

Program 1

The screenshot displays the Visual Studio Code interface with a C program open in the editor. The Explorer sidebar on the left shows a project structure with files 1.c through 14.c, 1.exe through 14.exe, program8.c, program12.c, tempCodeRunnerFile.c, test.c, and test.exe. The editor window shows the code for 1.c, which includes a main function that prints the value of a variable 'a' and its right and left shifted outputs. The TERMINAL panel at the bottom shows the execution of the program, displaying the output: a: 9, right shifted output: 18, and left shifted output: 4.

```
1.c > main(void)
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int a;
6
7      printf("a: ");
8      scanf("%d", &a);
9
10     // printf("b: ");
11     // scanf("%d", &b);
12
13     printf("right shifted output: %d\n", a << 1);
14     printf("left shifted output: %d", a >> 1);
15
16     return 0;
17 }
```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

```
PS D:\VSCode\NSUT\CP Sem1\Projects> cd "d:\VSCode\NSUT\CP Sem1\Projects"
PS D:\VSCode\NSUT\CP Sem1\Projects> cd "d:\VSCode\NSUT\CP Sem1\Projects\" ; if ($?) { gcc 1.c -o 1 } ; if ($?) { .\1 }
a: 9
right shifted output: 18
left shifted output: 4
PS D:\VSCode\NSUT\CP Sem1\Projects>
```

Ln 17, Col 2 Spaces: 4 UTF-8 CRLF C Win32

Program 2

The screenshot shows the Visual Studio Code interface with a C program open in the editor. The Explorer sidebar on the left lists project files from 1.c to 12.c and their corresponding executables. The editor displays the code for 2.c, which includes headers, variable declarations, input handling, a conditional check, and output formatting. The TERMINAL panel at the bottom shows the command prompt execution, including directory navigation, compilation, and the program's output for inputs 4 and 3, resulting in a quotient of 1.3.

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int a, b;
6
7      printf("a: ");
8      scanf("%d", &a);
9
10     printf("b: ");
11     scanf("%d", &b);
12
13     if(b == 0)
14     {
15         return 1;
16     }
17
18     printf("quotient: %.1f\n", (float) a/b);
19
20     return 0;
21 }
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL**

```
PS D:\VSCode\NSUT\CP Sem1\Projects> cd "d:\VSCode\NSUT\CP Sem1\Projects"
PS D:\VSCode\NSUT\CP Sem1\Projects> cd "d:\VSCode\NSUT\CP Sem1\Projects\" ; if ($?) { gcc 2.c -o 2 } ; if ($?) { .\2 }
a: 4
b: 3
quotient: 1.3
PS D:\VSCode\NSUT\CP Sem1\Projects> 
```

Ln 13, Col 15 Spaces: 4 UTF-8 CRLF C Win32

Program 3

The screenshot displays the Visual Studio Code interface with a C program open in the editor. The Explorer sidebar on the left shows a project structure with files 1.c through 14.c, 11.exe through 14.c, program8.c, program12.c, tempCodeRunnerFile.c, test.c, and test.exe. The main editor window shows the code for 3.c, which implements a XOR function and a main function that takes user input and prints the result. The terminal at the bottom shows the execution of the program, with input values 5 and 10, and the output 15.

```
1  #include <stdio.h>
2
3  int XOR(int a, int b);
4
5  int main(void)
6  {
7      int a = 0, b = 0;
8      // INPUT (Do While loop can be removed if -ve inputs are acceptable)
9      do
10     {
11         printf("a: ");
12         scanf("%i", &a);
13     }
14     while(a < 0);
15
16     do
17     {
18         printf("b: ");
19         scanf("%i", &b);
20     }
21     while(a < 0);
22
23     // OUTPUT
24     printf("%d\n", XOR(a, b));
25     printf("%d\n", a^b); //TO VERIFY THE PROGRAM
26     return 0;
27 }
28
29 int XOR(int a, int b)
30 {
31     int result = 0;
32     result = (a | b) & ~(a & b);
33
34     return result;
35 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\VSCode\NSUT\CP Sem1\Projects> cd "d:\VSCode\NSUT\CP Sem1\Projects"
PS D:\VSCode\NSUT\CP Sem1\Projects> cd "d:\VSCode\NSUT\CP Sem1\Projects\" ; if ($?) { gcc 3.c -o 3 } ; if ($?) { .\3 }
a: 5
b: 10
15
15
PS D:\VSCode\NSUT\CP Sem1\Projects>
```

Ln 23, Col 14 Spaces: 4 UTF-8 CRLF C Win32

Program 4

The image shows a Visual Studio Code editor window with a C program for logical operations. The Explorer sidebar on the left lists files: 1.c, 1.exe, 2.c, 2.exe, 3.c, 3.exe, 4.c (selected), 4.exe, 5.c, 5.exe, 6.c, 6.exe, 7.c, 7.exe, 8.c, 9.c, 9.exe, 10.c, 11.c, 11.exe, 12.c, 12.exe, 13.c, 14.c, program8.c, program12.c, tempCodeRunnerFile.c, test.c, and test.exe. The main editor displays the code for 4.c, which implements AND, OR, XOR, and NOT operations. The TERMINAL panel at the bottom shows the program's execution output.

