

Assignment C

Module 3.1 --> C Language Fundamental

1> Display this information using printf.

1.Your name, 2. Your Birth Date, 3. Your Age, 4. Your Address.

```
File Edit Selection View Go Run ...  
C Info.c  
Info.c > main()  
1 #include<stdio.h>  
2 int main()  
3 {  
4     printf("My name is Priyanka.");  
5     printf("\nMy birth date is 13/06/1997.");  
6     printf("\nMy age is 26.");  
7     printf("\nMy address is Balaji city.");  
8     return 0;  
9 }  
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> cd "c:\Users\nayna\Desktop\Programming\classes\Assignment C"\> if ($?) { gcc Info.c -o Info } ; if ($?) { .\Info }  
My name is Priyanka.  
My birth date is 13/06/1997.  
My age is 26.  
My address is Balaji city.  
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C>  
Ln 9, Col 2 Spaces: 4 UTF-8 CRLF ⌂ C ⌂ Go Live Win32 ⌂  
Ans.
```

2> Write a program to make Simple calculator (to make addition, subtraction, multiplication, division and modulo).

```
File Edit Selection View Go Run Terminal Help  
Assignment.c > main()  
Assignment.c > int main()  
Assignment.c > {  
Assignment.c >     int number1 , number2 ;  
Assignment.c >     char ch;  
Assignment.c >     printf("nEnter the operator from '+,-,/,*' :");  
Assignment.c >     scanf("%c",&ch);  
Assignment.c >     printf("nEnter two numbers : ");  
Assignment.c >     scanf("%d %d",&number1,&number2);  
Assignment.c >     switch (ch)  
Assignment.c >     {  
Assignment.c >         case '+':printf("\n %d + %d = %d", number1, number2, number1 + number2);  
Assignment.c >         break;  
Assignment.c >         case '-':printf("\n %d - %d = %d", number1, number2, number1 - number2);  
Assignment.c >         break;  
Assignment.c >         case '*':printf("\n %d * %d = %d", number1, number2, number1 * number2);  
Assignment.c >         break;  
Assignment.c >         case '/':printf("\n %d / %d = %d", number1, number2, number1 / number2);  
Assignment.c >         break;  
Assignment.c >         case '%':printf("\n %d %d = %d", number1, number2, number1 % number2);  
Assignment.c >         break;  
Assignment.c >         default:printf("Not an appropriate operator.");  
Assignment.c >         break;  
Assignment.c >     }  
Assignment.c >     return 0;  
Assignment.c > PS C:\Users\nayna\Desktop\classes\Assignment C> cd "c:\Users\nayna\Desktop\classes\Assignment C"\> if ($?) { gcc Assignment.c -o Assignment } ; if ($?) { .\Assignment }  
Enter the operator from '+,-,/,*' :  
Enter two numbers : 3 3  
3 + 3 = 6  
PS C:\Users\nayna\Desktop\classes\Assignment C> PS C:\Users\nayna\Desktop\classes\Assignment C> cd "c:\Users\nayna\Desktop\classes\Assignment C"\> if ($?) { gcc Assignment.c -o Assignment } ; if ($?) { .\Assignment }  
Enter the operator from '+,-,/,*' :  
Enter two numbers : 13 6  
13 - 6 = 7  
PS C:\Users\nayna\Desktop\classes\Assignment C> PS C:\Users\nayna\Desktop\classes\Assignment C> cd "c:\Users\nayna\Desktop\classes\Assignment C"\> if ($?) { gcc Assignment.c -o Assignment } ; if ($?) { .\Assignment }  
Enter the operator from '+,-,/,*' :  
Enter two numbers : 13 6  
13 * 6 = 78  
PS C:\Users\nayna\Desktop\classes\Assignment C> PS C:\Users\nayna\Desktop\classes\Assignment C> cd "c:\Users\nayna\Desktop\classes\Assignment C"\> if ($?) { gcc Assignment.c -o Assignment } ; if ($?) { .\Assignment }  
Enter the operator from '+,-,/,*' :  
Enter two numbers : 13 6  
13 / 6 = 2  
PS C:\Users\nayna\Desktop\classes\Assignment C> PS C:\Users\nayna\Desktop\classes\Assignment C> cd "c:\Users\nayna\Desktop\classes\Assignment C"\> if ($?) { gcc Assignment.c -o Assignment } ; if ($?) { .\Assignment }  
Enter the operator from '+,-,/,*' :  
Enter two numbers : 13 6  
13 % 6 = 1  
PS C:\Users\nayna\Desktop\classes\Assignment C> PS C:\Users\nayna\Desktop\classes\Assignment C> cd "c:\Users\nayna\Desktop\classes\Assignment C"\> if ($?) { gcc Assignment.c -o Assignment } ; if ($?) { .\Assignment }  
Enter the operator from '+,-,/,*' :  
Enter two numbers : 13 6  
Ans.
```

3> Write a program to find Area of circle, rectangle, and triangle.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files: Area-of-circle.c, Area-of-circle.exe, Area-of-circle.c, area-of-rectangle.c, Area-of-triangle.c, Area-of-triangle.exe, Assignment.c.
- Code Editor:** Displays the code for `Area-of-circle.c`.

```
C Area-of-circle.c
C Area-of-circle.c > main()
1 #include<stdio.h>
2 //Area of a circle
3 int main()
4 {
5     float radius;
6     printf("Area of a circle:");
7     scanf("%f",&radius);
8     printf("Area is : %f",3.14*radius*radius);
9     return 0;
10 }
```
- Terminal:** Shows the output of running the program.

```
PS C:\Users\nayna\Desktop\classes\Assignment C
> cd "c:\Users\nayna\Desktop\classes\Assignment C\"; if ($?) { gcc Area-of-circle.c -o Area-of-circle } ; if ($?) { ./Area-of-circle }
Area of a radius:4
Area is : 50.240000
PS C:\Users\nayna\Desktop\classes\Assignment C
>
```
- Bottom Status Bar:** Shows the file path as `Area-of-circle.c - Assignment C - Visual Studio Code`, line 10, column 2, and the date/time as 7/26/2023 5:13 PM.

Ans.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files: Area-of-triangle.c, Area-of-triangle.exe, Area-of-triangle.c, area-of-rectangle.c, Area-of-triangle.c, Area-of-triangle.exe, Assignment.c.
- Code Editor:** Displays the code for `Area-of-triangle.c`.

```
C Area-of-triangle.c
C Area-of-triangle.c > main()
1 #include<stdio.h>
2 int main()
3 {
4     float l,b,area;
5     printf("Enter the l : ");
6     scanf("%f",&l);
7     printf("Enter the b : ");
8     scanf("%f",&b);
9     area = 0.5*l*b;
10    printf("%f is area of a triangle",area);
11    return 0;
12 }
```
- Code Editor:** Displays the code for `area-of-rectangle.c`.

```
C area-of-rectangle.c
C area-of-rectangle.c > main()
1 #include<stdio.h>
2 int main()
3 {
4     float l,b,area;
5     printf("Enter the l : ");
6     scanf("%f",&l);
7     printf("Enter the b : ");
8     scanf("%f",&b);
9     area = (l*b);
10    printf("Area of a Rectengle : %f",area);
11    return 0;
12 }
```
- Terminal:** Shows the output of running the programs for triangle and rectangle area calculations.
- Bottom Status Bar:** Shows the file paths as `area-of-rectangle.c - Assignment C - Visual Studio Code` and `area-of-rectangle.c - Assignment C - Visual Studio Code`, line 10, column 33, and the date/time as 7/26/2023 5:21 PM and 5:23 PM respectively.

