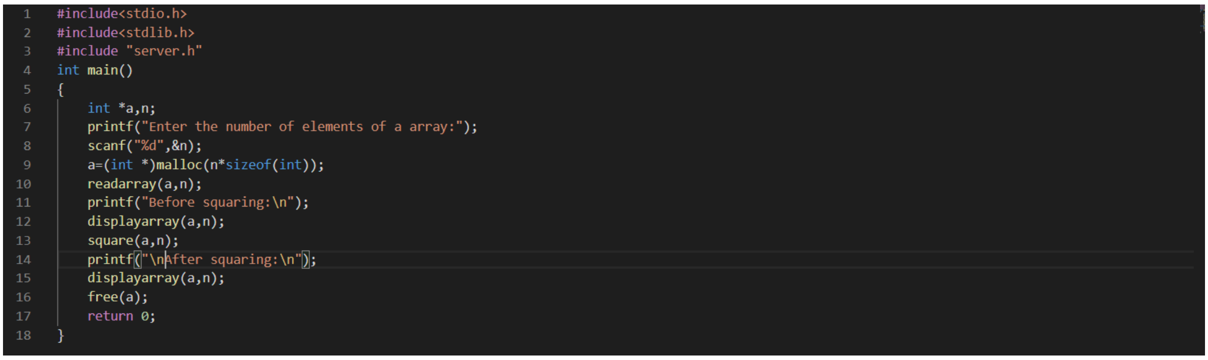
```c
#include<stdio.h>
#include "server.h"
void readarray(int *a,int n)
{
    printf("Enter array elements:\n");
    for(int i=0;i<n;i++)
    {
        scanf("%d",&*(a+i));
    }
}
void displayarray(int *a,int n)
{
    for(int i=0;i<n;i++)
    {
        printf("%d ",*(a+i));
    }
}
void square(int *a,int n)
{
    for(int i=0;i<n;i++)
    {
        *(a+i)=((*(a+i))*(*(a+i)));
    }
}

```

Server.h



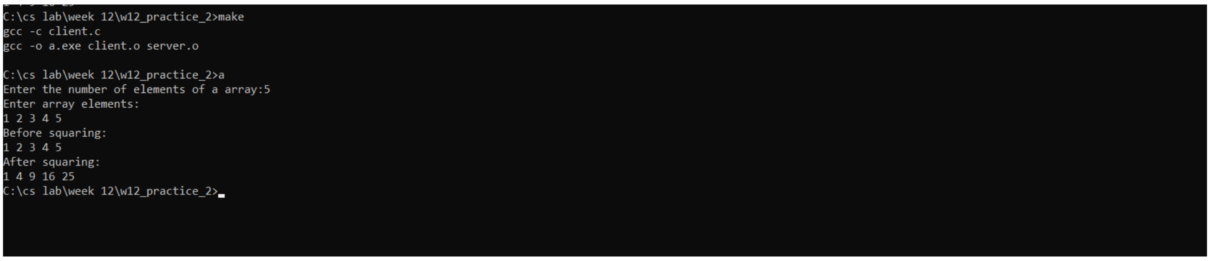
```c

void readarray(int *a,int n);
void displayarray(int *a,int n);
void square(int *a,int n);
```

Client.c



```c
#include<stdio.h>
#include<stdlib.h>
#include "server.h"
int main()
{
    int *a,n;
    printf("Enter the number of elements of a array:");
    scanf("%d",&n);
    a=(int *)malloc(n*sizeof(int));
    readarray(a,n);
    printf("Before squaring:\n");
    displayarray(a,n);
    square(a,n);
    printf("\nAfter squaring:\n");
    displayarray(a,n);
    free(a);
    return 0;
}
```

Output



```
C:\cs lab\week 12\w12_practice_2>make
gcc -c client.c
gcc -o a.exe client.o server.o

C:\cs lab\week 12\w12_practice_2>a
Enter the number of elements of a array:5
Enter array elements:
1 2 3 4 5
Before squaring:
1 2 3 4 5
After squaring:
1 4 9 16 25
C:\cs lab\week 12\w12_practice_2>
```