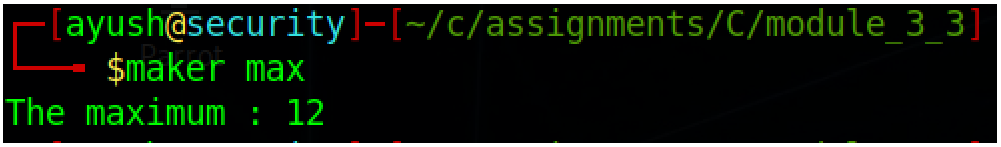
MODULE: 3.3 (File Handling and Debugging)

1. Write a program to find out the max number from given array using function

Code:

```c
#include <stdio.h>

int maxarr(int arr[], int size);
// Declaring prototype of func

int main() {
   int arr[] = {5, 8, 3, 12, 7, 10};
   // Integer array

   int size = sizeof(arr) / sizeof(arr[0]);
   // Calculating the length of the array

   int maxNumber = maxarr(arr, size);
   // Printing the max in the array
   printf("The maximum : %d\n", maxNumber);

   return 0;
}

int maxarr(int arr[], int size) {
   int max = arr[0];
   // Setting the first element as the max

   for (int i = 1; i < size; i++) {
       // Iterating through all elements in the array
       if (arr[i] > max) {
           // If array of i'th index is greater than max element than set the max as the i'th index
           max = arr[i];
       }
   }

   return max;
}
```

Output:

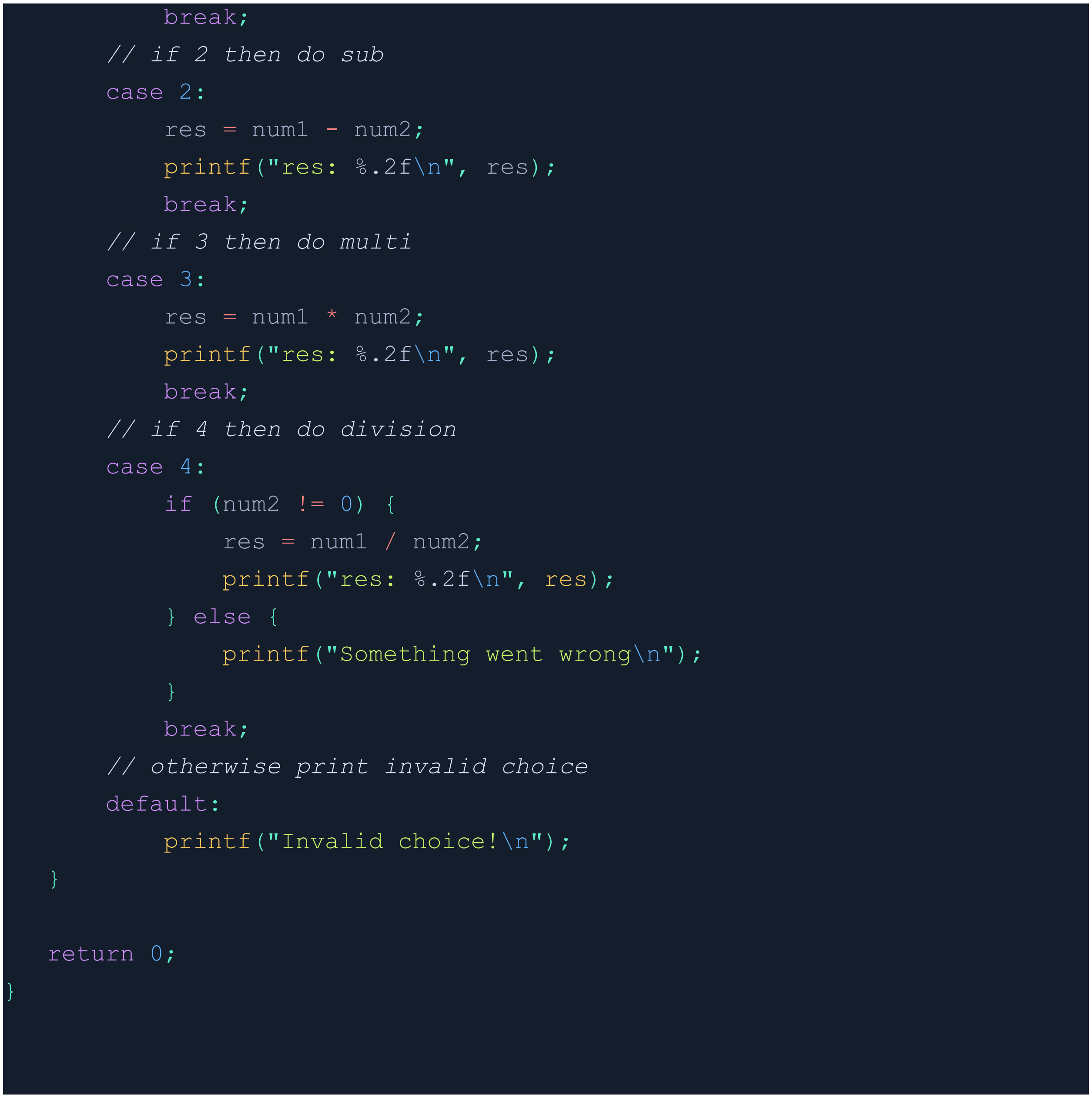


```
┌─[ayush@security]─[~/c/assignments/C/module_3_3]
└──╼ $maker max
The maximum : 12
```

2. WAP of Addition, Subtraction, Multiplication and Division using Switch case.(Must Be Menu Driven

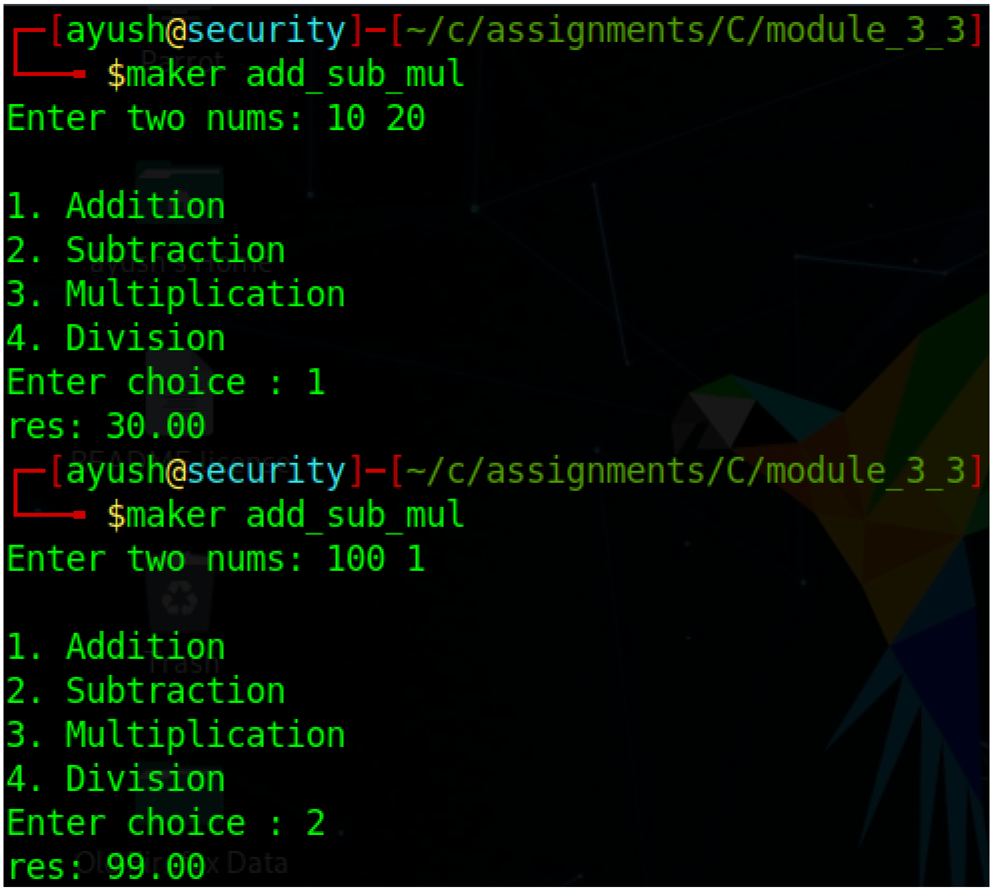
Code:

```c
#include <stdio.h>

int main() {
   // Storing choice in choice var
   int choice;
   // Taking two numbers and storing in results in float type
   float num1, num2, res;

   // Asking for input
   printf("Enter two nums: ");
   scanf("%f %f", &num1,&num2);

   // Priting meny
   printf("\n1. Addition\n");
   printf("2. Subtraction\n");
   printf("3. Multiplication\n");
   printf("4. Division\n");

   printf("Enter choice : ");
   scanf("%d", &choice);

   switch (choice) {
       // If choice is 1 then do sum
       case 1:
           res = num1 + num2;
           printf("res: %.2f\n", res);
```



```
            break;
        // if 2 then do sub
        case 2:
            res = num1 - num2;
            printf("res: %.2f\n", res);
            break;
        // if 3 then do multi
        case 3:
            res = num1 * num2;
            printf("res: %.2f\n", res);
            break;
        // if 4 then do division
        case 4:
            if (num2 != 0) {
                res = num1 / num2;
                printf("res: %.2f\n", res);
            } else {
                printf("Something went wrong\n");
            }
            break;
        // otherwise print invalid choice
        default:
            printf("Invalid choice!\n");
    }

    return 0;
}
```

