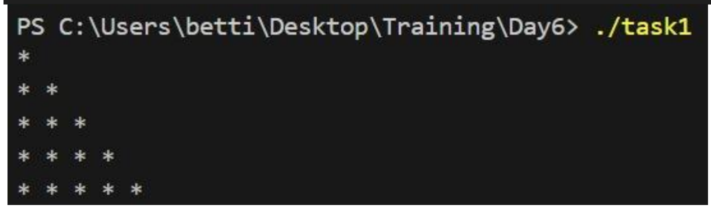
1.

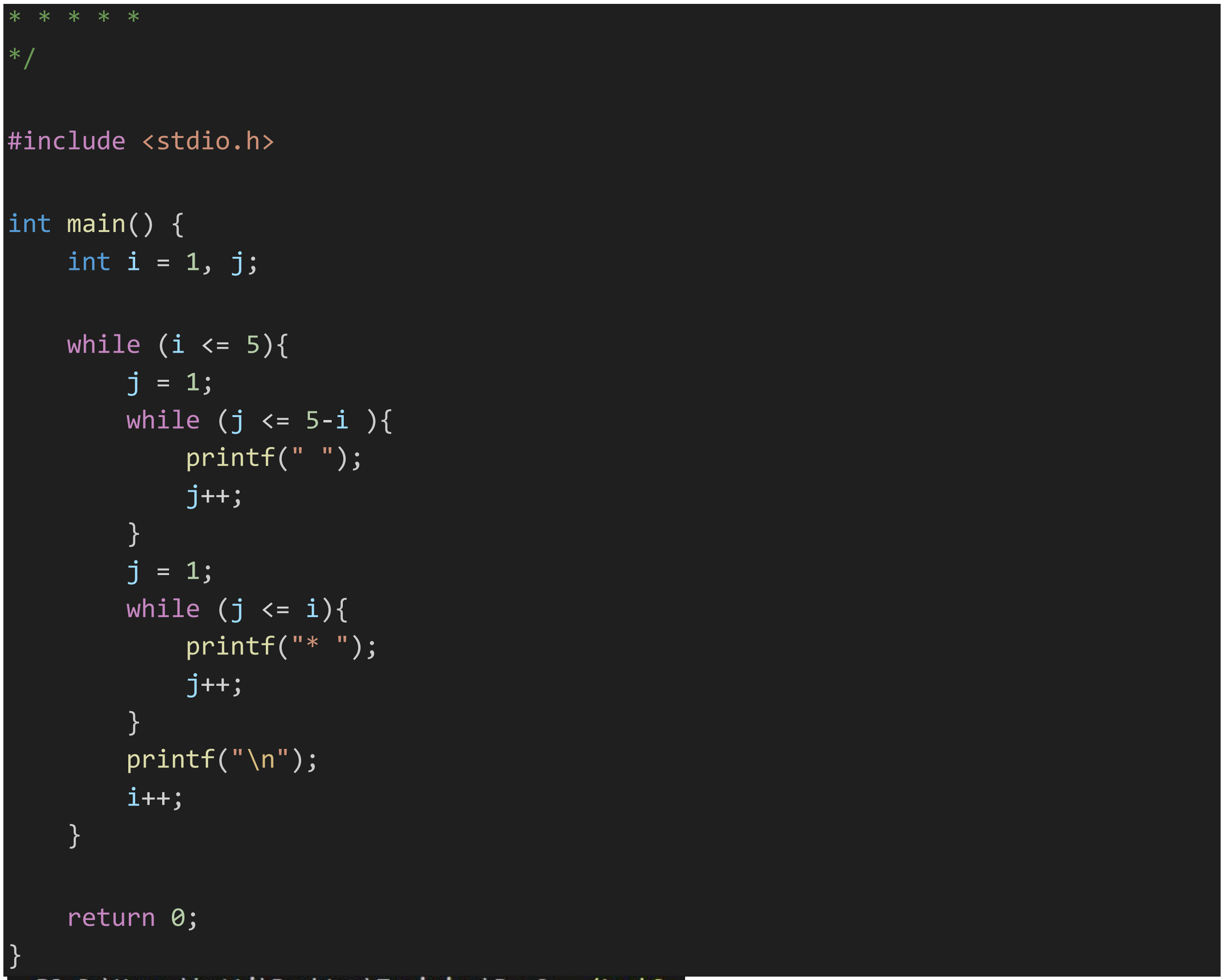
```
/*
program to print the below pattern
*
* *
* * *
* * * *
* * * * *
*/

#include <stdio.h>

int main() {
    int i = 0, j;

    while (i < 5){
        j = 0;
        while (j < i + 1){
            printf("* ");
            j++;
        }
        printf("\n");
        i++;
    }

    return 0;
}
```