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

int main()
{
    int num1, num2;
    printf("Enter the operation you wanted to operate\n 1) AND \n 2) OR \n 3) XOR \n 4) NOT \n");
    char str[5];
    scanf("%s", &str);
    if (strcmp(str, "AND") == 0)
    {
        printf("Enter 1st number\n");
        scanf("%d", &num1);
        printf("Enter 2nd number\n");
        scanf("%d", &num2);
        printf("the AND operation on these two a&b is : %d", num1 & num2);
    }
    else if (strcmp(str, "OR") == 0)
    {
        printf("Enter 1st number\n");
        scanf("%d", &num1);
        printf("Enter 2nd number\n");
        scanf("%d", &num2);
        printf("the OR operation on these two a|b is : %d", num1 | num2);
    }
    else if (strcmp(str, "XOR") == 0)
    {
        printf("Enter 1st number\n");
        scanf("%d", &num1);
        printf("Enter 2nd number\n");
        scanf("%d", &num2);
    }
}
```

TERMINAL

```
Enter the operation you wanted to operate
1) AND
2) OR
3) XOR
4) NOT
AND
Enter 1st number
4
Enter 2nd number
5
the AND operation on these two a&b is : 4
PS D:\VSCode\NSUT\CP Sem1\Projects>
```

Ln 46, Col 1 Spaces: 4 UTF-8 CRLF C Win32

Program 4

The screenshot displays the Visual Studio Code interface with a C program open in the editor. The program is a menu-driven application for performing bitwise operations: AND, OR, XOR, and NOT. It prompts the user to enter the operation and two numbers, then displays the result.

EXPLORER

- PROJECTS
 - 1.c
 - 1.exe
 - 2.c
 - 2.exe
 - 3.c
 - 3.exe
 - 4.c
 - 4.exe
 - 5.c
 - 5.exe
 - 6.c
 - 6.exe
 - 7.c
 - 7.exe
 - 8.c
 - 9.c
 - 9.exe
 - 10.c
 - 11.c
 - 11.exe
 - 12.c
 - 12.exe
 - 13.c
 - 14.c
 - program8.c
 - program12.c
 - tempCodeRunnerFile.c
 - test.c
 - test.exe

4.c

```
19 {
20     printf("Enter 1st number\n");
21     scanf("%d", &num1);
22     printf("Enter 2nd number\n");
23     scanf("%d", &num2);
24     printf("the OR operation on these two a|b is : %d", num1 | num2);
25 }
26 else if (strcmp(str, "XOR") == 0)
27 {
28     printf("Enter 1st number\n");
29     scanf("%d", &num1);
30     printf("Enter 2nd number\n");
31     scanf("%d", &num2);
32     printf("the XOR operation on these two a^b is : %d", num1 ^ num2);
33 }
34 else if (strcmp(str, "NOT") == 0)
35 {
36     printf("Enter 1st number\n");
37     scanf("%d", &num1);
38     printf("Enter 2nd number\n");
39     scanf("%d", &num2);
40     printf("the NOT operation on these two ~a and ~b is : %d and %d", ~num1, ~num2);
41 }else{
42     printf("Wrong input\n");
43 }
44
45 return 0;
46
```

TERMINAL

```
Enter the operation you wanted to operate
1) AND
2) OR
3) XOR
4) NOT
AND
Enter 1st number
4
Enter 2nd number
5
the AND operation on these two a&b is : 4
PS D:\VSCode\NSUT\CP Sem1\Projects>
```

Ln 46, Col 1 Spaces: 4 UTF-8 CRLF C Win32

Program 5

The screenshot shows the Visual Studio Code interface with a C program open in the editor. The program is a 2D array traversal. The Explorer sidebar on the left shows a project structure with files 1.c through 14.c, program8.c, program12.c, tempCodeRunnerFile.c, test.c, and test.exe. The file 5.c is selected. The editor displays the code for 5.c, which includes a header file, a main function, and nested loops for array traversal. The output window at the bottom shows the execution results.

```
#include <stdio.h>

int main(void)
{
    int n[31], m, arr[3][31];

    printf("m: ");
    scanf("%d", &m);

    for(int i = 0; i < m; i++)
    {
        printf("n%d: ", i+1);
        scanf("%d", &arr[i][0]);
    }

    for(int i = 0; i < m; i++)
    {
        for(int j = 1; j <= arr[i][0]; j++)
        {
            scanf("%d", &arr[i][j]);
        }
    }

    for(int i = 0; i < m; i++)
    {
        for(int j = 1; j <= arr[i][0]; j++)
        {
            printf("%d", arr[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

OUTPUT

```
m: 2
n1: 1
n2: 5
1
2
4
5
5
3
12455
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF C Win32

Program 6

The screenshot displays the Visual Studio Code interface with a C program open in the editor. The Explorer sidebar on the left shows a project structure with files 1.c through 14.c, 1.exe through 14.exe, program8.c, program12.c, and test.c. The main editor window shows the code for 6.c, which is a C program that reads a string and prints its length. The terminal at the bottom shows the command to compile and run the program, resulting in the output 'hello world'.

6.c - Projects - Visual Studio Code

EXPLORER

PROJECTS

- 1.c
- 1.exe
- 2.c
- 2.exe
- 3.c
- 3.exe
- 4.c
- 4.exe
- 5.c
- 5.exe
- 6.c
- 6.exe
- 7.c
- 7.exe
- 8.c
- 9.c
- 9.exe
- 10.c
- 10.exe
- 11.c
- 11.exe
- 12.c
- 12.exe
- 13.c
- 14.c
- program8.c
- program12.c
- tempCodeRunnerFile.c
- test.c
- test.exe

6.c > main()