4> Write a program to find simple interest.

The screenshot shows a Visual Studio Code interface with the following details:

- Title Bar:** Simple-interest.c - Assignment C - Visual Studio Code
- Left Sidebar:** Shows icons for file operations like Open, Save, and Find.
- Code Editor:** Displays the following C code:

```
Simple-interest.c
1 #include<stdio.h>
2
3 int main(){
4
5     int P,R,T,SimpleInterest;
6
7     printf("Enter p : ");
8     scanf("%d",&P);
9
10    printf("Enter r : ");
11    scanf("%d",&R);
12
13    printf("Enter t : ");
14    scanf("%d",&T);
15
16    SimpleInterest = P * R * T * 100;
17    printf("Final Simple Interest is : %d",SimpleInterest);
18
19    return 0;
20 }
```
- Terminal:** Shows the command-line output of the program execution.

```
PS C:\Users\nayna\Desktop\classes\Assignment C> cd "c:\Users\nayna\Desktop\classes\Assignment C"
PS C:\Users\nayna\Desktop\classes\Assignment C> if ($?) {
  gcc Simple-interest.c -o Simple-interest } ; if ($?) {
  ./Simple-interest }
Enter p : 3
Enter r : 4
Enter t : 5
Final Simple Interest is : 6000
PS C:\Users\nayna\Desktop\classes\Assignment C>
```
- Bottom Status Bar:** Includes file navigation, workspace status (0△0), system icons (Windows, Task View, Start, File Explorer, Edge, Spotify, Taskbar), and system information (34°C Cloudy, 5:37 PM, 7/26/2023).

Ans.

5> Write a program to check if the given year is a leap year or not.

The screenshot shows a Visual Studio Code interface with the following details:

- Title Bar:** leap-year.c - Assignment C - Visual Studio Code
- Left Sidebar:** Shows icons for file operations like Open, Save, and Find.
- Code Editor:** Displays the following C code:

```
leap-year.c
1 #include <stdio.h>
2 int main()
3 {
4     int Year;
5     printf ("Enter a year : ");
6     scanf ("%d", &Year);
7
8     if ( Year % 4 == 0 && Year % 100 == 0 && Year % 400 == 0 )
9     {
10         printf("It is LEAP YEAR.");
11     }
12
13     else if (Year % 4 == 0 && Year % 100 != 0)
14     {
15         printf("It is LEAP YEAR.");
16     }
17
18     else
19     {
20         printf ("It is NOT LEAP YEAR.");
21     }
22
23     return 0;
24 }
```
- Terminal:** Shows the command-line output of the program execution.

```
PS C:\Users\nayna\Desktop\classes\Assignment C> cd "c:\Users\nayna\Desktop\classes\Assignment C"
PS C:\Users\nayna\Desktop\classes\Assignment C> if ($?) {
  gcc leap-year.c -o leap-year } ; if ($?) {
  ./leap-year }
Enter a year : 2020
It is LEAP YEAR.
PS C:\Users\nayna\Desktop\classes\Assignment C> cd "c:\Users\nayna\Desktop\classes\Assignment C"
PS C:\Users\nayna\Desktop\classes\Assignment C> if ($?) {
  gcc leap-year.c -o leap-year } ; if ($?) {
  ./leap-year }
Enter a year : 2014
It is NOT LEAP YEAR.
PS C:\Users\nayna\Desktop\classes\Assignment C>
```
- Bottom Status Bar:** Includes file navigation, workspace status (0△0), system icons (Windows, Task View, Start, File Explorer, Edge, Spotify, Taskbar), and system information (34°C Cloudy, 5:47 PM, 7/26/2023).

Ans.

6> Write a program to convert years into days and days into years.

The screenshot shows the Visual Studio Code interface. The left pane displays the C code for 'days-into-years.c'. The right pane shows the terminal output:

```
PS C:\Users\nayna\Desktop\classes\Assignment C> cd "c:\Users\nayna\Desktop\classes\Assignment C\" ; if ($?) { gcc days-into-years.c -o days-into-years } ; if ($?) { ./days-into-years }
Enter days: 1360
YEARS: 3
PS C:\Users\nayna\Desktop\classes\Assignment C> ^B
```

The code itself is as follows:

```
1 #include <stdio.h>
2 int main()
3 {
4     int Days, Years;
5     //Ask user to input number of days
6     printf("Enter days: ");
7     scanf("%d", &Days);
8     // Ignoring leap year
9     Years = (Days / 365);
10    //Print the result
11    printf("YEARS: %d\n", Years);
12    return 0;
13 }
```

Ans.

The screenshot shows the Visual Studio Code interface. The left pane displays the C code for 'years-into-days.c'. The right pane shows the terminal output:

```
PS C:\Users\nayna\Desktop\classes\Assignment C> cd "c:\Users\nayna\Desktop\classes\Assignment C\" ; if ($?) { gcc years-into-days.c -o years-into-days } ; if ($?) { ./years-into-days }
Enter year: 1997
DAYS: 728905
PS C:\Users\nayna\Desktop\classes\Assignment C> ^B
```

The code itself is as follows:

```
1 #include <stdio.h>
2 int main()
3 {
4     int Days, Year;
5     //Ask user to input number of years
6     printf("Enter year: ");
7     scanf("%d", &Year);
8     // Ignoring leap year
9     Days = (Year * 365);
10    //Print the result
11    printf("DAYS: %d\n", Days);
12    return 0;
13 }
```

Module 3.2 --> C Language Programming with C

1> Write a program to swap two numbers without using Third variable.

The screenshot shows a Visual Studio Code interface. On the left, there are two tabs: 'years-into-days.c' and 'Without-swapping.c'. The 'Without-swapping.c' tab is active, displaying the following code:

```
#include<stdio.h>
int main()
{
    //Without swapping
    int P = 13 , U = 30;
    printf("Before swap P is %d and U is %d", P , U );
    P = P + U;//P = 13 + 30 = 43
    U = P - U;//U = 43 - 30 = 13
    P = P - U;//P = 43 - 13 = 30
    printf("\nAfter swap P is %d and U is %d", P , U );
    return 0;
}
```

The terminal window on the right shows the output of the program:

```
PS C:\Users\nayna\Desktop\classes\Assignment C>
cd "c:\Users\nayna\Desktop\classes\Assignment C"
; if ($?) { gcc Without-swapping.c -o Without-swapping } ; if ($?) { ./Without-swapping }
Before swap P is 13 and U is 30
After swap P is 30 and U is 13
PS C:\Users\nayna\Desktop\classes\Assignment C>
```

Ans.

2> Write a program to find number is even or odd using ternary operator.

The screenshot shows a Visual Studio Code interface. On the left, there is one tab: 'ternary-operator.c'. The code is:

```
#include<stdio.h>
int main(){
    int number;

    number >= 35? printf("Odd \n"):printf("Even \n");

    return 0;
}
```

The terminal window on the right shows the output of the program:

```
PS C:\Users\nayna\Desktop\classes\Assignment C>
cd "c:\Users\nayna\Desktop\classes\Assignment C"
; if ($?) { gcc ternary-operator.c -o ternary-operator } ; if ($?) { ./ternary-operator }
Odd
PS C:\Users\nayna\Desktop\classes\Assignment C>
```

Ans.