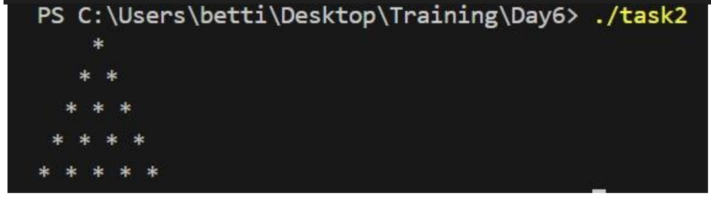
```
PS C:\Users\betti\Desktop\Training\Day6> ./task1
*
* *
* * *
* * * *
* * * * *
```

2.

```
/*
program to print the below pattern
    *
   * *
  * * *
 * * * *
```

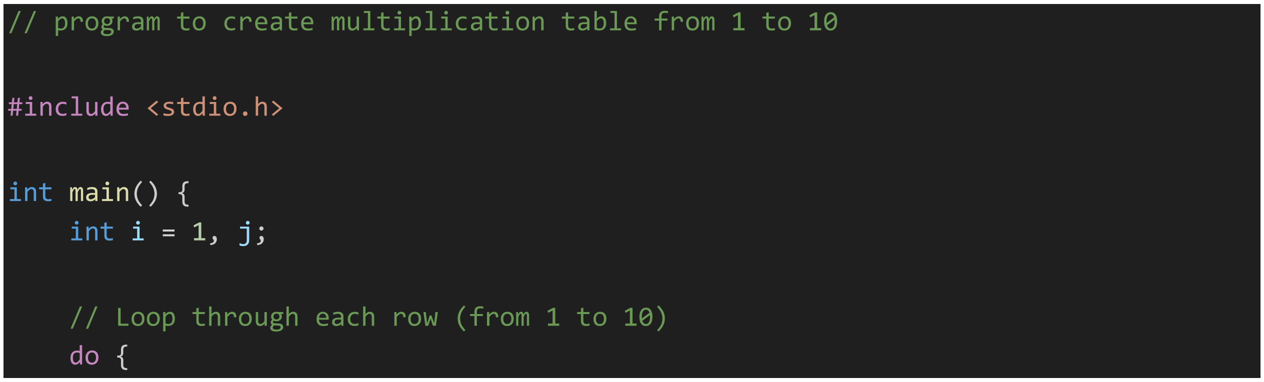


```
* * * * *
*/

#include <stdio.h>

int main() {
    int i = 1, j;

    while (i <= 5){
        j = 1;
        while (j <= 5-i ){
            printf(" ");
            j++;
        }
        j = 1;
        while (j <= i){
            printf("* ");
            j++;
        }
        printf("\n");
        i++;
    }

    return 0;
}
```

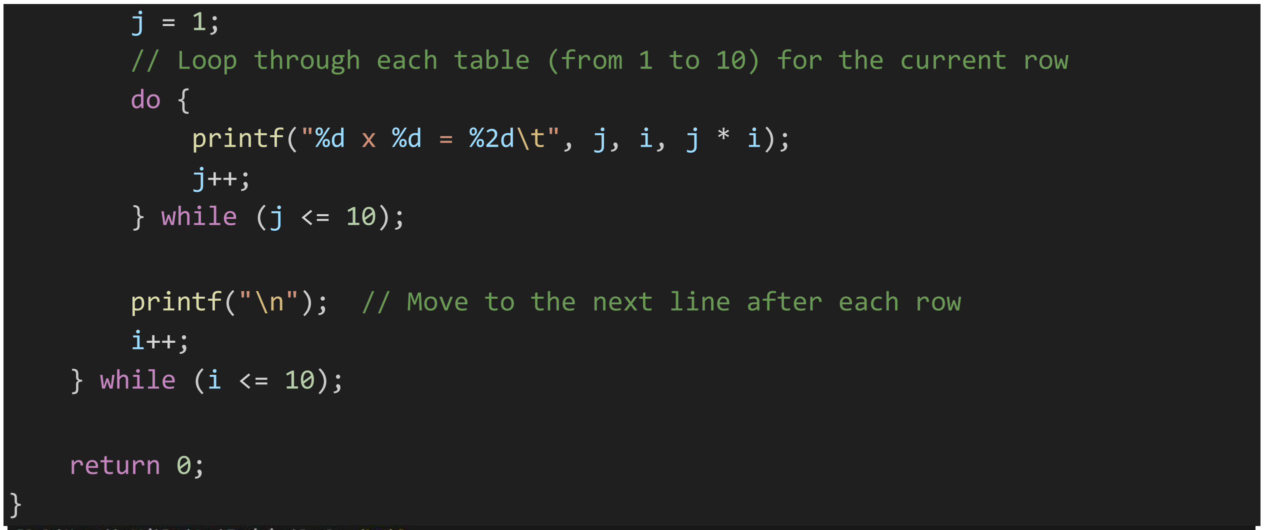


```
PS C:\Users\betti\Desktop\Training\Day6> ./task2
    *
   * *
  * * *
 * * * *
* * * * *
```