```
1 #include <stdio.h>
2
3 int main()
4 {
5     char *s;
6
7     gets(s);
8
9     int i = 0, counter = 0;
10    do
11    {
12        counter++;
13        i++;
14    } while (s[i] != '\0' || s[i] == ' ');
15
16    printf("%d", counter);
17
18 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\VSCode\NSUT\CP Sem1\Projects> cd "d:\VSCode\NSUT\CP Sem1\Projects"
PS D:\VSCode\NSUT\CP Sem1\Projects> cd "d:\VSCode\NSUT\CP Sem1\Projects\" ; if ($?) { gcc 6.c -o 6 } ; if ($?) { .\6 }
hello world
11
PS D:\VSCode\NSUT\CP Sem1\Projects>
```

OUTLINE

Ln 18, Col 2 Spaces: 4 UTF-8 CRLF C Win32

Program 7

The screenshot shows the Visual Studio Code interface with a C program open in the editor. The Explorer sidebar on the left lists various files and executables. The main editor window displays the source code for '7.c', which includes headers, variable declarations, and loops for input and calculation. The bottom panel shows the TERMINAL with the execution output.

EXPLORER

- PROJECTS
 - 1.c
 - 1.exe
 - 2.c
 - 2.exe
 - 3.c
 - 3.exe
 - 4.c
 - 4.exe
 - 5.c
 - 5.exe
 - 6.c
 - 6.exe
 - 7.c
 - 7.exe
 - 8.c
 - 9.c
 - 9.exe
 - 10.c
 - 11.c
 - 11.exe
 - 12.c
 - 12.exe
 - 13.c
 - 14.c
 - program8.c
 - program12.c
 - tempCodeRunnerFile.c
 - test.c
 - test.exe

7.c

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int n;
6      printf("n: ");
7      scanf("%d", &n);
8
9      int arr[n];
10
11
12     int age = 0, AGE = 0;
13     for(int i = 0; i < n; i++)
14     {
15         scanf("%d", &*(arr+i));
16     }
17
18     for(int i = 0; i < n; i++)
19     {
20         age += *(arr+i);
21     }
22
23     printf("Avg age of employees: %.2f", (float) age/n);
24
25     return 0;
26 }
```

TERMINAL

```
PS D:\VSCode\NSUT\CP Sem1\Projects> cd "d:\VSCode\NSUT\CP Sem1\Projects"
PS D:\VSCode\NSUT\CP Sem1\Projects> cd "d:\VSCode\NSUT\CP Sem1\Projects\" ; if ($?) { gcc 7.c -o 7 } ; if ($?) { .\7 }
n: 3
40
50
60
Avg age of employees: 50.00
PS D:\VSCode\NSUT\CP Sem1\Projects>
```

Ln 14, Col 6 Spaces: 4 UTF-8 CRLF C Win32

Program 8

The screenshot shows the Visual Studio Code interface with a C program for a queue implementation. The Explorer panel on the left shows a project structure with folders and files. The main editor displays the code for 8.c, which includes a queue implementation using an array. The code is as follows:

```
1  #include <stdio.h>
2
3  int queue[20];
4  int *front = queue;
5  int *back = queue;
6
7  int enqueue(int data)
8  {
9
10     if (*back == 20 - 1)
11     {
12         return (-1);
13     }
14
15     else
16     {
17         *back = data;
18         back++;
19         return (1);
20     }
21 }
22
23 void dequeue(int data)
24 {
25     if (back == queue)
26     {
27         printf("Queue is full \n");
28     }
29
30     else
31     {
32         printf("%d is deleted\n", *front);
33         for (int i = 0; i < back - front; i++)
34         {
35             *(front + i) = *(front + 1 + i);
36         }
37         back--;
38     }
39 }
40
41 int main()
42 {
43     int data;
44
45     int num, response;
46     printf("\n 1.Enqueue  ");
47     printf("\n 2.Dequeue  ");
48     printf("\n 3.Print queue");
49     while (1)
50     {
```

The status bar at the bottom indicates the current position is Line 29, Column 6, with 4 spaces, UTF-8 encoding, CRLF line endings, and a C compiler.

Program 8

The screenshot shows the Visual Studio Code interface with a C project open. The Explorer sidebar on the left lists files and folders, including 'New folder', 'Cultural.py', 'My_Interest.py', 'Sports.py', and several C source and executable files (1.c to 14.c, 1.exe to 14.exe). The main editor window displays the code for '8.c', which implements a queue using an array. The code includes functions for enqueueing and dequeuing elements, and a main function that provides a menu-driven interface for these operations. The status bar at the bottom indicates the current position is Line 29, Column 6, with 4 spaces, UTF-8 encoding, CRLF line endings, and a C compiler.

```
File Edit Selection View Go Run Terminal Help
• 8.c - Projects - Visual Studio Code

EXPLORER
PROJECTS
  New folder
  Cultural.py
  My_Interest.py
  Sports.py
  1.c
  1.exe
  2.c
  2.exe
  3.c
  3.exe
  4.c
  4.exe
  5.c
  5.exe
  6.c
  6.exe
  7.c
  7.exe
  8.c
  9.c
  9.exe
  10.c
  10.exe
  11.c
  11.exe
  12.c
  12.exe
  13.c
  13.exe
  14.c
  14.exe
  abc.txt
  output.txt
  OUTLINE

8.c > enqueue(int)
19 printf("\n 3.Print queue");
20 while (1)
21 {
22     printf("Choice: ");
23     scanf("%d", &num);
24     switch (num)
25     {
26     case 1: // insertion
27         printf("Enter Element: ");
28         scanf("%d", &data);
29
30         response = enqueue(data);
31
32         if (response == -1)
33         {
34         }
35         else
36         {
37             printf("%d is inserted\n", data);
38         }
39
40         break;
41     case 2: // deletion
42
43         dequeue(data);
44
45         break;
46     case 3: // display
47         if (front == back)
48         {
49             printf("Queue is empty \n");
50         }
51         else
52         {
53             printf("Queue=> \n");
54             for (int i = 0; i < back - front; i++)
55             {
56                 printf("%d.%d\n", i + 1, queue[i]);
57             }
58         }
59
60         break;
61     case 4:
62         return 0;
63     default:
64         printf("Invalid input");
65         break;
66     }
67 }
68 }
```

Ln 29, Col 6 Spaces: 4 UTF-8 CRLF C Win32

Program 8

The image shows a Visual Studio Code editor window with a C program for a queue. The Explorer sidebar on the left shows a project structure with files 1.c through 14.c and their corresponding .exe files, along with a folder named 'New folder' containing Cultural.py, My_Interest.py, and Sports.py. The main editor displays the code for 8.c, which implements a queue using an array. The terminal at the bottom shows the execution of the program, including the compilation command and the interactive input/output sequence.