3> Write a program to show

1. Monday to Sunday using switch case.

2. Vowel or Consonant using switch case.

The screenshot shows the Visual Studio Code interface. On the left, there are two tabs: 'ternary-operator.c' and 'm-s-switch-case.c'. The right pane displays the code for 'm-s-switch-case.c'. The code defines a function main() that prints the day of the week based on user input. It uses a switch statement with fall-through cases for each day from Monday to Sunday. The terminal window at the bottom shows the output of running the program, which correctly prints the days of the week.

```
#include<stdio.h>
int main()
{
    /*Switch statement*/
    char day;//m-mon t-tues w-wed T-turs s-sat S-sun;
    printf("Enter day(m-S) : ");
    scanf("%s",day);
    switch (day)
    {
        case 'm': printf("\nMonday");
        break;
        case 't': printf("\nTuesday");
        break;
        case 'w': printf("\nWednesday");
        break;
        case 'T': printf("\nThursday");
        break;
        case 's': printf("\nFriday");
        break;
        case 's': printf("\nSaturday");
        break;
        case 'S': printf("\nSunday");
        break;
        default: printf("\nNot a valid day!");
    }
}
```

Ans. (1) Ans.

The screenshot shows the Visual Studio Code interface. On the left, there is one tab: 'v-cons-switch-case.c'. The right pane displays the code for this program. It prompts the user to enter a character and then checks if it is a vowel or a consonant using a switch statement. The terminal window shows the output of running the program, which correctly identifies the input character as either a vowel or a consonant.

```
#include<stdio.h>
int main()
{
    /*Switch statement*/
    char ch;
    printf("Enter character : ");
    scanf("%c",ch);

    //condition to check character is alphabet or not
    if ( (ch >='A' && ch <='Z') || (ch >='a' && ch <='z') )
    {

        //check for VOWEL or CONSONANT
        switch (ch)
        {
            case 'A': printf("\nCh is a Vowel.",ch);
            break;
            case 'E': printf("\nCh is a Vowel.",ch);
            break;
            case 'I': printf("\nCh is a Vowel.",ch);
            break;
            case 'O': printf("\nCh is a Vowel.",ch);
            break;
            case 'U': printf("\nCh is a Vowel.",ch);
            break;
            case 'a': printf("\nCh is a Vowel.",ch);
            break;
            case 'e': printf("\nCh is a Vowel.",ch);
            break;
            case 'i': printf("\nCh is a Vowel.",ch);
            break;
            case 'o': printf("\nCh is a Vowel.",ch);
            break;
            case 'u': printf("\nCh is a Vowel.",ch);
            break;
            default: printf("\nCh is a consonant.",ch);
        }
    }
    else
    {
        printf("\nCh is not an alphabet.",ch);
    }
}

return 0;
```

(2) Ans.

4> Looping programs:

1. Write a program to print 972 to 897 using for loop.
2. Write a program to take 10 no. Input from user and find out

How many Even numbers are there?

How many odd numbers are there?

Sum of even numbers.

Sum of odd numbers.

A screenshot of Visual Studio Code showing a C program named 'for-loop.c'. The code uses a for loop to print integers from 972 down to 897. The terminal output shows the printed values. The system tray indicates it's 7:28 PM on 7/26/2023, the weather is 32°C mostly cloudy, and the battery level is at 0%.

```
C for-loop.c
C for-loop.c > cd main()
1 #include<stdio.h>
2
3 int main(){
4
5     // For loop
6
7     for ( int i = 972 ; i >= 897 ; i = i - 1 )
8     {
9         printf("%d\n", i);
10    }
11
12 } return 0;
```

for-loop.c - Assignment C - Visual Studio Code

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Users\Nayna\Desktop\Classes\Assignment C> cd "c:\Users\Nayna\Desktop\Classes\Assignment C"
for-loop.c -> for-loop.c
972
971
970
969
968
967
966
965
964
963
962
961
960
959
958
957
956
955
954
953
952
951
950
949
948
947
946
945
944
943
942
941
940
939
938
937
936
935
934
933
932
931

ln 9, Col 21 Spaces: 4 UTRF-B CRLF () C Go Use Win32 7:28 PM 7/26/2023

Ans. (1) Ans.

A second screenshot of Visual Studio Code showing the same C program 'for-loop.c'. The terminal output shows the printed values from 972 to 897. The system tray indicates it's 7:26 PM on 7/26/2023, the weather is 32°C mostly cloudy, and the battery level is at 0%.

```
C for-loop.c
C for-loop.c > cd main()
1 #include<stdio.h>
2
3 int main(){
4
5     // For loop
6
7     for ( int i = 972 ; i >= 897 ; i = i - 1 )
8     {
9         printf("%d\n", i);
10    }
11
12 } return 0;
```

for-loop.c - Assignment C - Visual Studio Code

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Users\Nayna\Desktop\Classes\Assignment C> cd "c:\Users\Nayna\Desktop\Classes\Assignment C"
for-loop.c -> for-loop.c
972
971
970
969
968
967
966
965
964
963
962
961
960
959
958
957
956
955
954
953
952
951
950
949
948
947
946
945
944
943
942
941
940
939
938
937
936
935
934
933
932
931

ln 9, Col 21 Spaces: 4 UTRF-B CRLF () C Go Use Win32 7:26 PM 7/26/2023

The screenshot shows the Visual Studio Code interface with the file "User-input.c" open. The code prompts the user to enter a number and then calculates the sum of even and odd numbers from 1 to 10. The terminal window on the right shows the execution of the program and its output.

```
C User-input.c X
C User-input.c > main()
1 #include <stdio.h>
2 int main(){
3     int number, i, sum = 0;
4     // Take number from user
5     printf("Enter number :");
6     scanf("%d", &i);
7     // Find Even numbers and its sum
8     for (number = 1; number <= 10; number++){
9         if (number % 2 == 0){
10             sum = sum + number;
11             printf("\nEven numbers is : %d", number);
12         }
13     }
14     printf("\nSum is : %d", sum);
15     // Find Odd numbers and its sum
16     for (number = 1; number <= 10; number++){
17         if (number % 2 != 0){
18             sum = sum + number;
19             printf("\nOdd numbers is : %d", number);
20         }
21     }
22     printf("\nSum is : %d", sum);
23     return 0;
24 }
```

```
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> cd "c:\Users\nayna\Desktop\Programming\classes\Assignment C\" ; if ($?) { gcc user-input.c -o user-input } ; if ($?) { ./user-input }
Enter number :10

Even numbers is : 2
Even numbers is : 4
Even numbers is : 6
Even numbers is : 8
Even numbers is : 10
Sum is : 30
Odd numbers is : 1
Odd numbers is : 3
Odd numbers is : 5
Odd numbers is : 7
Odd numbers is : 9
Sum is : 55
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C>
```

(2) Ans.

5> Write a program to print the table of the given number.

