

The screenshot shows a dark-themed code editor interface, likely Visual Studio Code, with multiple tabs open in the background. The active tab is 'day19.c' (C++). The code implements a function to find the Least Common Multiple (LCM) of two integers.

```
c++ > C day19.c > ...
1 #include <stdio.h>
2
3 int main() {
4     int num1, num2, max;
5
6     // Input two numbers
7     printf("Enter two numbers: ");
8     scanf("%d %d", &num1, &num2);
9
10    // Find maximum of the two numbers
11    max = (num1 > num2) ? num1 : num2;
12
13    // Loop until a common multiple is found
14    while (1) {
15        if (max % num1 == 0 && max % num2 == 0) {
16            printf("LCM of %d and %d = %d\n", num1, num2, max);
17            break;
18        }
19        max++;
20    }
21
22    return 0;
23 }
```

The terminal window at the bottom shows the command to run the program and the resulting output:

```
cd "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day19.c -o day19 && "/Users/rishabhtrivedi/Documents/vs/c++/"day19
● rishabhtrivedi@Rishabhs-MacBook-Air vs & cd "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day19.c -o day19 && "/Users/rishabhtrivedi/Documents/vs/c++/"day19
y19
Enter two numbers: 4 5
LCM of 4 and 5 = 20
● rishabhtrivedi@Rishabhs-MacBook-Air c++ %
```

At the bottom right, the status bar indicates: Ln 24, Col 1 Spaces: 4 UTF-8 LF () C Go Live macos-clang-x64 Prettier.

The screenshot shows a terminal window with the following content:

```
cd "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day19.2.c -o day19.2 && "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day19.2.c -o day19.2 && "/Users/rishabhtrivedi/Documents/vs/c++/" && cd "/Users/rishabhtrivedi/Documents/vs/c++/" && ./day19.2
rishabhtrivedi@Rishabhs-MacBook-Air vs % cd "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day19.2.c -o day19.2 && "/Users/rishabhtrivedi/Documents/vs/c++/" && ./day19.2
Enter a number: 4
Sum of digits = 4
rishabhtrivedi@Rishabhs-MacBook-Air c++ %
```

The screenshot shows a dark-themed code editor interface, likely Visual Studio Code, displaying a C++ program named `day20.c`. The code is designed to calculate the product of odd digits of a given number.

```
1 #include <stdio.h>
2
3 int main() {
4     int num, digit;
5     int product = 1;
6     int hasOdd = 0; // flag to check if there is any odd digit
7
8     // Input a number
9     printf("Enter a number: ");
10    scanf("%d", &num);
11
12    // Loop through each digit
13    while (num != 0) {
14        digit = num % 10;      // Get last digit
15        if (digit % 2 != 0) { // Check if digit is odd
16            product *= digit; // Multiply it to product
17            hasOdd = 1;
18        }
19        num = num / 10;       // Remove last digit
20    }
21
22    // Display result
23    if (hasOdd)
24        printf("Product of odd digits = %d\n", product);
25    else
26        printf("No odd digits found.\n");
27
28    return 0;
29 }
30
```

The code uses standard input (`scanf`) and output (`printf`) functions from the `<stdio.h>` header. It iterates through each digit of the input number, checking if it is odd. If it is, it multiplies it to the `product`. A flag `hasOdd` is used to ensure at least one odd digit is found. Finally, it prints the result or a message if no odd digits were found.

In the bottom right corner of the terminal window, there are two tabs: `JavaSE-21...` and `Code c++`.

The bottom status bar displays the following information:

- Ln 30, Col 1
- Spaces: 4
- UTF-8
- LF
- (C)
- Go Live
- macos-clang-x64
- Prettier

The bottom left status bar shows the current workspace and file status:

- c++ master*
- 0 △ 0
- vs/c++
- Debug
- Indexing completed.