EXPLORER

- PROJECTS
 - New folder
 - Cultural.py
 - My_Interest.py
 - Sports.py
 - 1.c
 - 1.exe
 - 2.c
 - 2.exe
 - 3.c
 - 3.exe
 - 4.c
 - 4.exe
 - 5.c
 - 5.exe
 - 6.c
 - 6.exe
 - 7.c
 - 7.exe
 - 8.c
 - 8.exe
 - 9.c
 - 9.exe
 - 10.c
 - 10.exe
 - 11.c
 - 11.exe
 - 12.c
 - 12.exe
 - 13.c
 - 13.exe
 - 14.c
 - 14.exe
 - abc.txt
- OUTLINE

8.c

```
1 deque(int)
2
3 int queue[20];
4 int *front = queue;
5 int *back = queue;
6
7 int enqueue(int data)
8 {
9     if (*back == 20 - 1)
10    {
11        return -1;
```

TERMINAL

```
PS D:\VSCode\NSUT\CP Sem1\Projects\New folder> cd "d:\VSCode\NSUT\CP Sem1\Projects"
SUT\CP Sem1\Projects> cd "d:\VSCode\NSUT\CP Sem1\Projects\" ; if ($?) { gcc 8.c -o 8 } ; if ($?) { .\8 }

1.Enqueue
2.Dequeue
3.Print queueChoice: 1
Enter Element: 5
5 is inserted
Choice: 1
Enter Element: 4
4 is inserted
Choice: 3
Queue=>
1.5
2.4
Choice: 2
5 is deleted
Choice: 3
Queue=>
1.4
Choice: █
```

Ln 32, Col 43 Spaces: 4 UTF-8 CRLF C Win32

Program 9

The screenshot shows the Visual Studio Code interface with a C program open in the editor. The Explorer sidebar on the left shows a project structure with files 1.c through 14.c, 1.exe through 14.exe, program8.c, program12.c, tempCodeRunnerFile.c, test.c, and test.exe. The editor displays the code for 9.c, which is a C program for entering and displaying employee details. The code includes a main function that prompts the user for the number of employees, then uses a loop to collect details for each employee, including name, age, degree, experience, street number, city, district, and state. The details are then printed for each employee.

```
9.c > main()
1  #include <stdio.h>
2
3  int main()
4  {
5      int a;
6      printf("How many employees are there in your company?\n");
7      scanf("%d", &a);
8      getchar();
9      typedef struct employeeDetails
10     {
11         char name[50];
12         int age;
13         char degree[30];
14         float exp;
15         struct address
16         {
17
18             char streetno[30];
19             char city[30];
20             char district[30];
21             char state[30];
22
23         };
24     } ED;
25
26     ED arr[a];
27     for (int i = 0; i < a; i++)
28     {
29         printf("Enter the name of %d employee\n", i + 1);
30         gets(arr[i].name);
31         printf("Enter the age of %d employee\n", i + 1);
32         scanf("%d", &arr[i].age);
33         getchar();
34         printf("Enter the degree of %d employee\n", i + 1);
35         gets(arr[i].degree);
36         printf("Enter the experience %d employee\n", i + 1);
37         scanf("%f", &arr[i].exp);
38         getchar();
39         printf("Enter the streetno. of %d employee\n", i + 1);
40         gets(arr[i].streetno);
41         printf("Enter the city of %d employee\n", i + 1);
42         gets(arr[i].city);
43         printf("Enter the district of %d employee\n", i + 1);
44         gets(arr[i].district);
45         printf("Enter the state of %d employee\n", i + 1);
46         gets(arr[i].state);
47         printf("\n");
```

Ln 19, Col 23 Spaces: 4 UTF-8 CRLF C Win32

Program 9

The image shows a Visual Studio Code editor with a C program in the main editor and its execution output in the terminal. The program is a C file named 9.c, and the terminal shows the command to compile and run it, followed by the user's input for two employees.

```
9.c - Projects - Visual Studio Code
```

```
EXPLORER
```

- 1.c
- 2.c
- 3.c
- 5.c
- 9.c
- 12.c
- program8.c
- 8.c
- 10.c
- 6.c

```
main()
43
44     printf("Enter the district of %d employee\n", i + 1);
45     gets(arr[i].district);
46     printf("Enter the state of %d employee\n", i + 1);
47     gets(arr[i].state);
48     printf("\n");
49 }
50 for (int i = 0; i < a; i++)
51 {
52     printf("the name of employee no.%d is : %s\n", i + 1, arr[i].name);
53     printf("the age of employee no.%d is : %d\n", i + 1, arr[i].age);
54     printf("the degree of employee no.%d is : %s\n", i + 1, arr[i].degree);
55     printf("the experience of employee no.%d is : %f\n", i + 1, arr[i].exp);
56     printf("the streetno. of employee no.%d is : %s\n", i + 1, arr[i].streetno);
57     printf("the city of employee no.%d is : %s\n", i + 1, arr[i].city);
58     printf("the district of employee no.%d is : %s\n", i + 1, arr[i].district);
59     printf("the state of employee no.%d is : %s\n", i + 1, arr[i].state);
60     printf("\n");
61 }
62 return 0;
```

```
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL
```

```
PS D:\VSCode\NSUT\CP Sem1\Projects> cd "d:\VSCode\NSUT\CP Sem1\Projects\" ; if ($?) { gcc 9.c -o 9 } ; if ($?) { .\9 }
How many employees are there in your company?
2
Enter the name of 1 employee
stg
Enter the age of 1 employee
25
Enter the degree of 1 employee
btech
Enter the experience 1 employee
5
Enter the streetno. of 1 employee
1212
Enter the city of 1 employee
delhi
Enter the district of 1 employee
delhi
Enter the state of 1 employee
delhi

Enter the name of 2 employee
njo
```