Output:



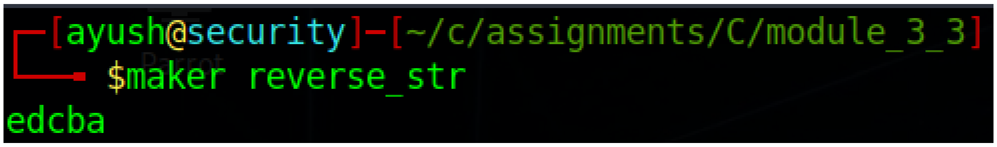
```
┌─[ayush@security]─[~/c/assignments/C/module_3_3]
└──╼ $maker add_sub_mul
Enter two nums: 10 20

1. Addition
2. Subtraction
3. Multiplication
4. Division
Enter choice : 1
res: 30.00
┌─[ayush@security]─[~/c/assignments/C/module_3_3]
└──╼ $maker add_sub_mul
Enter two nums: 100 1

1. Addition
2. Subtraction
3. Multiplication
4. Division
Enter choice : 2
res: 99.00
```

3. WAP to find reverse of string using recursion

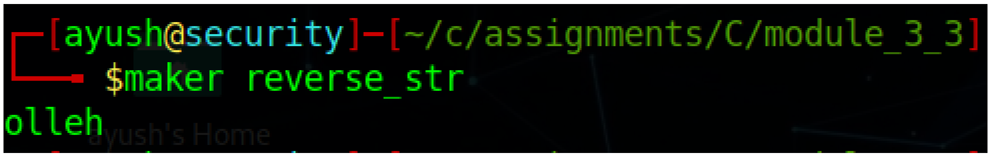
Code:

```
#include <stdio.h>

void reverse(char abc[], int length) {
   // Setting the base case if the length is always greater than or eq to
0
   if (length >= 0) {
       // print from the last element
       printf("%c", abc[length]);
       // Decrement the length by everytime
       reverse(abc, length - 1);
   }
}

int main(void) {
   // Passing the array and length
   char abc[] = "abcde";
   reverse(abc, 4);

   return 0;
}
```

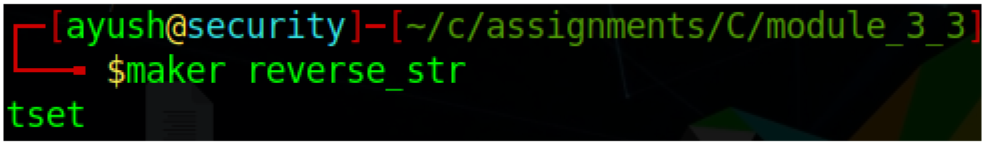
Output:



```
┌─[ayush@security]─[~/c/assignments/C/module_3_3]
└──╼ $maker reverse_str
edcba
```



```
┌─[ayush@security]─[~/c/assignments/C/module_3_3]
└──╼ $maker reverse_str
olleh
```



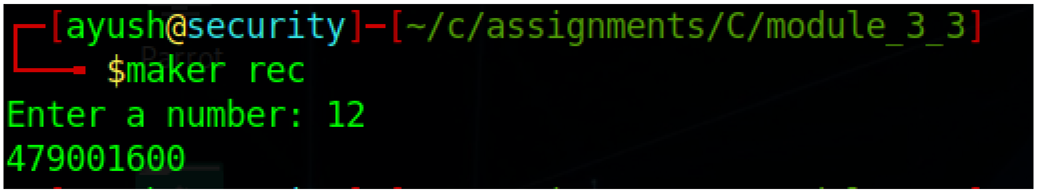
```
┌─[ayush@security]─[~/c/assignments/C/module_3_3]
└──╼ $maker reverse_str
tset
```

4. WAP to find factorial using recursion

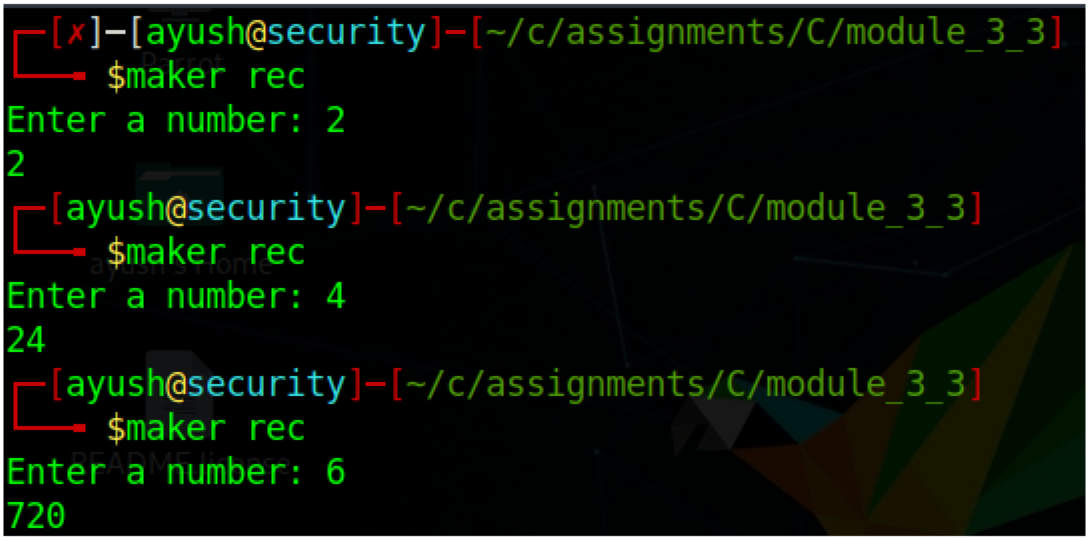
Code:

```c
#include <stdio.h>

// Defining the function
int recur(int n){
    // Setting the base case when to exit the function
    if(n==1){
        return 1;
    }

    // Otherwise just do multiplication of n * n-1;
    else{
        return n*recur(n-1);
    }
}



int main(void){
    // Variable to store n
    int n;

    // Asking user for input
    printf("Enter a number: ");
    scanf("%d",&n);

    // Printing the factorial

    printf("%d",recur(n));
    printf("\n");

}
```