The screenshot shows the Visual Studio Code interface with the file "multiplication-table.c" open. The code prompts the user to enter a number and then prints its multiplication table from 1 to 10. The terminal window on the right shows the execution of the program and its output.

```
C multiplication-table.c X
C multiplication-table.c > main()
1 #include <stdio.h>
2
3 int main()
4 {
5     int number, p;
6
7     printf("Enter number : ");
8     scanf("%d", &number);
9
10    for (p = 1; p <= 10; p++)
11    {
12        printf("\nYour output is %d = %d x %d", number, p, number * p);
13    }
14
15    return 0;
16 }
```

```
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> cd "c:\Users\nayna\Desktop\Programming\classes\Assignment C\" ; if ($?) { gcc multiplication-table.c -o multiplication-table } ; if ($?) { ./multiplication-table }
Enter number : 6

Your output is 6 = 1 x 6
Your output is 6 = 2 x 12
Your output is 6 = 3 x 18
Your output is 6 = 4 x 24
Your output is 6 = 5 x 30
Your output is 6 = 6 x 36
Your output is 6 = 7 x 42
Your output is 6 = 8 x 48
Your output is 6 = 9 x 54
Your output is 6 = 10 x 60
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C>
```

Ans.

6> Write a program to print factorial of given number.

The screenshot shows the Visual Studio Code interface. On the left, the code editor displays a C program named factorial.c. The code defines a main function that includes stdio.h, prompts for a number, calculates its factorial using a for loop, and prints the result. On the right, the terminal window shows the command to compile the file, the input of '3', and the output 'Final factorial is : 6'.

```
C:\Users\nayna\Desktop\Programming\classes\Assignment C> cd "c:\Users\nayna\Desktop\Programming\classes\Assignment C\" ; if ($?) { gcc factorial.c -o factorial } ; if ($?) { ./factorial }
Enter number : 3
Final factorial is : 6
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C>
```

Ans.

7> Write a program to print Fibonacci series up to given numbers.

The screenshot shows the Visual Studio Code interface. On the left, the code editor displays a C program named fibonacci.c. The code defines a main function that includes stdio.h, prompts for a number, and prints the Fibonacci series up to that number using a for loop and conditional statements. On the right, the terminal window shows the command to compile the file, the input of '28', and the output of the first 28 numbers in the Fibonacci sequence.

```
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> cd "c:\Users\nayna\Desktop\Programming\classes\Assignment C\" ; if ($?) { gcc fibonacci.c -o fibonacci } ; if ($?) { ./fibonacci }
Enter number : 28
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C>
```

Ans.

8> Write a program to print numbers in reverse order.

The screenshot shows a Visual Studio Code interface. On the left, there are two tabs: "user-input.c" and "reverse-number.c". The "reverse-number.c" tab is active, displaying the following code:

```
#include<stdio.h>
int main(){
    int number;
    printf("Enter number : ");
    scanf("%d",&number);

    for(int j=10 ; j>=0 ; j--){
        printf("\n%d",j);
    }
    return 0;
}
```

To the right of the code editor is a terminal window showing the output of the program. The command run was:

```
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> cd "c:\Users\nayna\Desktop\Programming\classes\Assignment C\" ; if ($?) { gcc reverse-number.c -o reverse-number } ; if ($?) { ./reverse-number }
```

The terminal then prompts for input:

```
Enter number : 11
```

And displays the reversed output:

```
10
9
8
7
6
5
4
3
2
1
0
```

The status bar at the bottom shows the file path as "C:\Users\nayna\Desktop\Programming\classes\Assignment C", the line and column as "Ln 6, Col 17", and the date and time as "8/1/2023 6:37 PM".

Ans.

9> Write a program to find out the max from the given number.

The screenshot shows a Visual Studio Code interface. On the left, there is one tab: "maximum-number.c". The code is as follows:

```
#include<stdio.h>
int main()
{
    int n1,n2,n3;
    printf("Enter three number : ");
    scanf("%d %d %d",&n1,&n2,&n3);

    if(n1>n2 && n1>n3)
    {
        printf("n1 is maximum.");
    }
    else if(n2>n3 && n2>n1)
    {
        printf("n2 is maximum.");
    }
    else
    {
        printf("n3 is maximum.");
    }

    return 0;
}
```

To the right of the code editor is a terminal window showing the output of the program. The command run was:

```
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> cd "c:\Users\nayna\Desktop\Programming\classes\Assignment C\" ; if ($?) { gcc maximum-number.c -o maximum-number } ; if ($?) { ./maximum-number }
```

The terminal then prompts for input:

```
Enter three number : 3 4 5
```

And displays the result:

```
n3 is maximum.
```

The status bar at the bottom shows the file path as "C:\Users\nayna\Desktop\Programming\classes\Assignment C", the line and column as "Ln 18, Col 30", and the date and time as "8/2/2023 7:18 PM".

Ans.

10> Write a program and make a summation of the given number.

The screenshot shows the Visual Studio Code interface. On the left, there are two tabs: 'maximum-number.c' and 'summation.c'. The 'summation.c' tab is active, displaying the following C code:

```
C summation.c > main()
1 #include<stdio.h>
2 int main()
3 {
4     int number1 , number2 , sum = 0;
5
6     printf("Enter two numbers : ");
7     scanf("%d %d",&number1,&number2);
8
9     for (int i = number1; i <= number2; i++)
10    {
11        sum = sum + i;
12    }
13
14    printf("%d",sum);
15
16    return 0;
17 }
```

To the right of the code editor is a terminal window showing the execution of the program. The command entered was 'cd "c:\Users\nayna\Desktop\Programming\classes\Assignment C"'; if (\$?) { gcc summation.c -o summation }; if (\$?) { ./summation }. The user then entered 'Enter two numbers : 6 13' and the program output was '76'.

Ans.

11> Write a program you must make a summation of first and last Digit.

The screenshot shows the Visual Studio Code interface. On the left, there are three tabs: 'Function.c', 'first-last-digit.c', and 'Loops.c'. The 'first-last-digit.c' tab is active, displaying the following C code:

```
C first-last-digit.c > main()
1 #include<stdio.h>
2
3 int main(){
4
5     int first,last, Summation;
6
7     printf("Enter the first and last digit : ");
8     scanf("%d %d",&first,&last);
9
10    Summation = first + last ;
11
12    printf("%d", Summation);
13
14    return 0;
15 }
```

To the right of the code editor is a terminal window showing the execution of the program. The command entered was 'cd "c:\Users\nayna\Desktop\Programming\classes\C Tutorials"'; if (\$?) { gcc first-last-digit.c -o first-last-digit }; if (\$?) { ./first-last-digit }. The user then entered 'Enter the first and last digit : 13 6' and the program output was '19'.

Ans.

12>Patterns Programs:

(1)

```
*  
* * *  
* * * * *  
* * * * * * *  
* * * * * * * *
```

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** The file "Star-Patterns.c" is open. The code prints a triangle star pattern based on user input for the number of rows. The code uses nested loops to calculate the number of spaces and stars per row.
- Terminal:** The terminal shows the command to run the program and the output for 5 rows.
- Bottom Status Bar:** Lines 21, Column 13, Spaces: 4, UTF-8, CRLF, Go Live, Win32, 400 PM, 8/6/2023.
- Bottom Icons:** Weather (32°C, Cloudy), Taskbar icons (File Explorer, Task View, Taskbar, Spotify, Microsoft Edge, File Explorer, Taskbar, Taskbar, Taskbar).

```
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> cd "C:\Users\nayna\Desktop\Programming\classes\Assignment C\" ; if ($?) { gcc Star-Patterns.c -o Star-Patterns } ; if ($?) { .\Star-Patterns }

Enter the number of rows : 5
*
* * *
* * * * *
* * * * * *
* * * * * * *
```

Ans.