Ln 62, Col 2 Spaces: 4 UTF-8 CRLF C Win32

Program 9

The screenshot displays the Visual Studio Code interface with a C program open in the editor. The program is titled '9.c' and is located in the 'PROJECTS' folder. The code is as follows:

```
9.c > main()
43     printf("Enter the district of %d employee\n", i + 1);
44     gets(arr[i].district);
45     printf("Enter the state of %d employee\n", i + 1);
46     gets(arr[i].state);
47     printf("\n");
48 }
49 for (int i = 0; i < a; i++)
50 {
51     printf("the name of employee no.%d is : %s\n", i + 1, arr[i].name);
52     printf("the age of employee no.%d is : %d\n", i + 1, arr[i].age);
53     printf("the degree of employee no.%d is : %s\n", i + 1, arr[i].degree);
54     printf("the experience of employee no.%d is : %f\n", i + 1, arr[i].exp);
55     printf("the streetno. of employee no.%d is : %s\n", i + 1, arr[i].streetno);
56     printf("the city of employee no.%d is : %s\n", i + 1, arr[i].city);
57     printf("the district of employee no.%d is : %s\n", i + 1, arr[i].district);
58     printf("the state of employee no.%d is : %s\n", i + 1, arr[i].state);
59     printf("\n");
60 }
61 return 0;
62 }
```

The output window shows the execution results for two employees:

```
Enter the name of 2 employee
njo
Enter the age of 2 employee
55
Enter the degree of 2 employee
bsc
Enter the experience 2 employee
10
Enter the streetno. of 2 employee
212
Enter the city of 2 employee
patna
Enter the district of 2 employee
patna
Enter the state of 2 employee
bihar

the name of employee no.1 is : stg
the age of employee no.1 is : 25
the degree of employee no.1 is : btech
the experience of employee no.1 is : 5.000000
the streetno. of employee no.1 is : 1212
```

The status bar at the bottom indicates the current line is 62, column 2, with 4 spaces, UTF-8 encoding, CRLF line endings, and a Win32 architecture.

Program 9

The screenshot displays the Visual Studio Code interface with a C program open in the editor. The program is titled '9.c' and is located in the 'Projects' folder. The code is as follows:

```
9.c > main()
43     printf("Enter the district of %d employee\n", i + 1);
44     gets(arr[i].district);
45     printf("Enter the state of %d employee\n", i + 1);
46     gets(arr[i].state);
47     printf("\n");
48 }
49 for (int i = 0; i < a; i++)
50 {
51     printf("the name of employee no.%d is : %s\n", i + 1, arr[i].name);
52     printf("the age of employee no.%d is : %d\n", i + 1, arr[i].age);
53     printf("the degree of employee no.%d is : %s\n", i + 1, arr[i].degree);
54     printf("the experience of employee no.%d is : %f\n", i + 1, arr[i].exp);
55     printf("the streetno. of employee no.%d is : %s\n", i + 1, arr[i].streetno);
56     printf("the city of employee no.%d is : %s\n", i + 1, arr[i].city);
57     printf("the district of employee no.%d is : %s\n", i + 1, arr[i].district);
58     printf("the state of employee no.%d is : %s\n", i + 1, arr[i].state);
59     printf("\n");
60 }
61 return 0;
62 }
```

The output window shows the execution results for two employees:

```
bihar

the name of employee no.1 is : stg
the age of employee no.1 is : 25
the degree of employee no.1 is : btech
the experience of employee no.1 is : 5.000000
the streetno. of employee no.1 is : 1212
the city of employee no.1 is : delhi
the district of employee no.1 is : delhi
the state of employee no.1 is : delhi

the name of employee no.2 is : njo
the age of employee no.2 is : 55
the degree of employee no.2 is : bsc
the experience of employee no.2 is : 10.000000
the streetno. of employee no.2 is : 212
the city of employee no.2 is : patna
the district of employee no.2 is : patna
the state of employee no.2 is : bihar

PS D:\VSCode\NSUT\CP Sem1\Projects>
```

The status bar at the bottom indicates the current line is 62, column 2, with 4 spaces, UTF-8 encoding, CRLF line endings, and a Windows 32-bit architecture.

Program 10

```
File Edit Selection View Go Run Terminal Help
10.c - Projects - Visual Studio Code

EXPLORER
PROJECTS
1.c
1.exe
2.c
2.exe
3.c
3.exe
4.c
4.exe
5.c
5.exe
6.c
6.exe
7.c
7.exe
8.c
9.c
9.exe
10.c
11.c
11.exe
12.c
12.exe
13.c
14.c
program8.c
program12.c
tempCodeRunnerFile.c
test.c
test.exe
OUTLINE

10.c > info
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  char *ptr;
6  typedef struct information
7  {
8      char name[30];
9      char street[30];
10     char city[30];
11     char state[30];
12     char zipcode[30];
13 } info;
14
15 void add_mailing_address()
16 {
17     // int c;
18     // printf("size of dynamic mem: ");
19     // scanf("%d", &c);
20
21     ptr = (char *)malloc(144 * sizeof(char));
22     info a;
23     getchar();
24     printf("Enter your name\n");
25     gets(a.name);
26     printf("Enter your street name\n");
27     gets(a.street);
28     printf("Enter your city\n");
29     gets(a.city);
30     printf("Enter your state\n");
31     gets(a.state);
32     printf("Enter your zipcode\n");
33     gets(a.zipcode);
34 }
35 void delete_mailing_address()
36 {
37     free(ptr);
38     printf("done\n");
39 }
40
41 int main()
42 {
43     char str[4];
44     printf("Do you want to add your mailing address?\n");
45     scanf("%s", &str);
46
47     if (strcmp(str, "yes")==0)
```