The screenshot shows a laptop screen displaying a Microsoft Visual Studio Code (VS Code) interface. The code editor has a dark theme and is showing a C++ file named `day20.2.c`. The code implements a function to calculate the 1's complement of a binary number entered as a string. The terminal below the editor shows the command-line interface used to run the program and its output.

```
c++ > C day20.2.c > ...
1 #include <stdio.h>
2
3 int main() {
4     char binary[50];
5     int i;
6
7     // Input a binary number as string
8     printf("Enter a binary number: ");
9     scanf("%s", binary);
10
11    // Find 1's complement
12    for (i = 0; binary[i] != '\0'; i++) {
13        if (binary[i] == '0')
14            binary[i] = '1';
15        else if (binary[i] == '1')
16            binary[i] = '0';
17        else {
18            printf("Invalid binary number.\n");
19            return 1;
20        }
21
22    }
23
24    // Display result
25    printf("1's Complement = %s\n", binary);
26
27 }
28
```

TERMINAL

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 1
+ ... | x
cd "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day20.c -o day20 && "/Users/rishabhtrivedi/Documents/vs/c++/"day20
rishabhtrivedi@Rishabhs-MacBook-Air vs % cd "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day20.c -o day20 && "/Users/rishabhtrivedi/Documents/vs/c++/"da
y20
Enter a number: 5
Product of odd digits = 5
rishabhtrivedi@Rishabhs-MacBook-Air c++ %
```

Ln 28, Col 1 Spaces: 4 UTF-8 LF ⚡ Go Live macos-clang-x64 Prettier

The screenshot shows a MacBook Air displaying the Visual Studio Code (VS Code) interface. The code editor window is open, showing a file named `day21.c`. The code implements a program to swap the first and last digits of a given number. The terminal window below shows the command to run the program, the user inputting the number 5, and the resulting swapped number 10.

```
c++ > C day21.c ...
1 #include <stdio.h>
2 #include <math.h>
3
4 int main() {
5     int num, firstDigit, lastDigit, digits, swappedNum, middlePart;
6
7     // Input a number
8     printf("Enter a number: ");
9     scanf("%d", &num);
10
11    // Find number of digits
12    digits = (int)log10(num) + 1;
13
14    // Get first and last digits
15    firstDigit = num / pow(10, digits - 1);
16    lastDigit = num % 10;
17
18    // Extract the middle part of the number
19    middlePart = (num % (int)pow(10, digits - 1)) / 10;
20
21    // Form the new swapped number
22    swappedNum = lastDigit * pow(10, digits - 1) + middlePart * 10 + firstDigit;
23
24    // Display result
25    printf("Number after swapping first and last digit: %d\n", swappedNum);
26
27    return 0;
28
29 }
```

TERMINAL

```
+ ~ | ⌂ x
cd "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day21.c -o day21 && "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day21.c -o day21 && "/Users/rishabhtrivedi/Documents/vs/c++/" da
rishabhtrivedi@Rishabhs-MacBook-Air vs % cd "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day21.c -o day21 && "/Users/rishabhtrivedi/Documents/vs/c++/" da
y21
Enter a number: 5
Number after swapping first and last digit: 10
rishabhtrivedi@Rishabhs-MacBook-Air c++ %
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 1

Ln 29, Col 1 Spaces: 4 UTF-8 LF () C ⌂ Go Live macos-clang-x64 ⌂ Prettier

The screenshot shows a dark-themed interface of a code editor, likely Visual Studio Code, displaying a C++ program. The code is intended to find perfect numbers by calculating the sum of proper divisors of a given number and checking if it equals the number itself.

```
2
3 int main() {
4     int num, i, sum = 0;
5
6     // Input a number
7     printf("Enter a number: ");
8     scanf("%d", &num);
9
10    // Find sum of proper divisors
11    for (i = 1; i <= num / 2; i++) {
12        if (num % i == 0) {
13            sum += i;
14        }
15    }
16
17    // Check if perfect
18    if (sum == num)
19        printf("%d is a Perfect Number.\n", num);
20    else
21        printf("%d is not a Perfect Number.\n", num);
22
23    return 0;
24
25 }
```

The code editor's sidebar shows a file tree under the 'VS' section, with a folder 'c++' containing files like day16.c through day22.c. The status bar at the bottom indicates the code is indexed.