(2)

```
*  
* *  
* * *  
* * * *  
* * * * *  
* * * * * *  
* * * * *  
* * *  
* *  
*
```

The screenshot shows a Visual Studio Code interface with three tabs open: "Character-pattern.c", "Number-Pattern.c", and "Cross-Triangle-Star-Pattern.c". The "Cross-Triangle-Star-Pattern.c" tab is active, displaying the following code:

```
#include<stdio.h>  
int main(){  
    // Cross Triangle Star Pattern  
    int p, u = 1;  
    printf("Enter the number of columns : ");  
    scanf("%d", &p);  
    for (int i = 1; i <= p; i++)  
    {  
        for (int j = 1; j <= i; j++)  
        {  
            printf("* ");  
        }  
        printf("\n");  
    }  
    for (int i = p - 1; i >= 1; i--)  
    {  
        for (int j = 1; j <= i; j++)  
        {  
            printf("* ");  
        }  
        printf("\n");  
    }  
    return 0;  
}
```

The terminal window shows the output of the program:

```
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> cd "c:\Users\nayna\Desktop\Programming\classes\Assignment C\" ; if ($?) { gcc Cross-Triangle-Star-Pattern.c -o Cross-Triangle-Star-Pattern } ; if ($?) { .\Cross-Triangle-Star-Pattern }  
Enter the number of columns : 5  
*  
* *  
* * *  
* * * *  
* * * * *
```

The status bar at the bottom shows the file path "C:\Users\nayna\Desktop\Programming\classes\Assignment C\Cross-Triangle-Star-Pattern.c", line 10, column 21, and the date/time "8/6/2023 4:09 PM".

Ans.

(3) 1

2 3

4 5 6

7 8 9 10

11 12 13 14 15

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** The main editor window displays a C program named "Number-Pattern.c".

```
1 #include<stdio.h>
2
3 int main(){
4
5     // Number Pattern
6
7     int rows, i, j, number = 1;
8
9     printf("Enter the number of rows: ");
10    scanf("%d", &rows);
11
12    for (i = 1; i <= rows; i++)
13    {
14        for (j = 1; j <= i; ++j)
15        {
16            printf("%d ", number);
17            ++number;
18        }
19        printf("\n");
20    }
21
22    return 0;
23 }
```
- Terminal:** The terminal window shows the output of the program. It starts with the command "PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> cd "c:/Users/nayna\Desktop\Programming\classes\Assignment C\"; if (\$?) { gcc Number-Pattern.c -o Number-Pattern } ; if (\$?) { ./Number-Pattern }". Then it prompts "Enter the number of rows: 5" and lists the output:

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

PS C:\Users\nayna\Desktop\Programming\classes\Assignment C>
- Bottom Status Bar:** Shows file statistics (Ln 5, Col 14, Spaces: 4, CRLF), encoding (UTF-8), and other icons.
- Bottom Icons:** Includes icons for Cloud (32°C, Cloudy), Taskbar (Windows Start, Task View, File Explorer, Edge, Spotify, Taskbar Icons), and system status (Battery, Network, Volume).
- Bottom Right:** Shows the date and time (4:15 PM, 8/6/2023) and a blue circular icon with the number 13.

Ans.

(4) 1

1 0

1 0 1

1 0 1 0

1 0 1 0 1

The screenshot shows a Visual Studio Code interface with a dark theme. On the left is the code editor containing a C program named 'Numbers-pattern.c'. The code defines a 'pattern' function that prints a triangular number pattern. It includes a main function that calls the pattern function with the value 5. The terminal on the right shows the output of the program, which is:

```
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> cd "c:\Users\nayna\Desktop\Programming\classes\Assignment C\" ; if ($?) { gcc Numbers-pattern.c -o Numbers-pattern } ; if ($?) { ./Numbers-pattern }
```

1
1 0
1 0 1
1 0 1 0
1 0 1 0 1

The status bar at the bottom shows the file is 0 bytes, the current line is 7, column 10, and the encoding is UTF-8. The bottom right corner shows the date and time as 8/6/2023 8:03 PM.

Ans.

(5) A

A B

A B C

A B C D

A B C D E

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** The active tab is "Character-pattern.c - Assignment C - Visual Studio Code". The code is as follows:

```
#include<stdio.h>
int main(){
    // Character Pattern
    int n, x, y;
    printf("Enter number of rows to show character pattern: ");
    scanf("%d", &n);
    for (x = 1; x <= n; x++)
    {
        for (y = 1; y <= x; y++)
        {
            printf("%c ", 'A' + y - 1);
        }
        printf("\n");
    }
    return 0;
}
```

- Terminal:** The terminal shows the command line and the output of the program. It includes:
 - PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> cd "c:\Users\nayna\Desktop\Programming\classes\Assignment C"
 - if (\$?) { gcc Character-pattern.c -o Character-pattern ; if (\$?) { ./Character-pattern }
 - Enter number of rows to show character pattern: 5
 - A
 - A B
 - A B C
 - A B C D
 - A B C D E
 - PS C:\Users\nayna\Desktop\Programming\classes\Assignment C>
- Bottom Status Bar:** Ln 4, Col 5, Spaces: 4, UTF-8, CRLF, {} C, Go Live, Win32, 419 PM, 8/6/2023.
- Icons:** Weather (32°C Cloudy), Taskbar icons (File Explorer, Task View, Microsoft Edge, Spotify, VS Code, Taskbar icon).

Ans.

(6) A

B C

D E F

G H I J

K L M N O

The screenshot shows a Visual Studio Code window with the file "Characters-Pattern.c" open. The code prints a right-angled triangle of characters. The terminal on the right shows the command to run the program and its output.

```
#include <stdio.h>
int main()
{
    char alphabet = 'A';
    for(int i=1;i<=5;i++)
    {
        printf("\n ");
        for(int j=0;j<i;j++)
        {
            printf("%c ",alphabet);
            alphabet++;
        }
    }
    return 0;
}
```

Terminal Output:

```
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C>
cd "c:\Users\nayna\Desktop\Programming\classes\Assignment C\" ; if ($?) { gcc Characters-Pattern.c -o Characters-Pattern } ; if ($?) { .\Characters-Pattern }

A
B C
D E F
G H I J
K L M N O
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C>
```

Bottom status bar: Ln 13, Col 23 Spaces: 4 UTF-8 CRLF {} C Go Live Win32

Bottom icons: Cloudy 32°C, Taskbar icons (File Explorer, Mail, Spotify, VS Code, Task View), Date/Time: 10:26 PM, 8/6/2023.

Ans.

Module 3.3 --> File Handling and debugging

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

The screenshot shows a Visual Studio Code window with the file "maximum-number-of-array-withfunction.c" open. The code defines a function `max` that takes an array and its size as parameters and returns the maximum value. The main function prompts the user for the number of elements and the elements themselves, then iterates through the array to find and print the largest element. The terminal output shows the execution of the program and the user's input for six values, with the largest value being printed at the end.

```
File Edit Selection View Go ... maximum-number-of-array-withfunction.c - Assignment C - Visual Studio Code
C maximum-number-of-array-withfunction.c > max()
1 #include <stdio.h>
2 int max();
3 int main()
4 {
5     max();
6     return 0;
7 }
8
9 int max()
10 {
11     int number_of_elements;
12     printf("How many elements do you want to enter? \n");
13     scanf("%d", &number_of_elements);
14
15     int arr[number_of_elements];
16     for(int j = 0 ; j < number_of_elements ; j++)
17     {
18         int value;
19         printf("Enter %d value : ", j+1);
20         scanf("%d", &value);
21
22         arr[j] = value;
23     }
24
25     int max = arr[0];
26     for(int i = 1 ; i < number_of_elements ; i++)
27     {
28         if(arr[i] > max)
29         {
30             max = arr[i];
31         }
32     }
33     printf("\nLargest element is %d.",max);
34 }

PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> cd "c:\Users\nayna\Desktop\Programming\classes\Assignment C"
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> if ($?) { gcc maximum-number-of-array-withfunction.c -o maximum-number-of-array-withfunction }
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> ./maximum-number-of-array-withfunction
How many elements do you want to enter?
6
Enter 1 value : 13
Enter 2 value : 30
Enter 3 value : 10
Enter 4 value : 12
Enter 5 value : 5
Enter 6 value : 9
Largest element is 30.
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C>
```

Ans.