Ln 14, Col 8 Spaces: 4 UTF-8 CRLF C Win32

Program 10

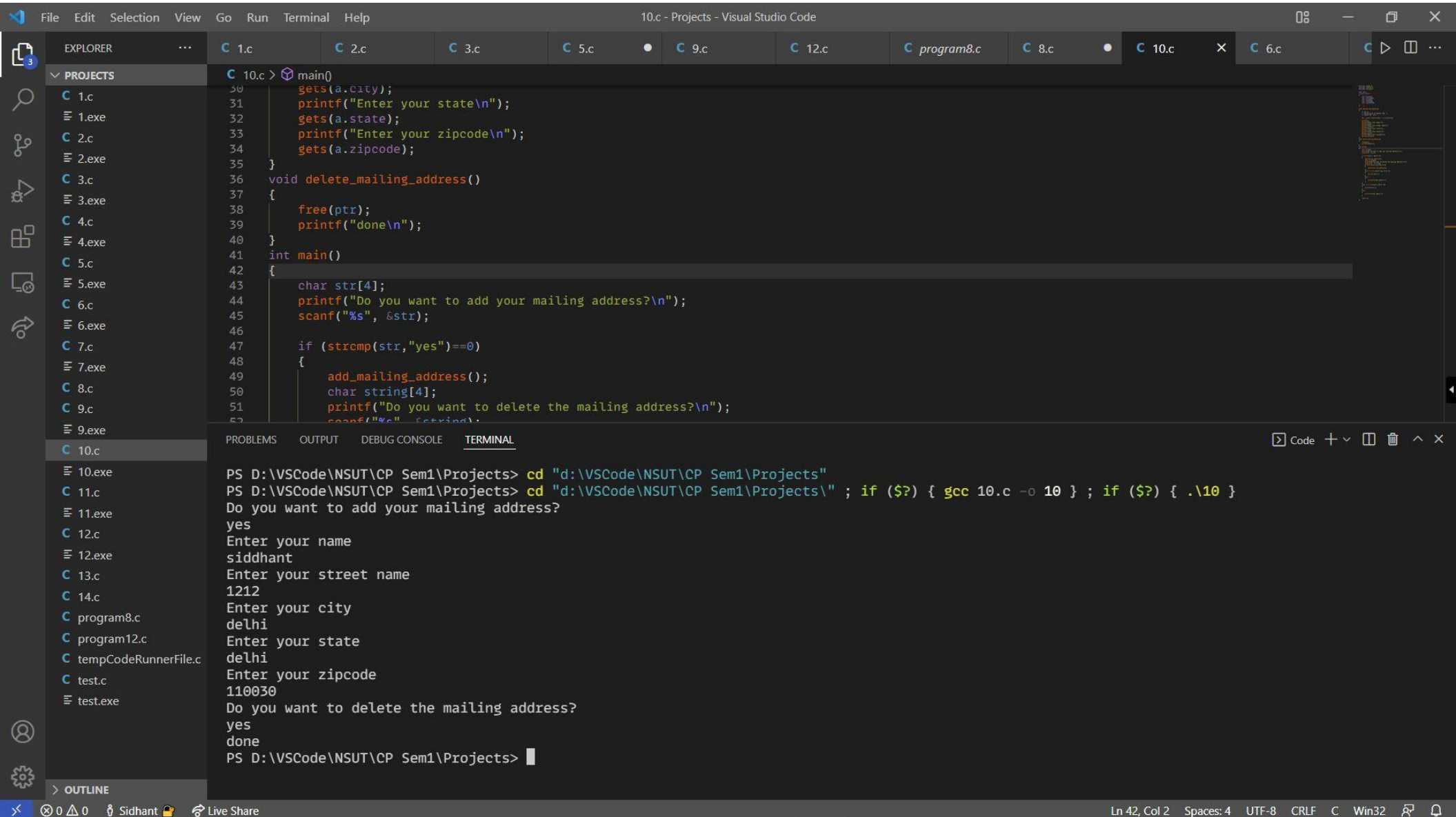
```
File Edit Selection View Go Run Terminal Help
10.c - Projects - Visual Studio Code

EXPLORER
PROJECTS
1.c
1.exe
2.c
2.exe
3.c
3.exe
4.c
4.exe
5.c
5.exe
6.c
6.exe
7.c
7.exe
8.c
9.c
9.exe
10.c
11.c
11.exe
12.c
12.exe
13.c
14.c
program8.c
program12.c
tempCodeRunnerFile.c
test.c
test.exe
OUTLINE

10.c > info
30 gets(a.city);
31 printf("Enter your state\n");
32 gets(a.state);
33 printf("Enter your zipcode\n");
34 gets(a.zipcode);
35 }
36 void delete_mailing_address()
37 {
38     free(ptr);
39     printf("done\n");
40 }
41 int main()
42 {
43     char str[4];
44     printf("Do you want to add your mailing address?\n");
45     scanf("%s", &str);
46
47     if (strcmp(str,"yes")==0)
48     {
49         add_mailing_address();
50         char string[4];
51         printf("Do you want to delete the mailing address?\n");
52         scanf("%s", &string);
53         if (strcmp(string,"yes")==0)
54         {
55             delete_mailing_address();
56         }
57         else if ( strcmp(string,"no")==0)
58         {
59             printf("ok\n");
60         }
61         else
62         {
63             printf("wrong input\n");
64         }
65     }
66     else if ( strcmp(str,"no") ==0)
67     {
68         printf("ok\n");
69     }
70     else
71     {
72         printf("wrong input\n");
73     }
74
75     return 0;
76 }
```