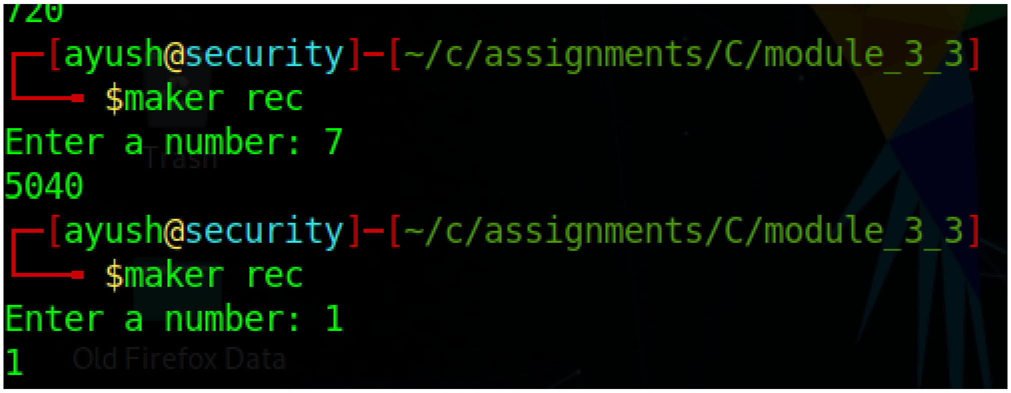
Output:



```
┌─[ayush@security]─[~/c/assignments/C/module_3_3]
└──╼ $maker rec
Enter a number: 12
479001600
```



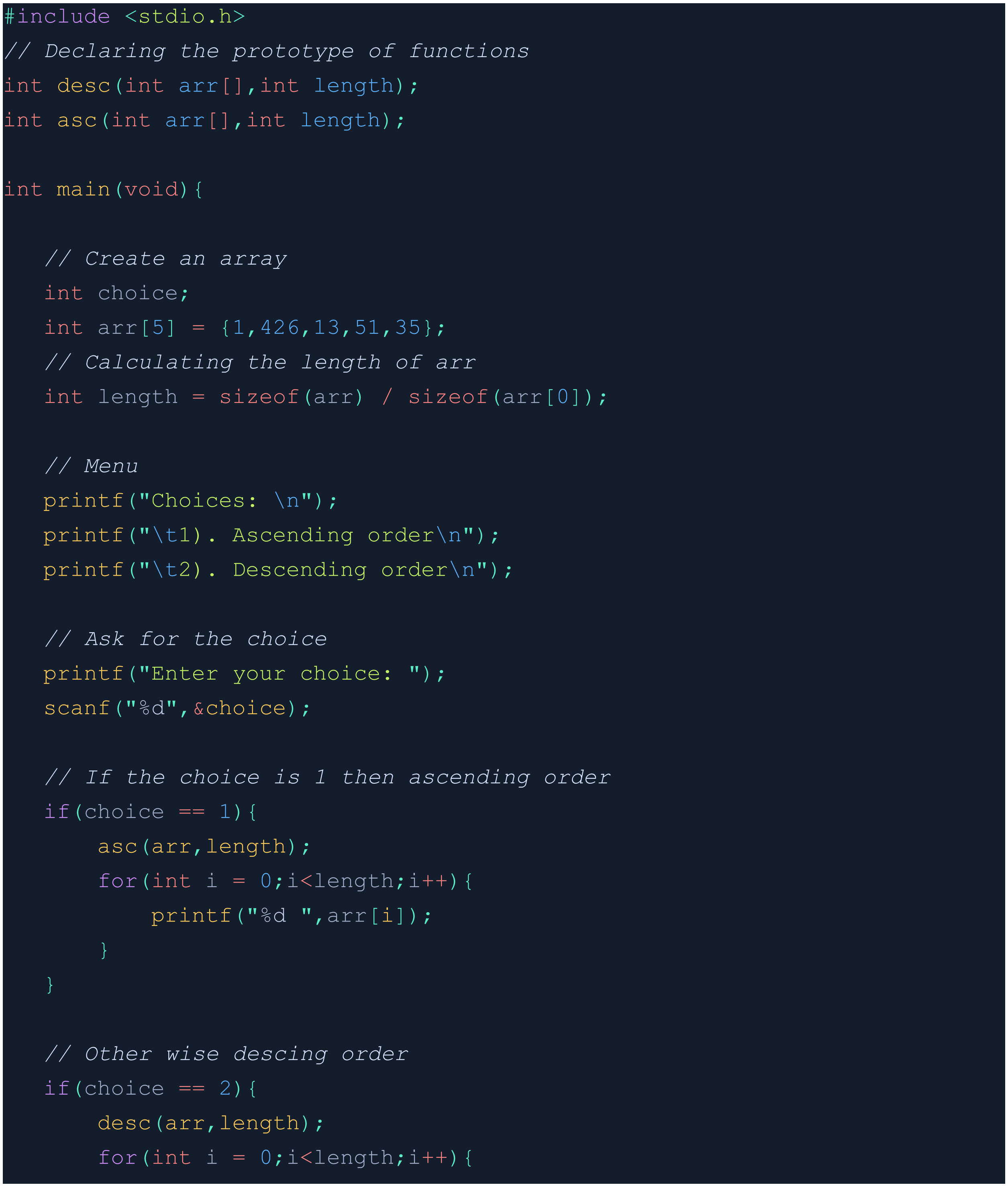
```
┌─[✗]─[ayush@security]─[~/c/assignments/C/module_3_3]
└──╼ $maker rec
Enter a number: 2
2
┌─[ayush@security]─[~/c/assignments/C/module_3_3]
└──╼ $maker rec
Enter a number: 4
24
┌─[ayush@security]─[~/c/assignments/C/module_3_3]
└──╼ $maker rec
Enter a number: 6
720
```



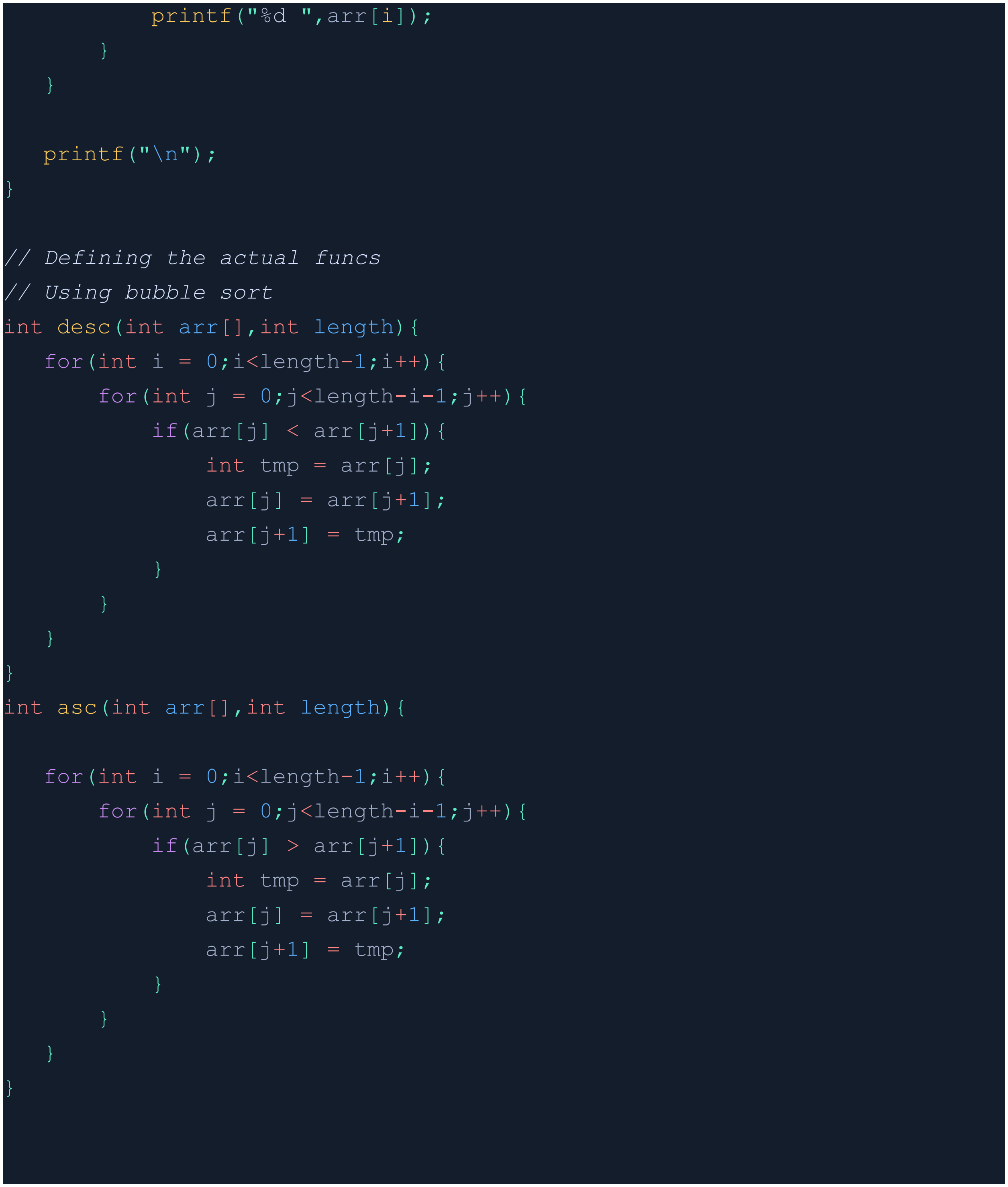
```
720
┌─[ayush@security]─[~/c/assignments/C/module_3_3]
└──╼ $maker rec
Enter a number: 7
5040
┌─[ayush@security]─[~/c/assignments/C/module_3_3]
└──╼ $maker rec
Enter a number: 1
1
```