The terminal below the code editor shows the command-line output of running the program:

```
cd "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day21.2.c -o day21.2 && "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day21.2.c -o day21.2 && "/Users/rishabhtrivedi/Documents/vs/c++/"
rishabhtrivedi@Rishabhs-MacBook-Air vs % cd "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day21.2.c -o day21.2 && "/Users/rishabhtrivedi/Documents/vs/c++/"
"/day21.2
Enter a number: 6
6 is a Perfect Number.
rishabhtrivedi@Rishabhs-MacBook-Air c++ %
```

The terminal also shows the current working directory as '/Users/rishabhtrivedi/Documents/vs/c++/' and the command used to run the program: 'cd "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day21.2.c -o day21.2 && "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day21.2.c -o day21.2 && "/Users/rishabhtrivedi/Documents/vs/c++/'.

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

```
cd "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day23.2.c -o day23.2 && "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day23.2.c -o day23.2 && "/Users/rishabhtrivedi/Documents/vs/c++/" && cd "/Users/rishabhtrivedi/Documents/vs/c++/" && ./day23.2
rishabhtrivedi@Rishabhs-MacBook-Air ~ % cd "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day23.2.c -o day23.2 && "/Users/rishabhtrivedi/Documents/vs/c++/" && ./day23.2
*****
*****
*****
*****
*****
rishabhtrivedi@Rishabhs-MacBook-Air ~ %
```

The screenshot shows a dark-themed interface of a code editor, likely Visual Studio Code, displaying a C++ file named `day24.c`. The code prints a right-angled triangle of stars using nested loops.

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

The code editor's sidebar shows a folder structure under `c++` containing files like `day20.2.c`, `day20.c`, etc. The terminal at the bottom shows the output of running the program, which prints a 5x5 right-angled star pattern:

```
y24
*
**
***
****
*****
```

The status bar at the bottom indicates the code is indexed and provides file navigation and indexing information.

The image shows a laptop screen with the Visual Studio Code interface. The code editor displays a C++ program named `day24.2.c`. The code prints a 5x5 identity matrix to the console. The terminal window shows the output of the program.

```
#include <stdio.h>
int main() {
    int i, j;
    // Outer loop for rows
    for (i = 1; i <= 5; i++) {
        // Inner loop for printing numbers
        for (j = 1; j <= i; j++) {
            printf("%d", j);
        }
        printf("\n"); // Move to next line after each row
    }
    return 0;
}
```

The terminal window shows the following output:

```
rishabhtrivedi@Rishabhs-MacBook-Air ~ % cd "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day24.2.c -o day24.2 && "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day24.2.c -o day24.2 && "/Users/rishabhtrivedi/Documents/vs/c++/" && ./day24.2
1
12
123
1234
12345
rishabhtrivedi@Rishabhs-MacBook-Air ~ %
```

The status bar at the bottom indicates the file is `day24.2.c`, the line is 17, column 1, and the file size is 0. It also shows indexing completed.

A screenshot of a dark-themed code editor, likely Visual Studio Code, showing a C++ file named `day25.c`. The code prints a right-angled triangle of numbers. The terminal below shows the output of the compiled program.

```
#include <stdio.h>

int main() {
    int i, j;

    // Outer loop for rows
    for (i = 5; i >= 1; i--) {
        // Inner loop for printing numbers
        for (j = i; j <= 5; j++) {
            printf("%d", j);
        }
        printf("\n"); // Move to next line after each row
    }

    return 0;
}
```

The terminal window shows the following output:

```
y25
5
45
345
2345
12345
```

At the bottom, the status bar indicates "Indexing completed."