Ln 14, Col 8 Spaces: 4 UTF-8 CRLF C Win32

Program 10



Visual Studio Code interface showing a C program for managing mailing addresses. The Explorer sidebar on the left lists project files (1.c to 10.c) and executables (1.exe to 10.exe). The main editor displays the source code for 10.c, which includes functions for adding and deleting mailing addresses. The TERMINAL panel at the bottom shows the execution of the program, with user input for name, street name, city, state, and zip code.

```
10.c > main()
30     gets(a.city);
31     printf("Enter your state\n");
32     gets(a.state);
33     printf("Enter your zipcode\n");
34     gets(a.zipcode);
35 }
36 void delete_mailing_address()
37 {
38     free(ptr);
39     printf("done\n");
40 }
41 int main()
42 {
43     char str[4];
44     printf("Do you want to add your mailing address?\n");
45     scanf("%s", &str);
46
47     if (strcmp(str, "yes") == 0)
48     {
49         add_mailing_address();
50         char string[4];
51         printf("Do you want to delete the mailing address?\n");
52         scanf("%s", &string);
```

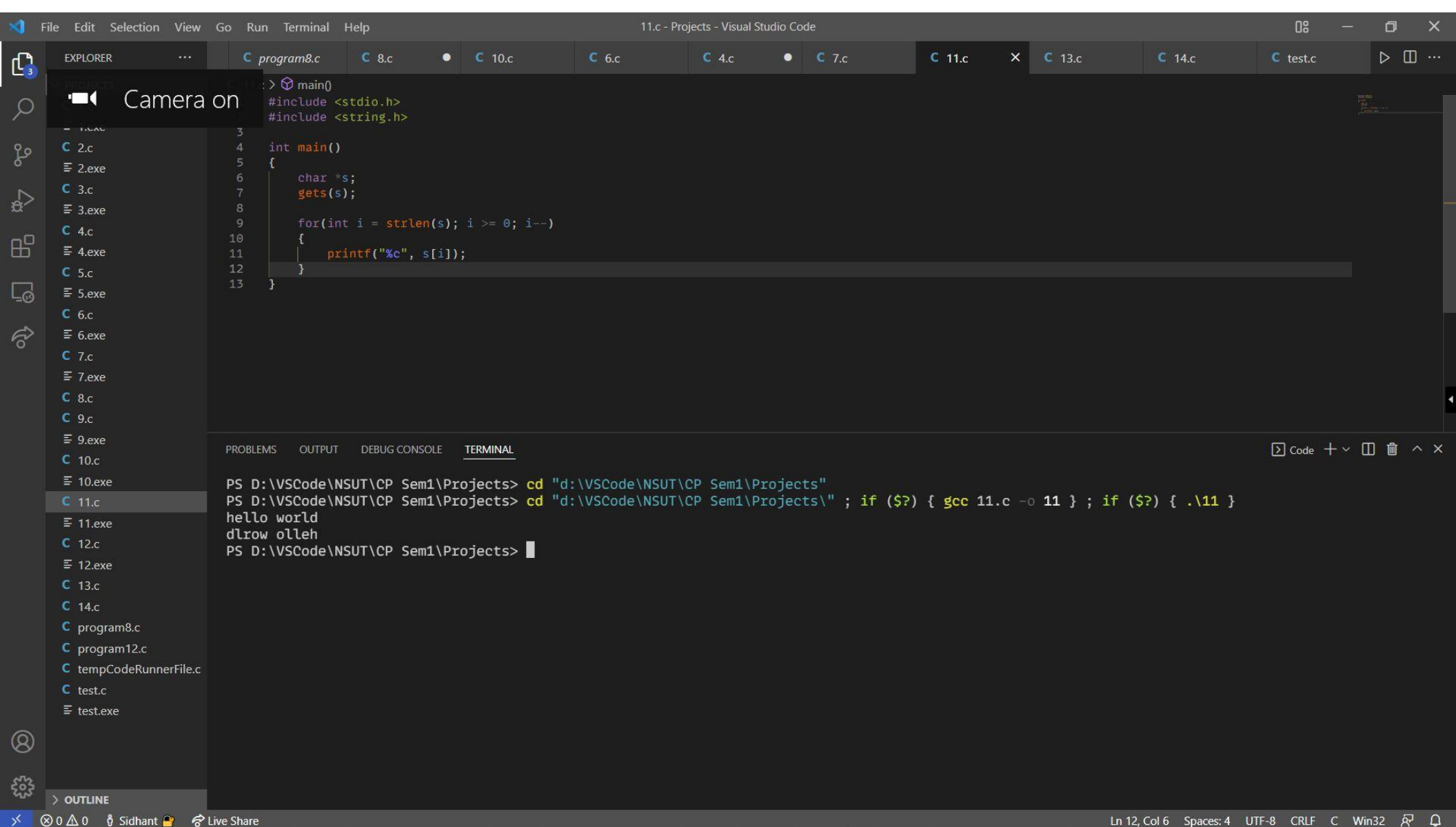
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\VSCode\NSUT\CP Sem1\Projects> cd "d:\VSCode\NSUT\CP Sem1\Projects"
PS D:\VSCode\NSUT\CP Sem1\Projects> cd "d:\VSCode\NSUT\CP Sem1\Projects\" ; if ($?) { gcc 10.c -o 10 } ; if ($?) { .\10 }
Do you want to add your mailing address?
yes
Enter your name
siddhant
Enter your street name
1212
Enter your city
delhi
Enter your state
delhi
Enter your zipcode
110030
Do you want to delete the mailing address?
yes
done
PS D:\VSCode\NSUT\CP Sem1\Projects>
```

OUTLINE

Ln 42, Col 2 Spaces: 4 UTF-8 CRLF C Win32

Program 11



Program 12

The screenshot shows the Visual Studio Code interface with a C program open in the editor. The Explorer sidebar on the left lists several files, including 1.c through 14.c, program8.c, program12.c, tempCodeRunnerFile.c, test.c, and test.exe. The file 12.c is selected and its content is displayed in the editor. The program is a C function that calculates and prints the Pascal's triangle for a given number of rows. The code includes a recursive function 'pascal' and a 'main' function that prompts the user for the number of rows and prints the triangle.