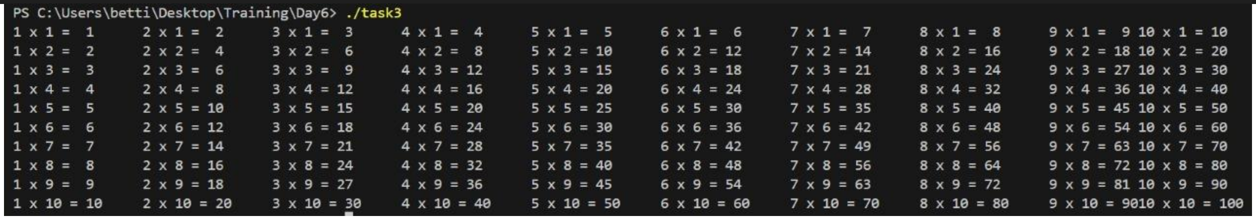
3.



```
// program to create multiplication table from 1 to 10

#include <stdio.h>

int main() {
    int i = 1, j;

    // Loop through each row (from 1 to 10)
    do {
```

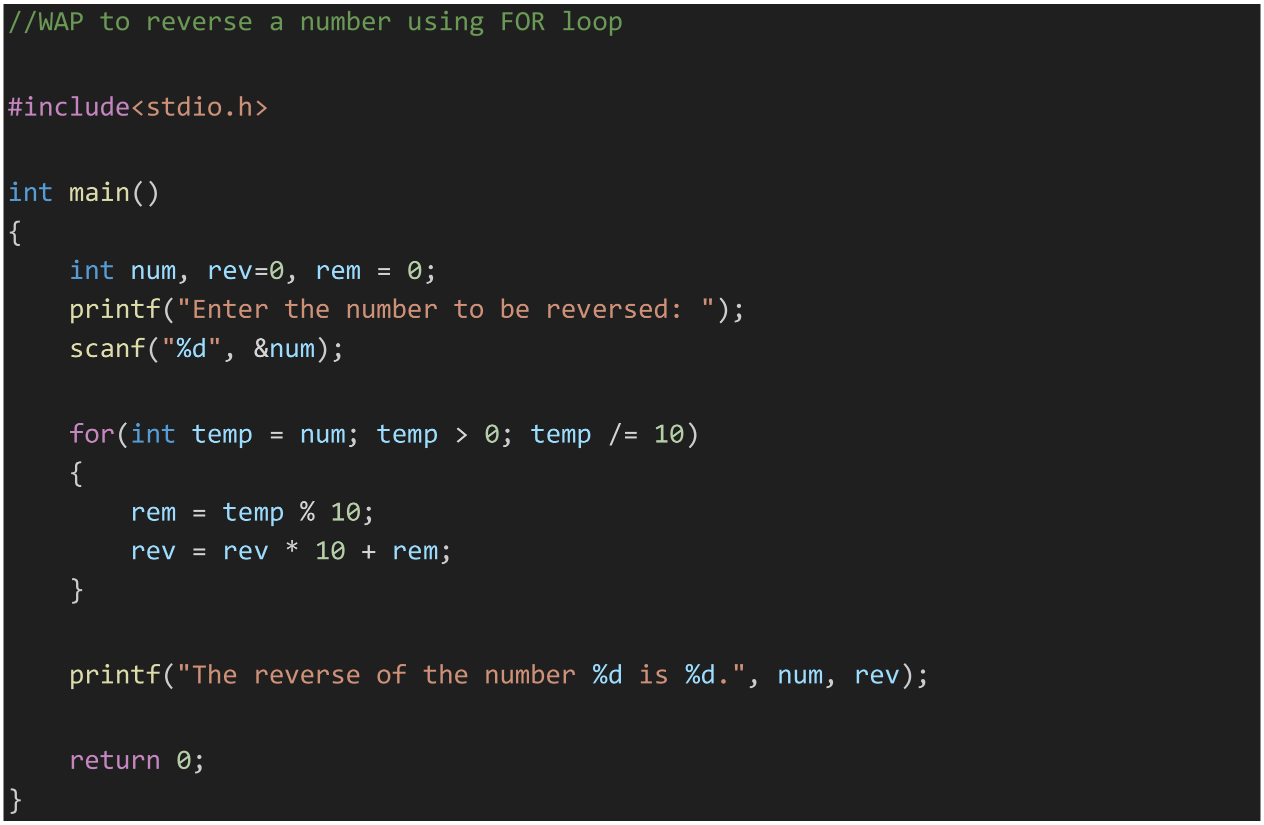


```
        j = 1;
        // Loop through each table (from 1 to 10) for the current row
        do {
            printf("%d x %d = %2d\t", j, i, j * i);
            j++;
        } while (j <= 10);

        printf("\n");  // Move to the next line after each row
        i++;
    } while (i <= 10);

    return 0;
}
```