A screenshot of the Visual Studio Code (VS Code) interface. The main area shows a C++ code editor with the following code:

```
#include <stdio.h>
int main() {
    int i, j, space;
    // Outer loop for rows
    for (i = 5; i >= 1; i--) {
        // Print leading spaces
        for (space = 5; space > i; space--) {
            printf(" ");
        }
        // Print stars
        for (j = 1; j <= i; j++) {
            printf("*");
        }
        printf("\n"); // Move to next line
    }
    return 0;
}
```

The code is part of a file named `day25.2.c`. The terminal at the bottom shows the output of running the code:

```
cd "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day25.2.c -o day25.2 && "/Users/rishabhtrivedi/Documents/vs/c++/"day25.2
rishabhtrivedi@Rishabhs-MacBook-Air vs % cd "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day25.2.c -o day25.2 && "/Users/rishabhtrivedi/Documents/vs/c++/"day25.2
*****
***
```

The status bar at the bottom indicates "Indexing completed." and shows the current file is `day25.2.c`.

A screenshot of the Visual Studio Code (VS Code) interface. The main area shows a C++ file named `day26.2.c` with the following code:

```
1 #include <stdio.h>
2
3 int main() {
4     // First pattern block
5     printf("*\n\n");
6
7     // Second pattern block
8     printf("*\n*\n*\n\n");
9
10    // Third pattern block
11    printf("*\n*\n*\n*\n\n");
12
13    // Fourth pattern block
14    printf("*\n*\n*\n\n");
15
16    // Fifth pattern block
17    printf("*\n");
18
19    return 0;
20}
21
```

The Explorer sidebar on the left lists various files in a folder named `c++`, including `day23.2.c`, `day23.c`, `day24.c`, `day24.2.c`, `day242.c`, `day24.c`, `day25.c`, `day25.2.c`, `day252.c`, `day252.2.c`, `day26.c`, `day26.2.c`, `day27.c`, `day27.2.c`, `day28.c`, `day28.2.c`, `day282.c`, `day282.2.c`, `day29.c`, `day29.2.c`, and `day30.c`. The file `day26.2.c` is currently selected.

The bottom status bar shows the terminal output:

```
rishabhtrivedi@Rishabhs-MacBook-Air c++ %
Ln 21, Col 1 Spaces: 4 UTF-8 LF () C Go Live macos-clang-x64 Prettier
c++ master*+ 0 △ 0 vs/c++ Debug Indexing completed.
```

The screenshot shows a dark-themed code editor interface, likely Visual Studio Code, displaying a C++ file named `day27.c`. The code prints two pyramids: an upper pyramid and a lower inverted pyramid, both made of asterisks (*). The terminal below the editor shows the output of the program, which is a symmetrical pyramid pattern.

```
#include <stdio.h>

int main() {
    int i, j, space;
    int n = 5; // height of the upper pyramid

    // Upper pyramid
    for (i = 1; i <= n; i++) {
        for (space = i; space < n; space++) {
            printf(" ");
        }
        for (j = 1; j <= (2 * i - 1); j++) {
            printf("*");
        }
        printf("\n");
    }

    // Lower inverted pyramid
    for (i = n - 1; i >= 1; i--) {
        for (space = n; space > i; space--) {
            printf(" ");
        }
        for (j = 1; j <= (2 * i - 1); j++) {
            printf("*");
        }
        printf("\n");
    }

    return 0;
}
```

The terminal output is:

```
rishabhtrivedi@Rishabhs-MacBook-Air c++ % *  
***  
*****  
*****  
*****  
****  
***  
*  
rishabhtrivedi@Rishabhs-MacBook-Air c++ % Debug  
Ln 31, Col 1 Spaces: 4 UTF-8 LF () C Go Live macos-clang-x64 Prettier
```

The screenshot shows a dark-themed instance of Visual Studio Code (VS Code) with several tabs open in the background:

- 3.2.c U
- day24.c U
- day24.2.c U
- day25.c U
- day25.2.c U
- day26.c U
- day26.2.c U
- day27.c U
- day27.2.c U
- day28.c ... U
- day28.2.c ... U
- day28.2.c U
- day29.c c++ U
- day29.2.c ... U
- day30.c c... U
- day30.2.c ... U

The current file, `day27.2.c`, contains C code to print two pyramids of asterisks. The code uses nested loops to determine the number of asterisks to print based on the current row index `i`. The output in the terminal window below shows the resulting pattern:

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

The status bar at the bottom indicates the following information:

- Ln 25, Col 1
- Spaces: 4
- UTF-8 LF
- (C)
- Go Live
- macos-clang-x64
- Prettier

The screenshot shows a dark-themed code editor interface, likely Visual Studio Code, displaying a C++ file named `day28.c`. The code prints prime numbers from 1 to `n`. It includes a prime number checking function and a loop to print primes. The terminal below shows the execution and output of the program.

```
c++ > C day28.c > ...
3 int main() {
10     printf("Prime numbers from 1 to %d are:\n", n);
11
12     // Loop through all numbers from 2 to n
13     for (i = 2; i <= n; i++) {
14         isPrime = 1; // assume number is prime
15
16         // Check divisibility
17         for (j = 2; j <= i / 2; j++) {
18             if (i % j == 0) {
19                 isPrime = 0; // not prime
20                 break;
21             }
22         }
23
24         // Print prime numbers
25         if (isPrime == 1)
26             printf("%d ", i);
27     }
28
29     printf("\n");
30     return 0;
31 }
32
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 1

```
cd "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day28.c -o day28 && "/Users/rishabhtrivedi/Documents/vs/c++/"day28
rishabhtrivedi@Rishabhs-MacBook-Air ~ % cd "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day28.c -o day28 && "/Users/rishabhtrivedi/Documents/vs/c++/"da
y28
Enter the value of n: 5
Prime numbers from 1 to 5 are:
2 3 5
rishabhtrivedi@Rishabhs-MacBook-Air c++ %
```

Ln 32, Col 1 Spaces: 4 UTF-8 LF ⌂ C ⌂ Go Live macos-clang-x64 ⌂ Prettier ⌂

The screenshot shows a laptop screen displaying a C++ code editor in Visual Studio Code (VS Code). The code in the editor is as follows:

```
#include <stdio.h>
int main() {
    int n, i;
    int arr[100]; // Declare an array of size 100
    // Input: number of elements
    printf("Enter number of elements: ");
    scanf("%d", &n);
    // Input: elements of the array
    printf("Enter %d elements:\n", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    // Output: elements of the array
    printf("The elements of the array are:\n");
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
    return 0;
}
```

The terminal tab at the bottom shows the following interaction:

```
Enter number of elements: 5
Enter 5 elements:
6778
789
8997
89
788
The elements of the array are:
6778 789 8997 89 788
```

The status bar at the bottom right indicates the following details:

Ln 26, Col 1 Spaces: 4 UTF-8 LF () C Go Live macos-clang-x64 Prettier

The screenshot shows a dark-themed interface of VS Code. In the Explorer sidebar, there's a folder named 'c++' containing files like day27, day27.2, day28, day28.2, day29, day29.2, day30, day30.2, and day32. The current file being edited is 'day29.c'. The code itself is as follows:

```
#include <stdio.h>
int main() {
    int n, i, sum = 0;
    int arr[100]; // Declare array of size 100

    // Input: number of elements
    printf("Enter number of elements: ");
    scanf("%d", &n);

    // Input: array elements
    printf("Enter %d elements:\n", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    // Calculate sum of elements
    for (i = 0; i < n; i++) {
        sum += arr[i];
    }

    // Output: sum of elements
    printf("Sum of array elements = %d\n", sum);
    return 0;
}
```

In the Terminal tab, the user has run the command to compile and execute the program:

```
cd "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day29.c -o day29 && "/Users/rishabhtrivedi/Documents/vs/c++/" day29
```

The terminal then prompts for the number of elements and the array values, showing the input and output of the program.

The image shows a screenshot of a code editor (Visual Studio Code) running on a Mac. The code editor has a dark theme. On the left is the Explorer sidebar showing a file tree under 'c++'. The main area displays the contents of 'day29.2.c'. The terminal at the bottom shows the execution of the program and its output.