```
1  #include <stdio.h>
2
3  int pascal(int a, int b)
4  {
5      if (a > 0 && b == 1)
6      {
7          return 1;
8      }
9
10     else if (a == b)
11     {
12         return 1;
13     }
14     else
15     {
16         return pascal(a - 1, b - 1) + pascal(a - 1, b);
17     }
18 }
19
20 int main()
21 {
22     int num;
23     printf("how many rows of pascal triangle you wanted to see?\n");
24     scanf("%d", &num);
25
26     for (int i = 1; i < num + 1; i++)
27     {
28         for (int k = num - i; k > 0; k--)
29         {
30             printf(" ");
31         }
32         for (int j = 1; j < i + 1; j++)
33         {
34             printf("%d ", pascal(i, j));
35         }
36         printf("\n");
37     }
38
39     return 0;
40 }
41
42
```

The status bar at the bottom indicates the current line and column (Ln 42, Col 1), the number of spaces (4), the encoding (UTF-8), the line ending (CRLF), the file type (C), and the window state (Win32).

Program 12

The screenshot displays the Visual Studio Code interface with a C program named `12.c` open. The program calculates and prints a Pascal's triangle for a given number of rows. The code is as follows:

```
1  #include <stdio.h>
2
3  int pascal(int a, int b)
4  {
5      if (a > 0 && b == 1)
6      {
7          return 1;
8      }
9
10     else if (a == b)
11     {
12         return 1;
13     }
14     else
15     {
16         return pascal(a - 1, b - 1) + pascal(a - 1, b);
17     }
18 }
19
20 int main()
21 {
22     int num;
23     printf("how many rows of pascal triangle you wanted to see?\n");
```

The terminal window shows the execution of the program. The user enters `5` for the number of rows. The output is a Pascal's triangle with 5 rows:

```
PS D:\VSCode\NSUT\CP Sem1\Projects> cd "d:\VSCode\NSUT\CP Sem1\Projects"
PS D:\VSCode\NSUT\CP Sem1\Projects>
> cd "d:\VSCode\NSUT\CP Sem1\Projects\" ; if ($?) { gcc 12.c -o 12 } ; if ($?) { .\12 }

how many rows of pascal triangle you wanted to see?
5
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
PS D:\VSCode\NSUT\CP Sem1\Projects>
```

The status bar at the bottom indicates the current position is Line 8, Column 6, with 4 spaces, UTF-8 encoding, CRLF line endings, and a C locale. The window title is "12.c - Projects - Visual Studio Code".

Program 13

The screenshot displays the Visual Studio Code interface with a C program named 13.c. The Explorer sidebar on the left shows a project structure with files 1.c through 14.c and their corresponding executables. The main editor window shows the source code of 13.c, which includes headers for stdio.h and stdlib.h, and a main function that prompts the user for a filename, creates the file if it doesn't exist, and then reads and prints its contents line by line. The TERMINAL panel at the bottom shows the command prompt output, including the directory path, the compilation command using gcc, and the program's execution output where it reads 'hello this is abc.txt file thanks' from 'abc.txt'.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      FILE *fp;
7      char fname1[50],ch;
8      printf("Enter filename: ");
9      scanf("%s", fname1);
10
11     fp = fopen(fname1, "r");
12
13     if (fp == NULL)
14     {
15         fp = fopen(fname1, "w");
16         printf("\n%s file does not exist hence file created..", fname1);
17     }
18
19     while((ch=fgetc(fp))!=EOF)
20     {
21         printf("%c", ch);
22     }
23
24     fclose(fp);
25
26     return 0;
27 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\VSCode\NSUT\CP Sem1\Projects> cd "d:\VSCode\NSUT\CP Sem1\Projects"
PS D:\VSCode\NSUT\CP Sem1\Projects> cd "d:\VSCode\NSUT\CP Sem1\Projects\" ; if ($?) { gcc 13.c -o 13 } ; if ($?) { .\13 }
Enter filename: abc.txt
hello this is abc.txt file thanks
PS D:\VSCode\NSUT\CP Sem1\Projects>
```

0 0 Sidhant Live Share Ln 24, Col 16 Spaces: 4 UTF-8 CRLF C Win32

Program 14

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(int argc, char *argv[])
5  {
6      FILE *fp;
7
8      fp = fopen(argv[1], "w");
9      if(fp == NULL)
10     {
11         return 1;
12     }
13
14     printf("input: ");
15     char *s;
16     gets(s);
17
18     for(int i = 0; s[i] != '$'; i++)
19     {
20         fputc(s[i], fp);
21     }
22
23     fclose(fp);
24
25     return 0;
26 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\VSCode\NSUT\CP Sem1\Projects> .\14.exe xyz.txt
input: hello world this is xyz.txt thanks
PS D:\VSCode\NSUT\CP Sem1\Projects>
```

Ln 23, Col 16 Spaces: 4 UTF-8 CRLF C Win32