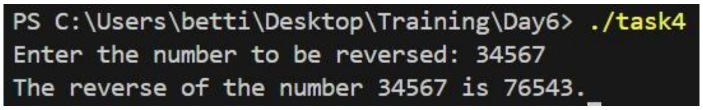


```
PS C:\Users\betti\Desktop\Training\Day6> ./task3
1 x 1 =  1      2 x 1 =  2      3 x 1 =  3      4 x 1 =  4      5 x 1 =  5      6 x 1 =  6      7 x 1 =  7      8 x 1 =  8      9 x 1 =  9 10 x 1 = 10
1 x 2 =  2      2 x 2 =  4      3 x 2 =  6      4 x 2 =  8      5 x 2 = 10      6 x 2 = 12      7 x 2 = 14      8 x 2 = 16      9 x 2 = 18 10 x 2 = 20
1 x 3 =  3      2 x 3 =  6      3 x 3 =  9      4 x 3 = 12      5 x 3 = 15      6 x 3 = 18      7 x 3 = 21      8 x 3 = 24      9 x 3 = 27 10 x 3 = 30
1 x 4 =  4      2 x 4 =  8      3 x 4 = 12      4 x 4 = 16      5 x 4 = 20      6 x 4 = 24      7 x 4 = 28      8 x 4 = 32      9 x 4 = 36 10 x 4 = 40
1 x 5 =  5      2 x 5 = 10      3 x 5 = 15      4 x 5 = 20      5 x 5 = 25      6 x 5 = 30      7 x 5 = 35      8 x 5 = 40      9 x 5 = 45 10 x 5 = 50
1 x 6 =  6      2 x 6 = 12      3 x 6 = 18      4 x 6 = 24      5 x 6 = 30      6 x 6 = 36      7 x 6 = 42      8 x 6 = 48      9 x 6 = 54 10 x 6 = 60
1 x 7 =  7      2 x 7 = 14      3 x 7 = 21      4 x 7 = 28      5 x 7 = 35      6 x 7 = 42      7 x 7 = 49      8 x 7 = 56      9 x 7 = 63 10 x 7 = 70
1 x 8 =  8      2 x 8 = 16      3 x 8 = 24      4 x 8 = 32      5 x 8 = 40      6 x 8 = 48      7 x 8 = 56      8 x 8 = 64      9 x 8 = 72 10 x 8 = 80
1 x 9 =  9      2 x 9 = 18      3 x 9 = 27      4 x 9 = 36      5 x 9 = 45      6 x 9 = 54      7 x 9 = 63      8 x 9 = 72      9 x 9 = 81 10 x 9 = 90
1 x 10 = 10     2 x 10 = 20     3 x 10 = 30     4 x 10 = 40     5 x 10 = 50     6 x 10 = 60     7 x 10 = 70     8 x 10 = 80     9 x 10 = 9010 x 10 = 100
```

4.