2> Write a program of Addition, Subtraction, Multiplication and Division using Switch case. (Must Be Menu Driven).

The screenshot shows a Visual Studio Code window with the file "menu-driven-switch-case.c" open. The program uses a switch statement to handle four operations based on user input: addition, subtraction, multiplication, and division. It prompts the user for two numbers and then performs the selected operation. The terminal output shows the execution of the program and the user's choice of operations like addition, subtraction, multiplication, and division.

```
File Edit Selection View Go Run ... menu-driven-switch-case.c - Assignment C - Visual Studio Code
C menu-driven-switch-case.c > main()
1 #include <stdio.h>
2
3 int main()
4 {
5     int a, b;
6     char choice;
7
8     printf("-----\n");
9     printf("1 for addition, Enter '-' for subtraction, Enter '*' for multiplication, Enter '/' for division.\n");
10    printf("Enter your choice : ");
11    scanf("%c", &choice);
12
13    printf("Enter two values : ");
14    scanf("%d %d", &a, &b);
15
16    switch(choice)
17    {
18        case '+': printf("%d + %d = %d", a, b, a + b); break;
19        case '-': printf("%d - %d = %d", a, b, a - b); break;
20        case '*': printf("%d * %d = %d", a, b, a * b); break;
21        case '/': printf("%d / %d = %d", a, b, a / b); break;
22        default: printf("You enter wrong choice."); break;
23    }
24
25    return 0;
26 }

PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> cd "c:\Users\nayna\Desktop\Programming\classes\Assignment C"
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> if ($?) { gcc menu-driven-switch-case.c -o menu-driven-switch-case }
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> ./menu-driven-switch-case
-----Man of the Program-----
Enter '+' for addition, Enter '-' for subtraction, Enter '*' for multiplication, Enter '/' for division.
Enter your choice : -
Enter two values : 13 6
13 - 6 = 7
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> cd "c:\Users\nayna\Desktop\Programming\classes\Assignment C"
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> if ($?) { gcc menu-driven-switch-case.c -o menu-driven-switch-case }
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> ./menu-driven-switch-case
-----Man of the Program-----
Enter '+' for addition, Enter '-' for subtraction, Enter '*' for multiplication, Enter '/' for division.
Enter your choice : *
Enter two values : 13 6
13 * 6 = 78
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> cd "c:\Users\nayna\Desktop\Programming\classes\Assignment C"
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> if ($?) { gcc menu-driven-switch-case.c -o menu-driven-switch-case }
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> ./menu-driven-switch-case
-----Man of the Program-----
Enter '+' for addition, Enter '-' for subtraction, Enter '*' for multiplication, Enter '/' for division.
Enter your choice : /
Enter two values : 13 6
13 / 6 = 2
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C>
```

Ans.

3> Write a program to find the reverse of string using recursion.

The screenshot shows a Visual Studio Code window with the file "reverse-string-with-recursion.c" open. The code defines a recursive function to reverse a string. It includes #include directives for stdio.h and string.h, and a main() function that prompts the user for a string, calls the reverse() function, and prints the result. The reverse() function takes a string, an index, and a size, and swaps characters at index and size - index. A condition checks if index is half the size. The terminal output shows the program being run in a Windows command prompt, entering "Priyanka" and getting the reversed output "aknayirP".

```
reverse-string-with-recursion.c
1 #include <stdio.h>
2 #include <string.h>
3 void reverse(char [], int index, int size);
4 int main(){
5     char str1[20];
6     int size;
7
8     printf("Enter a string to reverse: ");
9     scanf("%s", str1);
10    size = strlen(str1);
11    reverse(str1, 0, size - 1);
12    printf("The string after reversing is: %s\n", str1);
13    return 0;
14 }
15 void reverse(char str1[], int index, int size){
16     char temp;
17     temp = str1[index];
18     str1[index] = str1[size - index];
19     str1[size - index] = temp;
20     if (index == size / 2){
21         return;
22     }
23     reverse(str1, index + 1, size);
24 }
```

PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> cd "c:\Users\nayna\Desktop\Programming\classes\Assignment C\" ; if (\$?) { gcc reverse-string-with-recursion.c -o reverse-string-with-recursion } ; if (\$?) { ./reverse-string-with-recursion }
Enter a string to reverse: Priyanka
The string after reversing is: aknayirP
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C>

Ans.

4> Write a program to find factorial using recursion.

The screenshot shows a Visual Studio Code window with the file "factorial-with-recursion.c" open. The code defines a recursive function to calculate the factorial of a number. It includes #include<stdio.h>, and a main() function that prompts the user for a positive integer and prints its factorial. The factorial() function takes an integer n and returns the product of n and the factorial of n-1 until n is 1. The terminal output shows the program being run in a Windows command prompt, entering "6" and getting the output "Factorial of 6 = 720".

```
factorial-with-recursion.c
1 #include<stdio.h>
2 long int multiplyNumbers(int n);
3 int main() {
4     int n;
5     printf("Enter a positive integer: ");
6     scanf("%d",&n);
7     printf("Factorial of %d = %ld", n, multiplyNumbers(n));
8     return 0;
9 }
10 long int multiplyNumbers(int n) {
11     if (n>=1)
12         return n*multiplyNumbers(n-1);
13     else
14         return 1;
15 }
```

PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> cd "c:\Users\nayna\Desktop\Programming\classes\Assignment C\" ; if (\$?) { gcc factorial-with-recursion.c -o factorial-with-recursion } ; if (\$?) { ./factorial-with-recursion }
Enter a positive integer: 6
Factorial of 6 = 720
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C>

Ans.

5> Write a program to take two Array input from user and sort them in ascending or descending order as per user's choice.

Ans.