5. WAP to take two Array input from user and sort them in ascending or descending order as per user’s choice

Code:

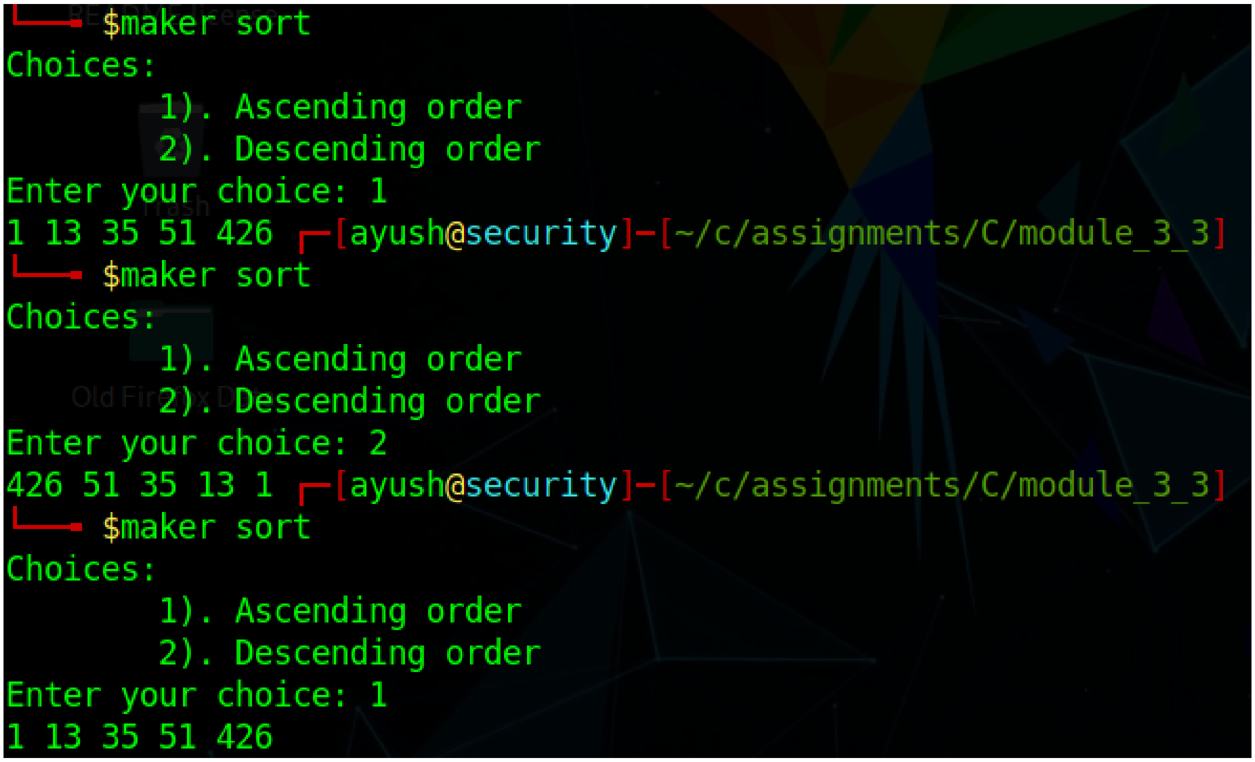


```
#include <stdio.h>
// Declaring the prototype of functions
int desc(int arr[],int length);
int asc(int arr[],int length);

int main(void){

   // Create an array
   int choice;
   int arr[5] = {1,426,13,51,35};
   // Calculating the length of arr
   int length = sizeof(arr) / sizeof(arr[0]);

   // Menu
   printf("Choices: \n");
   printf("\t1). Ascending order\n");
   printf("\t2). Descending order\n");

   // Ask for the choice
   printf("Enter your choice: ");
   scanf("%d",&choice);

   // If the choice is 1 then ascending order
   if(choice == 1){
       asc(arr,length);
       for(int i = 0;i<length;i++){
           printf("%d ",arr[i]);
       }
   }

   // Other wise descing order
   if(choice == 2){
       desc(arr,length);
       for(int i = 0;i<length;i++){
```



```
            printf("%d ",arr[i]);
        }
    }

    printf("\n");
}

// Defining the actual funcs
// Using bubble sort
int desc(int arr[],int length){
    for(int i = 0;i<length-1;i++){
        for(int j = 0;j<length-i-1;j++){
            if(arr[j] < arr[j+1]){
                int tmp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = tmp;
            }
        }
    }
}
int asc(int arr[],int length){

    for(int i = 0;i<length-1;i++){
        for(int j = 0;j<length-i-1;j++){
            if(arr[j] > arr[j+1]){
                int tmp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = tmp;
            }
        }
    }
}
```

Output:

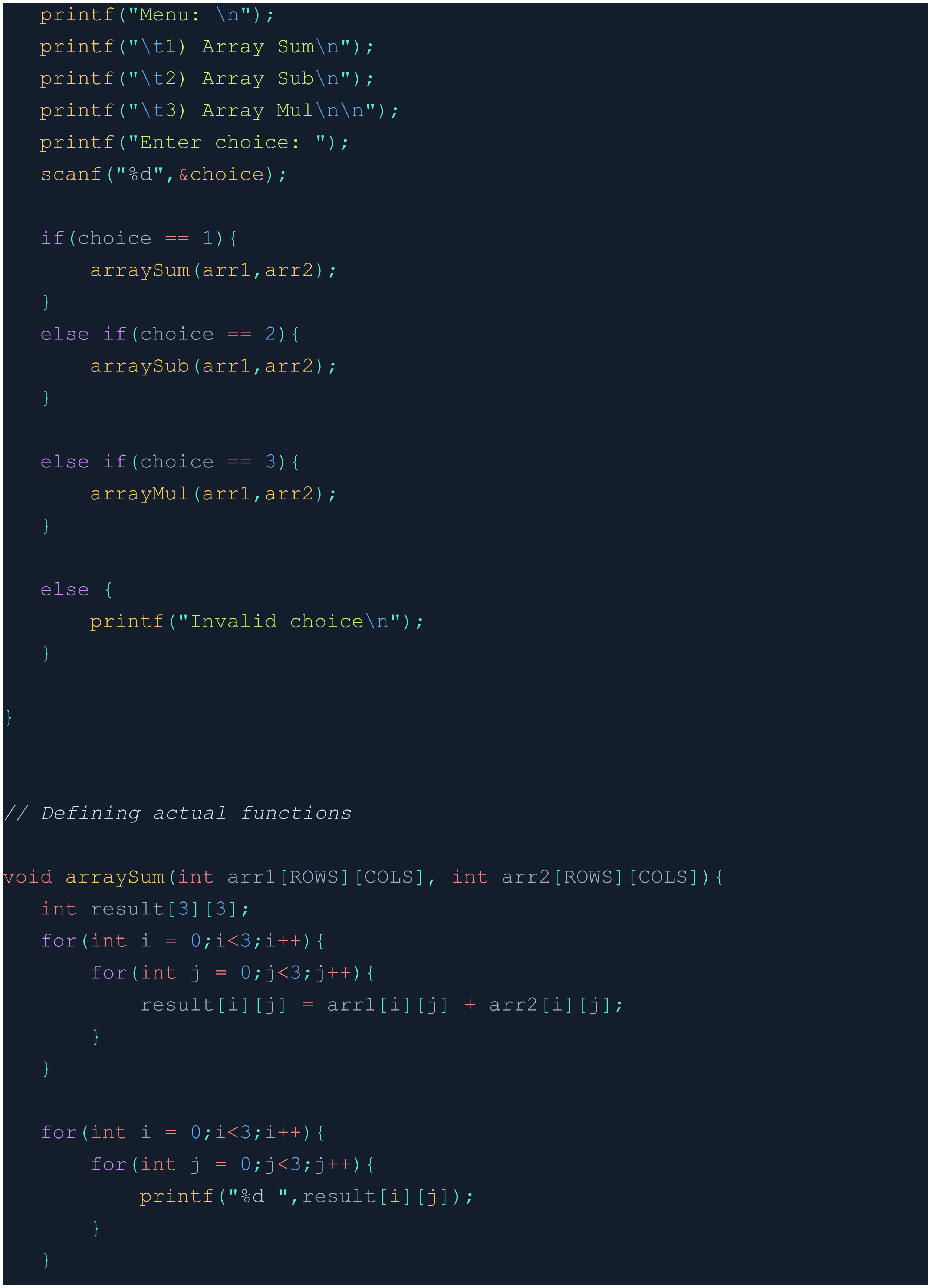


```
└──■ $maker sort
Choices:
        1). Ascending order
        2). Descending order
Enter your choice: 1
1 13 35 51 426 ┌─[ayush@security]─[~/c/assignments/C/module_3_3]
└──■ $maker sort
Choices:
        1). Ascending order
        2). Descending order
Enter your choice: 2
426 51 35 13 1 ┌─[ayush@security]─[~/c/assignments/C/module_3_3]
└──■ $maker sort
Choices:
        1). Ascending order
        2). Descending order
Enter your choice: 1
1 13 35 51 426
```

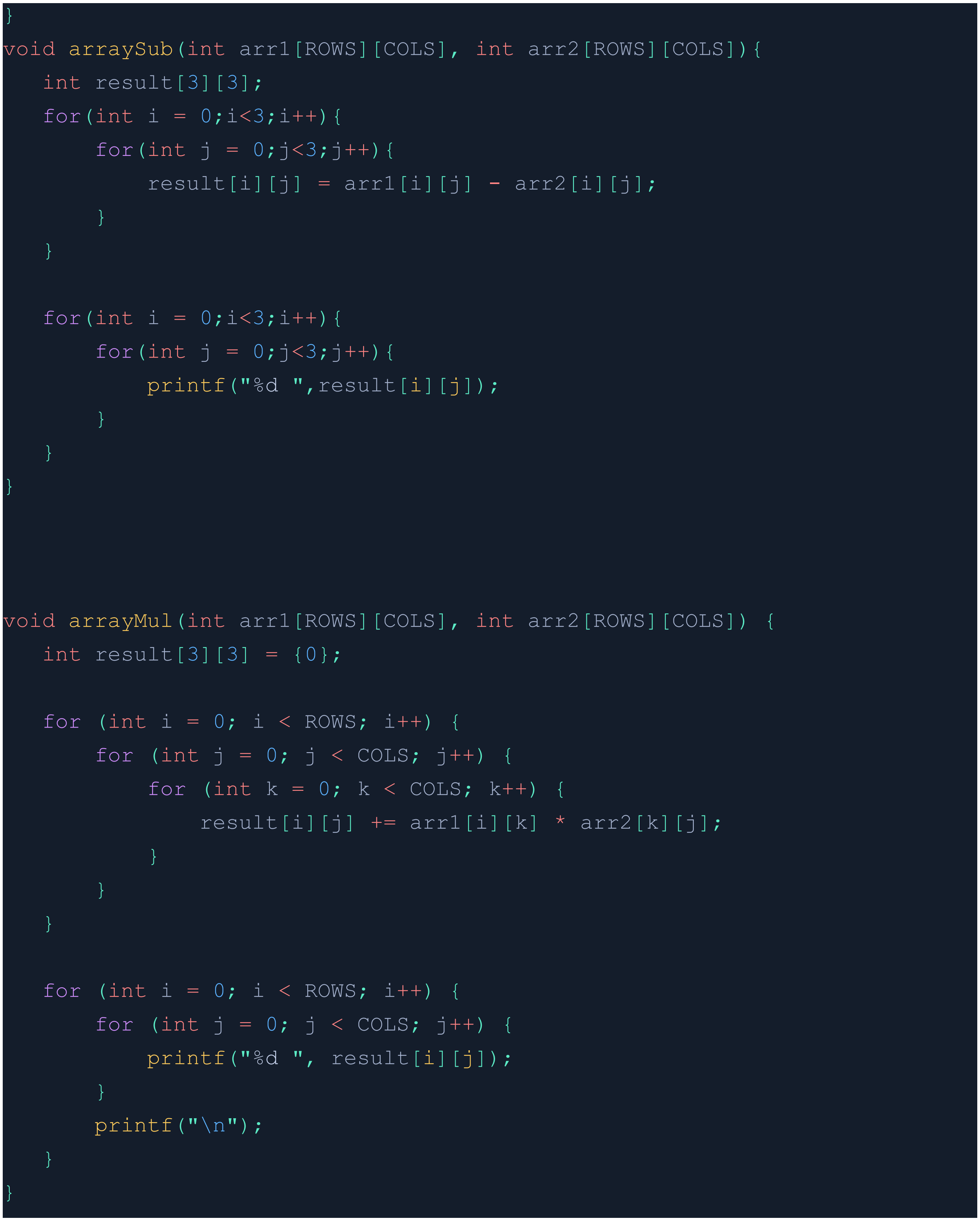
6. WAP to make addition, Subtraction and multiplication of two matrix using 2-D Array

Code:

```c
#include <stdio.h>



#define ROWS 3
#define COLS 3

// Declaring prototypes
void arraySum(int arr1[ROWS][COLS], int arr2[ROWS][COLS]);
void arraySub(int arr1[ROWS][COLS], int arr2[ROWS][COLS]);
void arrayMul(int arr1[ROWS][COLS], int arr2[ROWS][COLS]);



int main(void){
   // Storing menu item in Choice
   int choice;
   int arr1[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
   int arr2[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
   // Asking user for choice
```