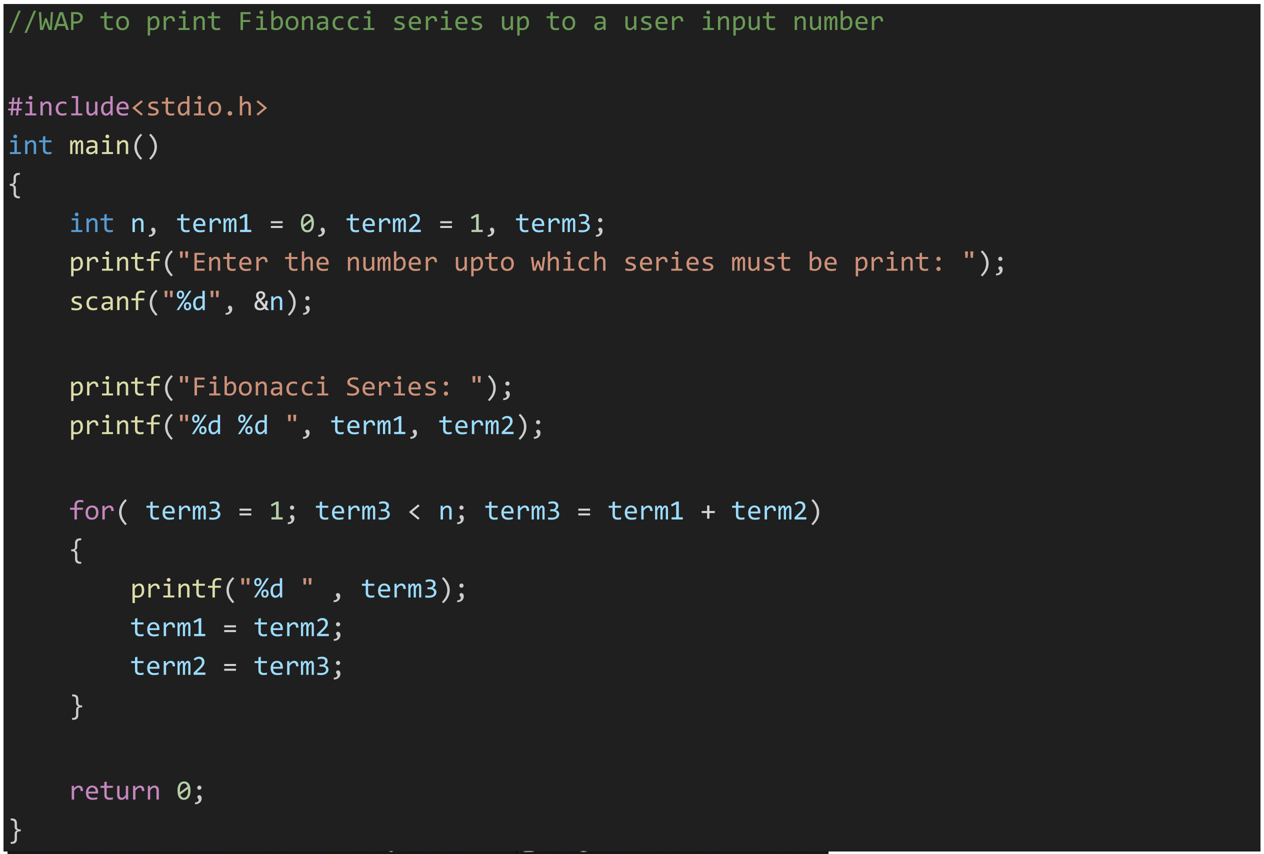


```
//WAP to reverse a number using FOR loop

#include<stdio.h>

int main()
{
    int num, rev=0, rem = 0;
    printf("Enter the number to be reversed: ");
    scanf("%d", &num);

    for(int temp = num; temp > 0; temp /= 10)
    {
        rem = temp % 10;
        rev = rev * 10 + rem;
    }

    printf("The reverse of the number %d is %d.", num, rev);

    return 0;
}
```

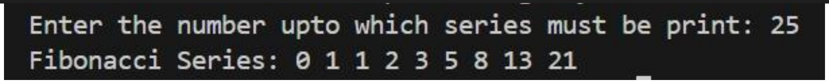


```
PS C:\Users\betti\Desktop\Training\Day6> ./task4
Enter the number to be reversed: 34567
The reverse of the number 34567 is 76543.
```

5.