Acceding Order

The screenshot shows the Visual Studio Code interface with the file 'Ascending-array-shorting.c' open. The code implements a bubble sort algorithm to sort an array in ascending order. The terminal window shows the execution of the program, prompting for the number of elements (6), displaying the initial array [13, 30, 10, 12, 6, 9], and then showing the sorted array [6, 9, 10, 12, 13, 30].

```
#include <stdio.h>
int main()
{
    int p,n,number[100],temp;
    printf("How many elements you have to enter? ");
    scanf("%d",&n);
    printf("Enter elements : ");
    for(p=0;p<n;p++)
    {
        scanf("%d",&number[p]);
    }
    printf("Before sorting elements are : \n");
    for(p=0;p<n;p++)
    {
        printf("%d\t",number[p]);
    }
    for(p=0;p<n;p++)
    {
        for(u=p+1;u<n;u++)
        {
            if(number[p]>number[u])
            {
                temp=number[p];
                number[p]=number[u];
                number[u]=temp;
            }
        }
    }
    printf("\nAfter sorting elements are: \n");
    for(p=0;p<n;p++)
    {
        printf("%d\t",number[p]);
    }
    return 0;
}
```

```
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> cd "c:\Users\nayna\Desktop\Programming\classes\Assignment C"
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> if ($?) { gcc Ascending-array-shorting.c -o Ascending-array-shorting } ; if ($?) { ./Ascending-array-shorting }
How many elements you have to enter? 6
Enter elements : 13 30 10 12 6 9
Before sorting elements are :
13 30 10 12 6 9
After sorting elements are:
6 9 10 12 13 30
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C>
```

Descending Order

The screenshot shows the Visual Studio Code interface with two files open: 'decending-array-shorting.c' and 'Ascending-array-shorting.c'. The code implements a bubble sort algorithm to sort an array in descending order. The terminal window shows the execution of the program, prompting for the number of elements (6), displaying the initial array [13, 12, 10, 9, 5], and then showing the sorted array [13, 12, 12, 10, 9, 5].

```
#include <stdio.h>
void main()
{
    int number[100],tmp,n,p,u;
    printf("Enter elements of array in descending order : \n");
    printf("-----\n");
    printf("How many elements you have to enter? ");
    scanf("%d", &n);
    printf("Before sorting elements are : \n");
    for(p=0;p<n;p++)
    {
        printf("element - %d : ",p);
        scanf("%d",&number[p]);
    }
    for(p=0; p<n; p++)
    {
        for(u=p+1; u<n; u++)
        {
            if(number[p] < number[u])
            {
                tmp = number[p];
                number[p] = number[u];
                number[u] = tmp;
            }
        }
    }
    printf("\nAfter sorting elements are : \n");
    for(p=0; p<n; p++)
    {
        printf("%d\t", number[p]);
    }
    printf("\n\n");
}
```

```
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> cd "c:\Users\nayna\Desktop\Programming\classes\Assignment C"
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> if ($?) { gcc decending-array-shorting.c -o decending-array-shorting } ; if ($?) { ./decending-array-shorting }
sort elements of array in descending order :
-----
How many elements you have to enter? 6
Before sorting elements are :
element - 0 : 13
element - 1 : 12
element - 2 : 12
element - 3 : 5
element - 4 : 9
element - 5 : 10
After sorting elements are :
13 12 12 10 9 5
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C>
```

6> Write a program to make Addition, Subtraction and Multiplication of two matrix using 2-D Array.

Ans.

Addition of two matrix using 2-D Array

```
PS C:\Users\nayna\Desktop\Programming\classes\C Language\C task> cd "c:\Users\nayna\Desktop\Programming\classes\C Language\C task"
PS C:\Users\nayna\Desktop\Programming\classes\C Language\C task> if ($?) { gcc addition-of-matrix.c -o addition-of-matrix }
Enter your first array : 12 30 10 5 9 6
12 30 10
5 9 6
Enter your second array : 13 4 97 75 76 12
13 4 97
75 76 12
The sum of two array :
25 34 107
80 85 18
PS C:\Users\nayna\Desktop\Programming\classes\C Language\C task>
```

addition-of-matrix.c

```
#include <stdio.h>
int main()
{
    int arr[2][3], arr[2][3], ans[2][3], p, u;
    printf("Enter your first array : ");
    for (p = 0; p < 2; p++)
        for (u = 0; u < 3; u++)
            scanf("%d", &arr[p][u]);
    printf("\n");
    printf("Enter your second array : ");
    for (p = 0; p < 2; p++)
        for (u = 0; u < 3; u++)
            scanf("%d", &arr[p][u]);
    printf("\n");
    for (p = 0; p < 2; p++)
        for (u = 0; u < 3; u++)
            ans[p][u] = arr[0][u] + arr[1][u];
    printf("\n");
    printf("The sum of two array : \n");
    for (p = 0; p < 2; p++)
        for (u = 0; u < 3; u++)
            printf("%d", ans[p][u]);
    printf("\n");
}
return 0;
```

Subtraction of two matrix using 2-D Array

```
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> cd "c:\Users\nayna\Desktop\Programming\classes\Assignment C"
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> if ($?) { gcc Subtraction-of-matrix.c -o Subtraction-of-matrix }
Enter your first array : 30 75 76 97 13 12
30 75 76
97 13 12
Enter your second array : 5 6 9 10 4 12
5 6 9
10 4 12
The sub of two array :
25 69 67
87 9 0
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C>
```

Subtraction-of-matrix.c

```
#include <stdio.h>
int main()
{
    int arr[2][3], arr[2][3], ans[2][3], p, u;
    printf("Enter your first array : ");
    for (p = 0; p < 2; p++)
        for (u = 0; u < 3; u++)
            scanf("%d", &arr[p][u]);
    printf("\n");
    printf("Enter your second array : ");
    for (p = 0; p < 2; p++)
        for (u = 0; u < 3; u++)
            scanf("%d", &arr[p][u]);
    printf("\n");
    for (p = 0; p < 2; p++)
        for (u = 0; u < 3; u++)
            ans[p][u] = arr[0][u] - arr[1][u];
    printf("\n");
    printf("The sub of two array : \n");
    for (p = 0; p < 2; p++)
        for (u = 0; u < 3; u++)
            printf("%d", ans[p][u]);
    printf("\n");
}
return 0;
```

Multiplication of two matrix using 2-D Array

The screenshot shows a Visual Studio Code window with the file "Multiplication-of-matrix.c" open. The code implements matrix multiplication using 2D arrays. It prompts the user for two 3x3 matrices and prints their product. The terminal output shows the execution of the program, inputting the matrices and displaying the resulting 3x3 matrix.

```
#include <stdio.h>
int main()
{
    int arr[2][3], arr[2][3], ans[2][3], p, u;
    printf("Enter your first array : ");
    for (p = 0; p < 2; p++)
    {
        for (u = 0; u < 3; u++)
        {
            scanf("%d", &arr[p][u]);
        }
    }
    printf("\n");
    for (p = 0; p < 2; p++)
    {
        for (u = 0; u < 3; u++)
        {
            printf("%d\t", arr[p][u]);
        }
        printf("\n");
    }
    printf("Enter your second array : ");
    for (p = 0; p < 2; p++)
    {
        for (u = 0; u < 3; u++)
        {
            scanf("%d", &arr[p][u]);
        }
    }
    printf("\n");
    for (p = 0; p < 2; p++)
    {
        for (u = 0; u < 3; u++)
        {
            printf("%d\t", arr[p][u]);
        }
        printf("\n");
    }
    for (p = 0; p < 2; p++)
    {
        for (u = 0; u < 3; u++)
        {
            ans[p][u] = arr[p][0] * arr[0][u];
        }
    }
    for (p = 0; p < 2; p++)
    {
        for (u = 0; u < 3; u++)
        {
            ans[p][u] = ans[p][u] + arr[p][1] * arr[1][u];
        }
    }
    for (p = 0; p < 2; p++)
    {
        for (u = 0; u < 3; u++)
        {
            ans[p][u] = ans[p][u] + arr[p][2] * arr[2][u];
        }
    }
    printf("The mul of two array : \n");
    for (p = 0; p < 2; p++)
    {
        for (u = 0; u < 3; u++)
        {
            printf("%d\t", ans[p][u]);
        }
        printf("\n");
    }
}
return 0;
```

PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> cd "c:\Users\nayna\Desktop\Programming\classes\Assignment C" ; if (\$?) { gcc Multiplication-of-matrix.c -o Multiplication-of-matrix } ; if (\$?) { ./Multiplication-of-matrix }

Enter your first array : 30 10 13 6 5
6 5 9

Enter your second array : 12 12 75 76 4 97
76 4 97

The mul of two array :
360 120 975
456 20 873

PS C:\Users\nayna\Desktop\Programming\classes\Assignment C>

7> Write a program to Find out length of string without using inbuilt function.