```
#include <stdio.h>
int main() {
    int n, i;
    int arr[100];
    int max, min;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    printf("Enter %d elements:\n", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    max = min = arr[0];
    for (i = 1; i < n; i++) {
        if (arr[i] > max)
            max = arr[i];
        if (arr[i] < min)
            min = arr[i];
    }
    printf("Maximum element = %d\n", max);
    printf("Minimum element = %d\n", min);
    return 0;
}
```

cd "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day29.2.c -o day29.2 && "/Users/rishabhtrivedi/Documents/vs/c++/" ./day29.2
rishabhtrivedi@Rishabhs-MacBook-Air ~ % cd "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day29.2.c -o day29.2 && "/Users/rishabhtrivedi/Documents/vs/c++/" ./day29.2
Enter number of elements: 2
Enter 2 elements:
3456
345
Maximum element = 3456
Minimum element = 345

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface running on a Mac. The main area displays a C++ code editor with the following code:

```
c++ > C day30.c > ...
1 #include <stdio.h>
2
3 int main() {
4     int n, i;
5     int arr[100];
6     int even = 0, odd = 0;
7
8     // Input: number of elements
9     printf("Enter number of elements: ");
10    scanf("%d", &n);
11
12     // Input: array elements
13     printf("Enter %d elements:\n", n);
14     for (i = 0; i < n; i++) {
15         scanf("%d", &arr[i]);
16     }
17
18     // Count even and odd numbers
19     for (i = 0; i < n; i++) {
20         if (arr[i] % 2 == 0)
21             even++;
22         else
23             odd++;
24     }
25
26     // Output results
27     printf("Total even numbers = %d\n", even);
28     printf("Total odd numbers = %d\n", odd);
29
30     return 0;
31 }
32
```

The terminal tab at the bottom shows the following output from the terminal window:

```
cd "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day30.c -o day30 && "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day30.c -o day30 && "/Users/rishabhtrivedi/Documents/vs/c++/" da
y30
rishabhtrivedi@Rishabhs-MacBook-Air vs % cd "/Users/rishabhtrivedi/Documents/vs/c++/" && gcc day30.c -o day30 && "/Users/rishabhtrivedi/Documents/vs/c++/" da
y30
Enter number of elements: 2
Enter 2 elements:
345
4567
Total even numbers = 0
Total odd numbers = 2
rishabhtrivedi@Rishabhs-MacBook-Air c++ %
```

The status bar at the bottom indicates the following information: Ln 32, Col 1, Spaces: 4, UTF-8, LF, { } C, Go Live, macos-clang-x64, Prettier.

The screenshot shows a code editor interface with a dark theme. The top navigation bar includes tabs for 'day27.2.c U', 'day28.c U', 'day28.2.c U', 'day28.2.c U', 'day29.c U', 'day29.2.c U', 'day30.c U', and 'day30.2.c U'. The left sidebar displays a file tree under 'VS' with various C++ files like 'day27', 'day27.2', 'day28', 'day28.2', 'day29', 'day29.2', 'day30', and 'day30.2'. The main editor area contains the following C++ code:

```
#include <stdio.h>

int main() {
    int n, i;
    int arr[100];
    int positive = 0, negative = 0, zero = 0;

    // Input: number of elements
    printf("Enter number of elements: ");
    scanf("%d", &n);

    // Input: array elements
    printf("Enter %d elements:\n", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    // Count positive, negative, and zero elements
    for (i = 0; i < n; i++) {
        if (arr[i] > 0)
            positive++;
        else if (arr[i] < 0)
            negative++;
        else
            zero++;
    }

    // Output results
    printf("Positive numbers = %d\n", positive);
    printf("Negative numbers = %d\n", negative);
    printf("Zero elements = %d\n", zero);

    return 0;
}
```

The terminal window at the bottom shows the execution of the program:

```
"/"day30.2
Enter number of elements: 3
Enter 3 elements:
3456
345
4567
Positive numbers = 3
Negative numbers = 0
Zero elements = 0
rishabhtrivedi@Rishabhs-MacBook-Air c++ %
```

The status bar at the bottom indicates the current line (Ln 35, Col 1), spaces (Spaces: 4), encoding (UTF-8), and tools (Go Live, macos-clang-x64, Prettier).