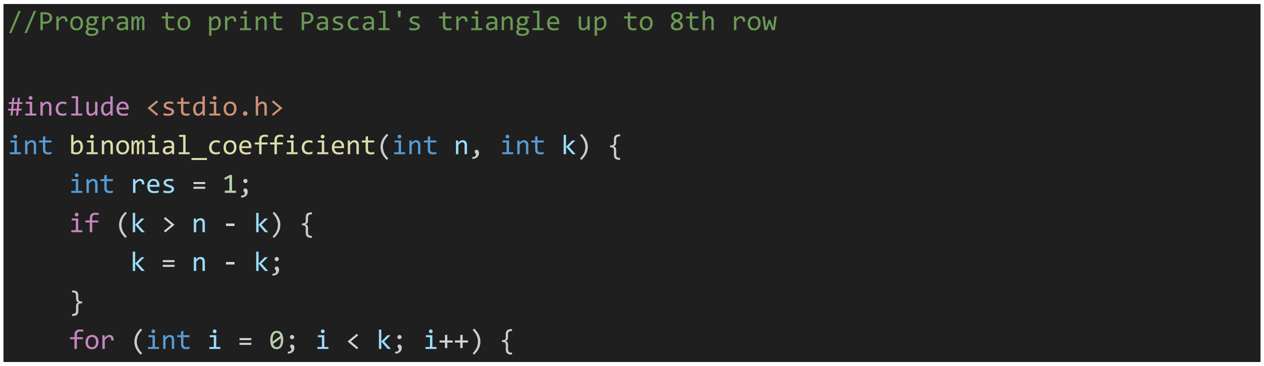


```c
//WAP to print Fibonacci series up to a user input number

#include<stdio.h>
int main()
{
    int n, term1 = 0, term2 = 1, term3;
    printf("Enter the number upto which series must be print: ");
    scanf("%d", &n);

    printf("Fibonacci Series: ");
    printf("%d %d ", term1, term2);

    for( term3 = 1; term3 < n; term3 = term1 + term2)
    {
        printf("%d " , term3);
        term1 = term2;
        term2 = term3;
    }

    return 0;
}
```

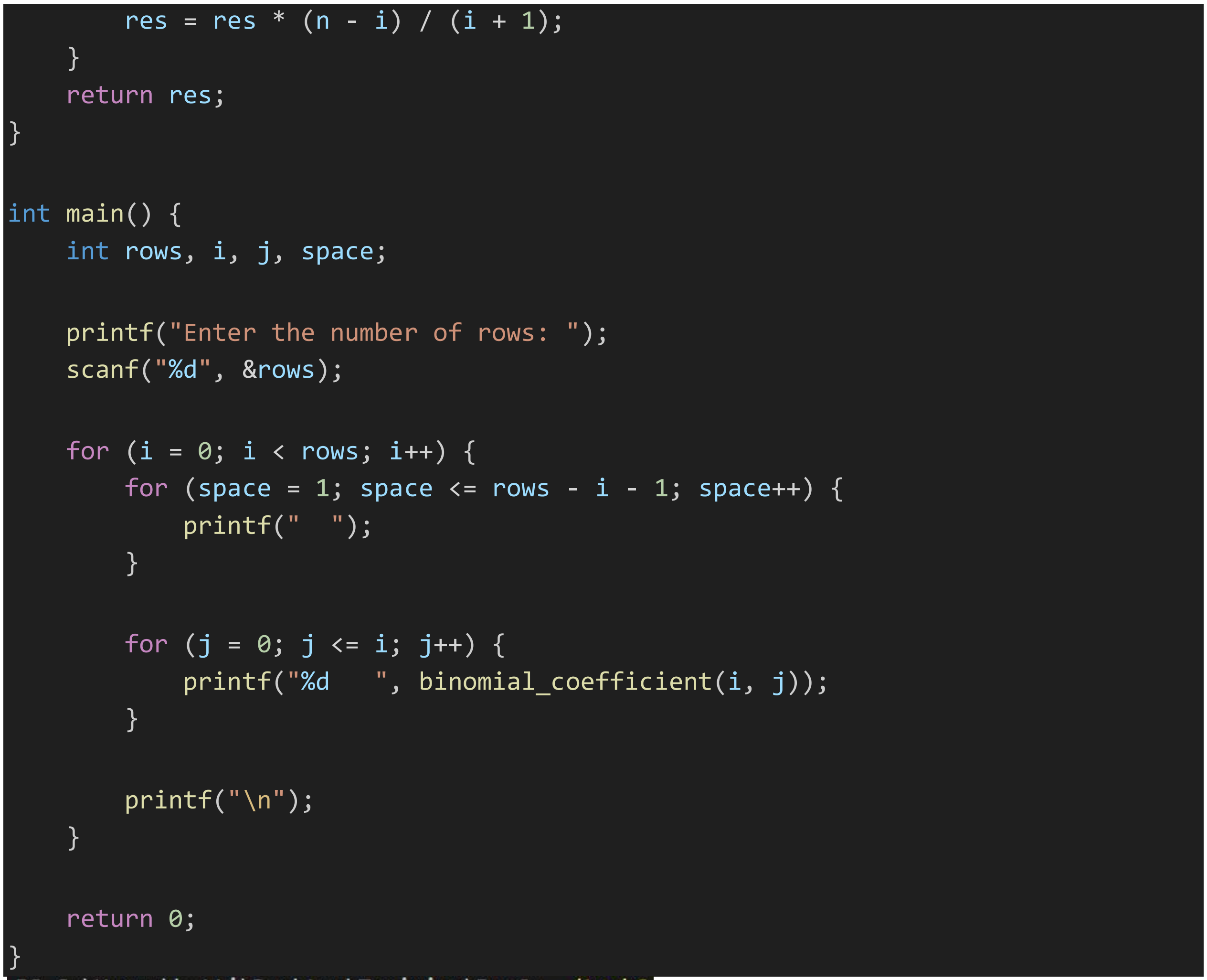


```
Enter the number upto which series must be print: 25
Fibonacci Series: 0 1 1 2 3 5 8 13 21
```