The screenshot shows a Visual Studio Code window with the file "length-of-string-without-using-inbuilt-function.c" open. The code defines a function to calculate the length of a character array (string) by iterating through it until it finds the null terminator '\0'. The terminal output shows the execution of the program, which correctly prints the length of the string "Priyanka".

```
#include<stdio.h>
int main()
{
    char name[]="Priyanka";
    int p,length;

    for(p=0; name[p] != '\0';p++)
    {
        length++;
    }

    printf("%d", length);

    return 0;
}
```

PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> cd "c:\Users\nayna\Desktop\Programming\classes\Assignment C" ; if (\$?) { gcc length-of-string-without-using-inbuilt-function.c -o length-of-string-without-using-inbuilt-function } ; if (\$?) { ./length-of-string-without-using-inbuilt-function }

PS C:\Users\nayna\Desktop\Programming\classes\Assignment C>

Ans.

8> Write a program to reverse a string and check if the string is palindrome or not.

The screenshot shows a Visual Studio Code window with a dark theme. On the left is the code editor containing a C program named `string-palindrome.c`. The code defines a function `main` that prints the input string `Priyanka` reversed. It then initializes a character array `name` with the value `"isi"`. A loop checks if the string is a palindrome by comparing characters from both ends. The terminal on the right shows the command line and the output of running the program, which is `isi is a palindrome`.

```
C string-palindrome.c > main()
1 #include <stdio.h>
2 #include <string.h>
3
4 int main()
5 {
6     char name[] = "Priyanka";
7     printf("%s\n", strrev(name));
8
9     char name[] = { "isi" };
10    int l = 0;
11    int h = strlen(name) - 1;
12
13    while (h > l) {
14        if (name[l++] != name[h--]) {
15            printf("%s is not a palindrome\n", name);
16            return 0;
17        }
18    }
19    printf("%s is a palindrome\n", name);
20    return 0;
21
22
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> cd "C:\Users\nayna\Desktop\Programming\classes\Assignment C"
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> gcc string-palindrome.c -o string-palindrome
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> ./string-palindrome
isi is a palindrome
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C>
```

Ans.

9> Write a program of employee structure that provides the following information - Print and Display Employee-number, Employee-name, Address and Age.

The screenshot shows a Visual Studio Code window with a dark theme. On the left is the code editor containing a C program named `structure-employee.c`. It defines a structure `employee` with fields `emp_name`, `emp_address`, `emp_age`, and `emp_number`. In the `main` function, it creates an instance of the structure `s1` with values `Priyanka`, `Bhuj`, `29`, and `1345678902`. It then prints these values. The terminal on the right shows the command line and the output of running the program, which displays the employee's name, address, age, and number.

```
C structure-employee.c > main()
1 #include <stdio.h>
2
3 /*structure declaration*/
4 struct employee
5 {
6     char emp_name[30], emp_address[30];
7     int emp_age, emp_number;
8 };
9
10 int main()
11 {
12     struct employee s1 = {"Priyanka", "Bhuj", 29, 1345678902};
13     printf("\nEmployee name : %s \nEmployee address : %s \nEmployee age : %d \nEmployee number : %d", s1.emp_name, s1.emp_address, s1.emp_age, s1.emp_number);
14     return 0;
15 }
```

```
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> cd "C:\Users\nayna\Desktop\Programming\classes\Assignment C"
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C> gcc structure-employee.c -o structure-employee
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C>
Employee name : Priyanka
Employee address : Bhuj
Employee age : 29
Employee number : 1345678902
PS C:\Users\nayna\Desktop\Programming\classes\Assignment C>
```

Ans.

10> Write a program of structure for five employees that provides the following information -print and display Employee number, Employee name, Address and Age.

```

File Edit Selection View Go Run Terminal Help
Structure-Employees.c - Assignment C - Visual Studio Code
C:\Users\aynaya\Desktop\Programming\classes\Assignment C>
1 #include<stdio.h>
2 #include<string.h>
3
4 struct employee{
5     char name[100];
6     char address[100];
7     int age;
8     int phonenumber;
9 };
10
11 int main(){
12     struct employee emp[100];
13     int n;
14     printf("How many employees details you want to enter?\n");
15     scanf("%d", &n);
16     for(int p=0;p<n;p++)
17     {
18         printf("\nEnter details for Employee %d : ", p+1);
19         scanf("%s", &emp[p].name);
20         printf("Enter employee Address : ");
21         scanf("%s", &emp[p].address);
22         printf("Enter employee Age : ");
23         scanf("%d", &emp[p].age);
24         printf("Enter employee Phone Number : ");
25         scanf("%d", &emp[p].phonenumber);
26     }
27     printf("\n-----Details of Employees-----");
28     for(int p=0;p<n;p++)
29     {
30         printf("\nEmployee Name is %s Employee Address is %s Employee Age is %d Employee Phone number is %d",emp[p].name,emp[p].address,emp[p].age,emp[p].phonenumber);
31     }
32 }
33 return 0;

```

The screenshot shows the code in the editor and the terminal output. The terminal shows the program running, prompting for the number of employees (5), then entering details for each employee (Name, Address, Age, Phone Number) and finally printing all the entered details.

Ans.

11> Write a program to show the difference between Structure and Union.

```

File Edit Selection View Go Run Terminal Help
difference-between-structure-union.c - Assignment C - Visual Studio Code
C:\Users\aynaya\Desktop\Programming\classes\Assignment C>
1 #include<stdio.h>
2 #include<string.h>
3
4 struct Student{
5     int rollno;
6     char name[100];
7     float pr;
8 };
9
10 union Student1{
11     int rollno;
12     char name[100];
13     float pr;
14 };
15
16 int main(){
17     //difference 1: For structure struct keyword is used
18     //           For union union keyword is used
19     struct Student s1;
20     s1.rollno = 13;
21     strcpy(s1.name,"Priyanka");
22     s1.pr = 89;
23     //difference 2: You can access multiple data of structure in single line
24     //           You can't access multiple data of union in different line
25     printf("Structure data :\n");
26     printf("s1.rollno is Rollno \ns1.name is Name \ns1.pr is Pr",s1.rollno,s1.name,s1.pr);
27     union Student1 u1;
28     u1.rollno = 13;
29     printf("Union data :\n");
30     strcpy(u1.name,"Priyanka");
31     printf("u1.name is Name",u1.name);
32     //difference 3: For union you can't update same data
33     //           For structure you can update same data
34     u1.pr = 99;
35     printf("u1.pr is Pr",u1.pr);
36     //difference 4:Declare the size of union
37     //           Declare the size of structure
38     printf("\nStructure Size: %d",sizeof(s1));
39     printf("\nUnion Size: %d",sizeof(u1));
40 }
41

```

The screenshot shows the code in the editor and the terminal output. The terminal output illustrates the differences between structures and unions, such as the ability to access multiple members of a structure in one line and the inability to do so with a union, and the fact that a union's size is determined by its largest member while a structure's size is the sum of its members' sizes.

Ans.