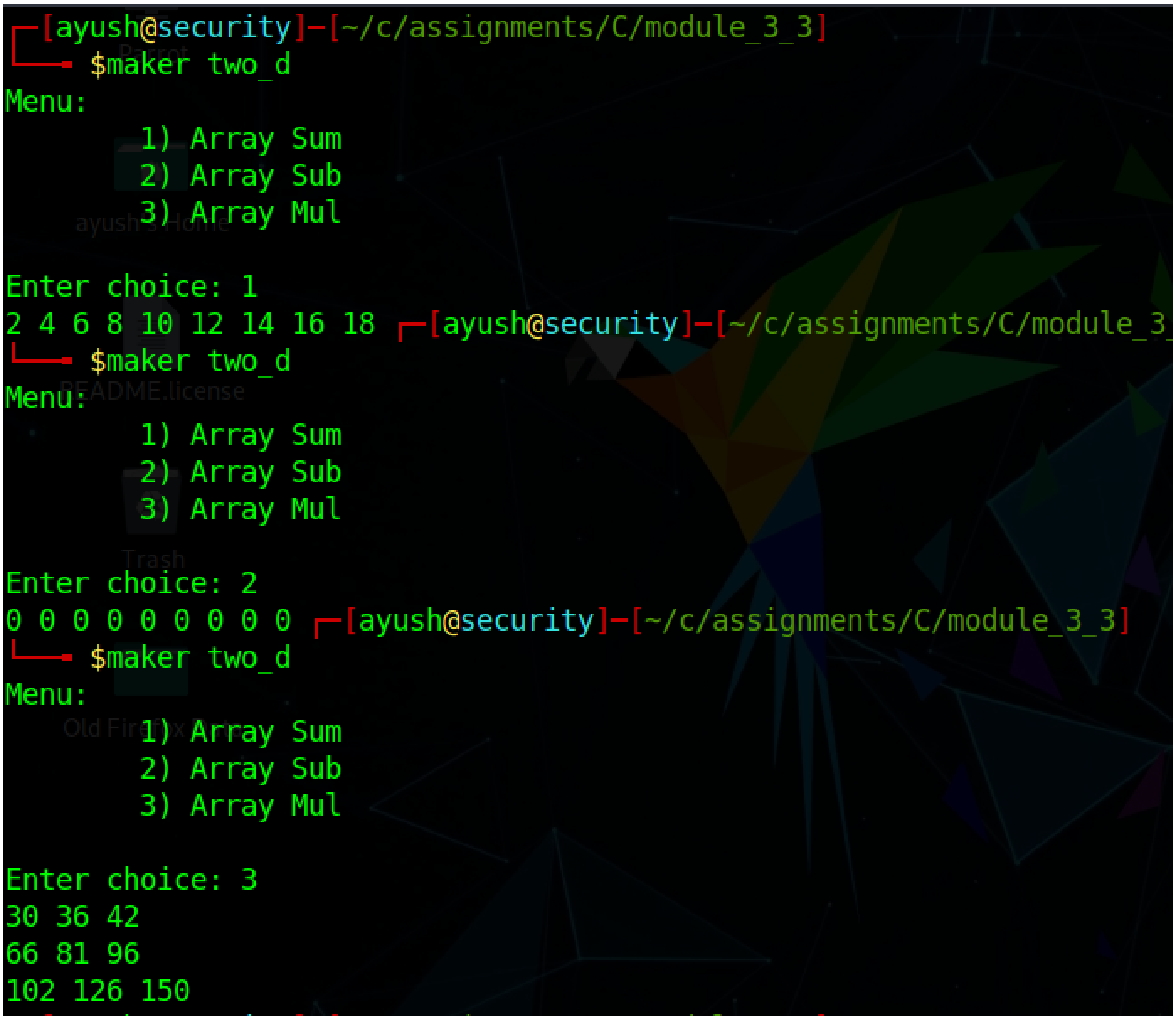


```c
    printf("Menu: \n");
    printf("\t1) Array Sum\n");
    printf("\t2) Array Sub\n");
    printf("\t3) Array Mul\n\n");
    printf("Enter choice: ");
    scanf("%d",&choice);

    if(choice == 1){
        arraySum(arr1,arr2);
    }
    else if(choice == 2){
        arraySub(arr1,arr2);
    }

    else if(choice == 3){
        arrayMul(arr1,arr2);
    }

    else {
        printf("Invalid choice\n");
    }

}


// Defining actual functions

void arraySum(int arr1[ROWS][COLS], int arr2[ROWS][COLS]){
    int result[3][3];
    for(int i = 0;i<3;i++){
        for(int j = 0;j<3;j++){
            result[i][j] = arr1[i][j] + arr2[i][j];
        }
    }

    for(int i = 0;i<3;i++){
        for(int j = 0;j<3;j++){
            printf("%d ",result[i][j]);
        }
    }
```



```
}
void arraySub(int arr1[ROWS][COLS], int arr2[ROWS][COLS]){
    int result[3][3];
    for(int i = 0;i<3;i++){
        for(int j = 0;j<3;j++){
            result[i][j] = arr1[i][j] - arr2[i][j];
        }
    }

    for(int i = 0;i<3;i++){
        for(int j = 0;j<3;j++){
            printf("%d ",result[i][j]);
        }
    }
}



void arrayMul(int arr1[ROWS][COLS], int arr2[ROWS][COLS]) {
    int result[3][3] = {0};

    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLS; j++) {
            for (int k = 0; k < COLS; k++) {
                result[i][j] += arr1[i][k] * arr2[k][j];
            }
        }
    }

    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLS; j++) {
            printf("%d ", result[i][j]);
        }
        printf("\n");
    }
}
```

Output:



```
┌─[ayush@security]─[~/c/assignments/C/module_3_3]
└──╼ $maker two_d
Menu:
        1) Array Sum
        2) Array Sub
        3) Array Mul

Enter choice: 1
2 4 6 8 10 12 14 16 18 ┌─[ayush@security]─[~/c/assignments/C/module_3_
└──╼ $maker two_d
Menu:
        1) Array Sum
        2) Array Sub
        3) Array Mul

Enter choice: 2
0 0 0 0 0 0 0 0 0 ┌─[ayush@security]─[~/c/assignments/C/module_3_3]
└──╼ $maker two_d
Menu:
        1) Array Sum
        2) Array Sub
        3) Array Mul

Enter choice: 3
30 36 42
66 81 96
102 126 150
```

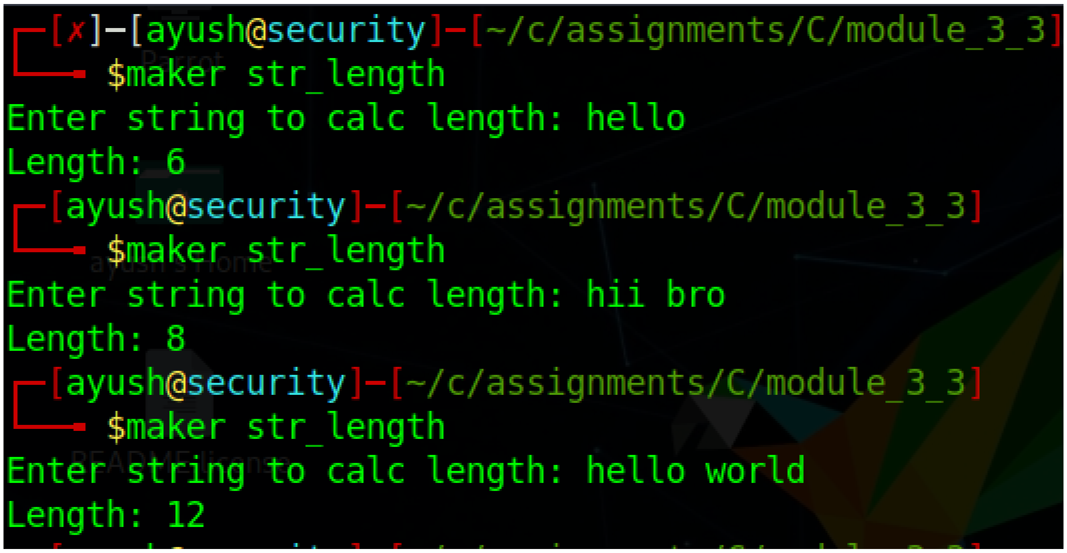
7. WAP Find out length of string without using inbuilt function