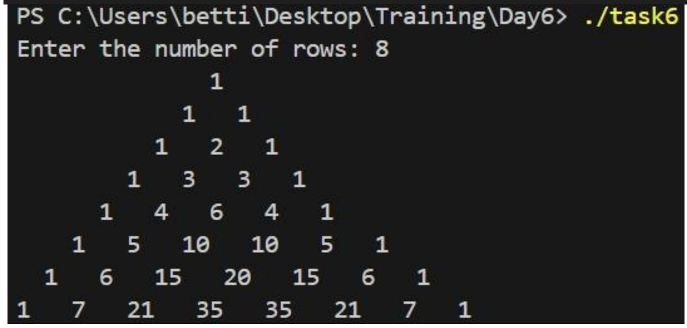
6.



```c
//Program to print Pascal's triangle up to 8th row

#include <stdio.h>
int binomial_coefficient(int n, int k) {
    int res = 1;
    if (k > n - k) {
        k = n - k;
    }
    for (int i = 0; i < k; i++) {
```

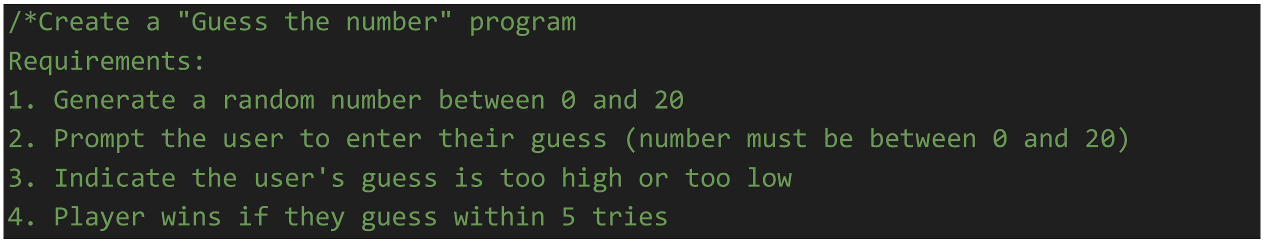


```c
        res = res * (n - i) / (i + 1);
    }
    return res;
}

int main() {
    int rows, i, j, space;

    printf("Enter the number of rows: ");
    scanf("%d", &rows);

    for (i = 0; i < rows; i++) {
        for (space = 1; space <= rows - i - 1; space++) {
            printf("  ");
        }

        for (j = 0; j <= i; j++) {
            printf("%d   ", binomial_coefficient(i, j));
        }

        printf("\n");
    }

    return 0;
}
```