Code:

```
#include <stdio.h>

int main(void){
   // Taking user input
   char str[100];
   int length=0;

   printf("Enter string to calc length: ");
   fgets(str,100,stdin);

```

```
    // loop until null byte is found
    while(str[length] != '\0'){
        length++;
    }

    // Print length
    printf("Length: %d\n",length);
}
```

Output:

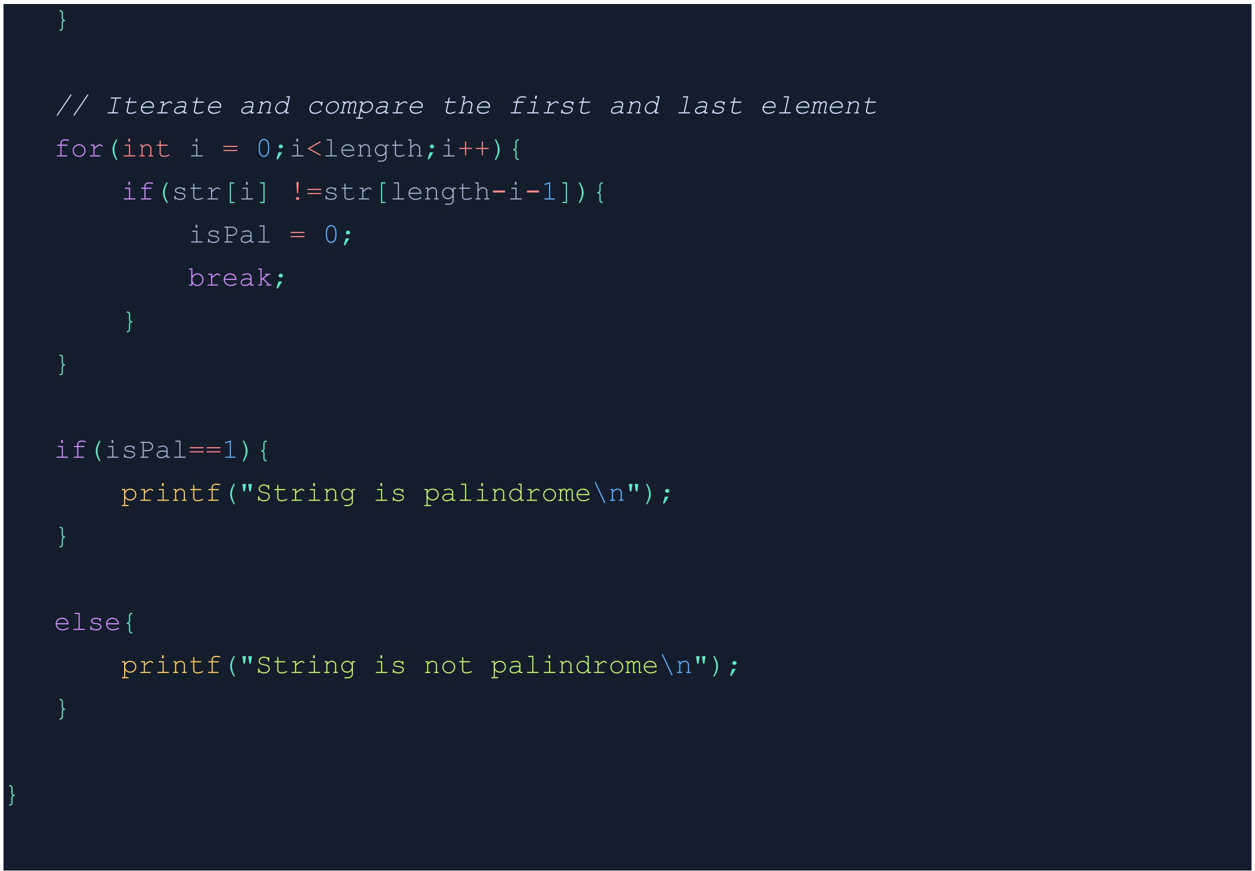


```
┌─[✗]─[ayush@security]─[~/c/assignments/C/module_3_3]
└──╼ $maker str_length
Enter string to calc length: hello
Length: 6
┌─[ayush@security]─[~/c/assignments/C/module_3_3]
└──╼ $maker str_length
Enter string to calc length: hii bro
Length: 8
┌─[ayush@security]─[~/c/assignments/C/module_3_3]
└──╼ $maker str_length
Enter string to calc length: hello world
Length: 12
```

8. WAP to reverse a string and check that the string is palindrome or not

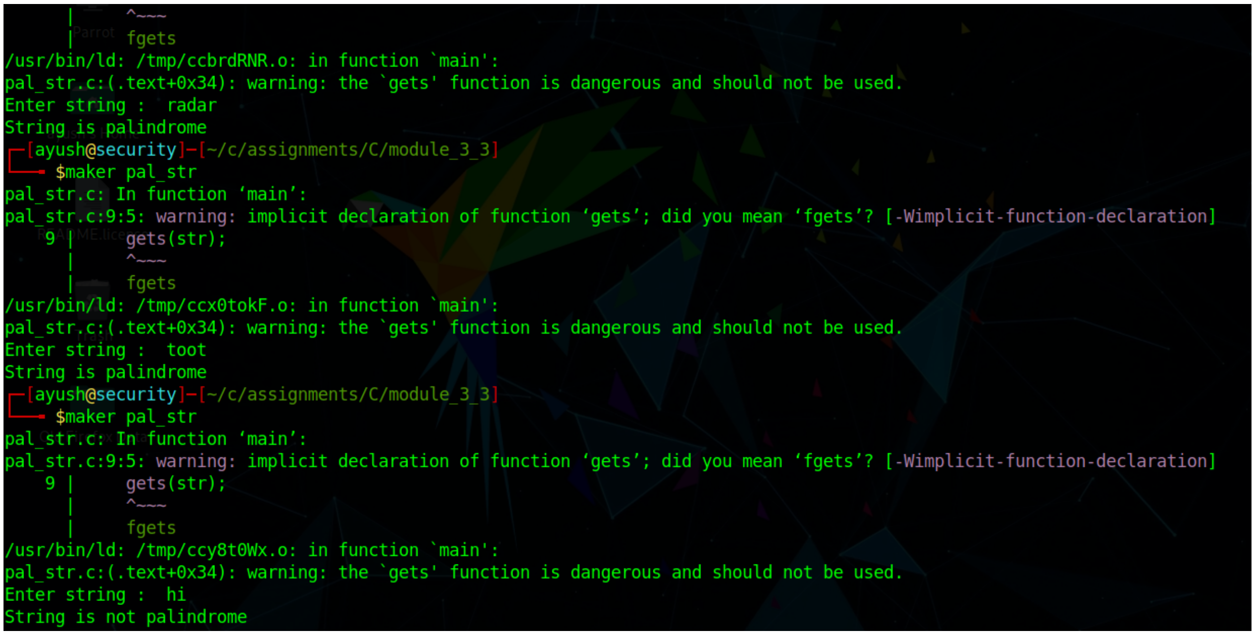
Code:

```
#include <stdio.h>

int main(void){
    // Taking user input
    char str[100];
    int length=0,isPal=1;

    printf("Enter string :  ");
    gets(str);

    // loop until null byte is found
    while(str[length] != '\0'){
        length++;
```



```c
    }

    // Iterate and compare the first and last element
    for(int i = 0;i<length;i++){
        if(str[i] !=str[length-i-1]){
            isPal = 0;
            break;
        }
    }

    if(isPal==1){
        printf("String is palindrome\n");
    }

    else{
        printf("String is not palindrome\n");
    }

}
```

Output:



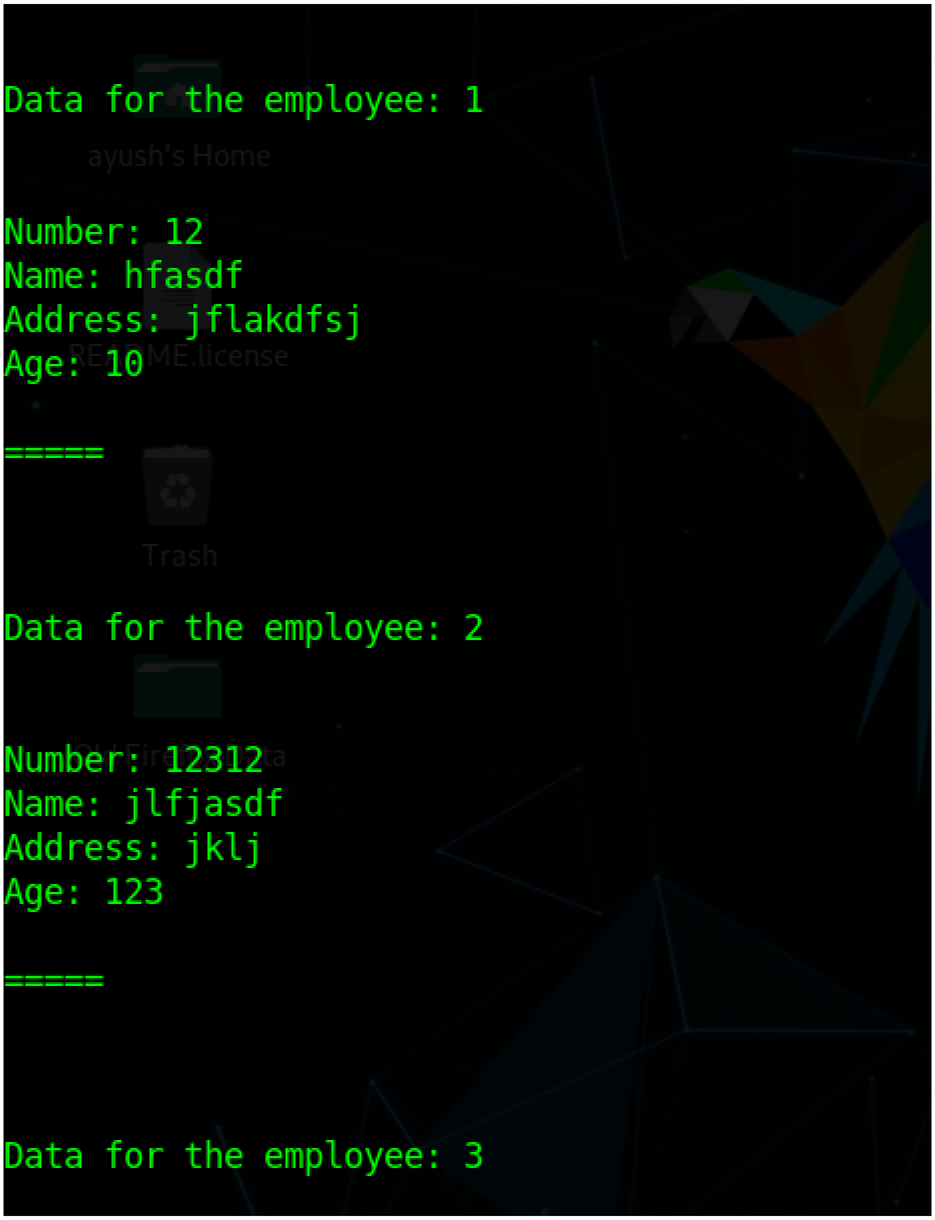
```
     |      ^~~~
     |      fgets
/usr/bin/ld: /tmp/ccbrdRNR.o: in function `main':
pal_str.c:(.text+0x34): warning: the `gets' function is dangerous and should not be used.
Enter string :  radar
String is palindrome
┌─[ayush@security]─[~/c/assignments/C/module_3_3]
└──╼ $maker pal_str
pal_str.c: In function ‘main’:
pal_str.c:9:5: warning: implicit declaration of function ‘gets’; did you mean ‘fgets’? [-Wimplicit-function-declaration]
    9 |     gets(str);
      |     ^~~~
      |     fgets
/usr/bin/ld: /tmp/ccx0tokF.o: in function `main':
pal_str.c:(.text+0x34): warning: the `gets' function is dangerous and should not be used.
Enter string :  toot
String is palindrome
┌─[ayush@security]─[~/c/assignments/C/module_3_3]
└──╼ $maker pal_str
pal_str.c: In function ‘main’:
pal_str.c:9:5: warning: implicit declaration of function ‘gets’; did you mean ‘fgets’? [-Wimplicit-function-declaration]
    9 |     gets(str);
      |     ^~~~
      |     fgets
/usr/bin/ld: /tmp/ccy8t0Wx.o: in function `main':
pal_str.c:(.text+0x34): warning: the `gets' function is dangerous and should not be used.
Enter string :  hi
String is not palindrome
```

9. Write a program of structure employee that provides the following information -print and display empno, empname, address and age

Code:

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

// Structure of employess
typedef struct Employees {
    int empno;
    char empName[20];
    char address[100];
    int age;
} Employees;

int main(void) {
    Employees emps[5];

    // Take data for 5 employees
    for (int i = 0; i < 5; i++) {
        printf("\n\n\nData for the employee: %d\n\n\n", i + 1);

        printf("Enter employee Number : ");
        scanf("%d", &emps[i].empno);
        getchar(); // Consume the newline character

        printf("Enter employee Name: ");
        fgets(emps[i].empName, 20, stdin);

        printf("Enter address: ");
        fgets(emps[i].address, 100, stdin);


        printf("Enter age : ");
        scanf("%d", &emps[i].age);
    }

    // Print the stored data of employees
    for (int i = 0; i < 5; i++) {
        printf("\n\n\nData for the employee: %d\n\n\n", i + 1);
        printf("Number: %d\n", emps[i].empno);
        printf("Name: %s", emps[i].empName);
        printf("Address: %s", emps[i].address);
        printf("Age: %d\n", emps[i].age);

        printf("\n=====\n");
    }

    return 0;
}
```

Output:



```
Data for the employee: 1


Number: 12
Name: hfasdf
Address: jflakdfsj
Age: 10

=====



Data for the employee: 2


Number: 12312
Name: jlfjasdf
Address: jklj
Age: 123

=====



Data for the employee: 3
```

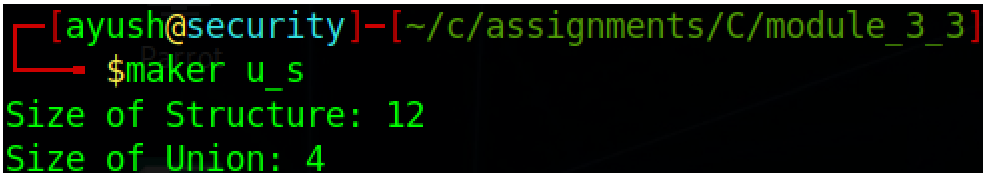
10. WAP to show difference between Structure and Union.

Code:

```
#include <stdio.h>

// The size of the structure is the size of all data in it
struct ExampleStruct {
   int x;
   char y;
   float z;
};
```

```c
// The size of the union is the size of the largest data type in it.
union ExampleUnion {
   int x;
   char y;
   float z;
};

int main() {
   // Creating a variable to create union and struct
   struct ExampleStruct myStruct;
   union ExampleUnion myUnion;
   // Printing the size of structure and union
   printf("Size of Structure: %lu\n", sizeof(myStruct));
   printf("Size of Union: %lu\n", sizeof(myUnion));

   return 0;
}
```

Output:



```
┌─[ayush@security]─[~/c/assignments/C/module_3_3]
└──╼ $maker u_s
Size of Structure: 12
Size of Union: 4
```