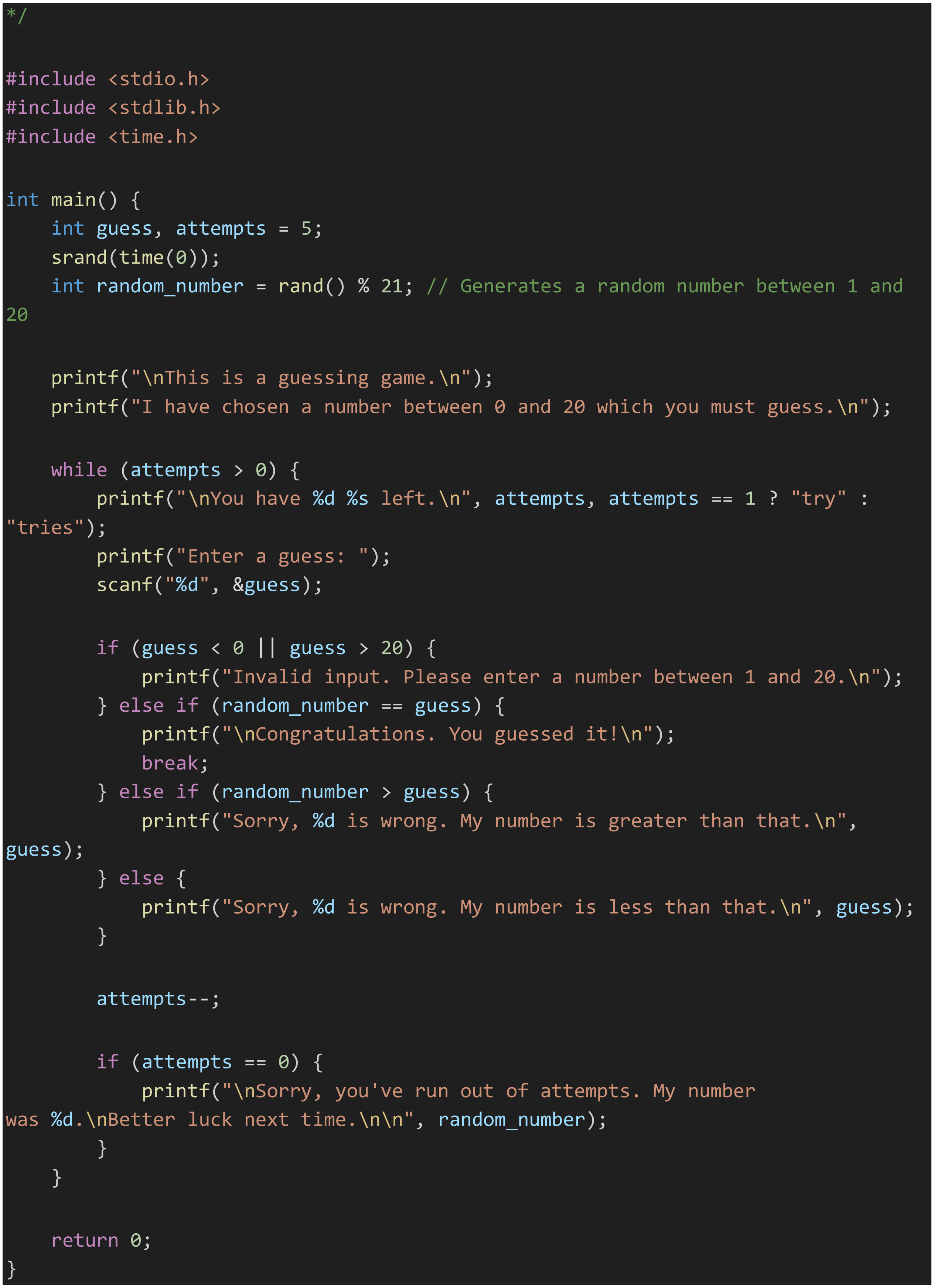


```
PS C:\Users\betti\Desktop\Training\Day6> ./task6
Enter the number of rows: 8
              1
            1   1
          1   2   1
        1   3   3   1
      1   4   6   4   1
    1   5   10   10   5   1
  1   6   15   20   15   6   1
1   7   21   35   35   21   7   1
```

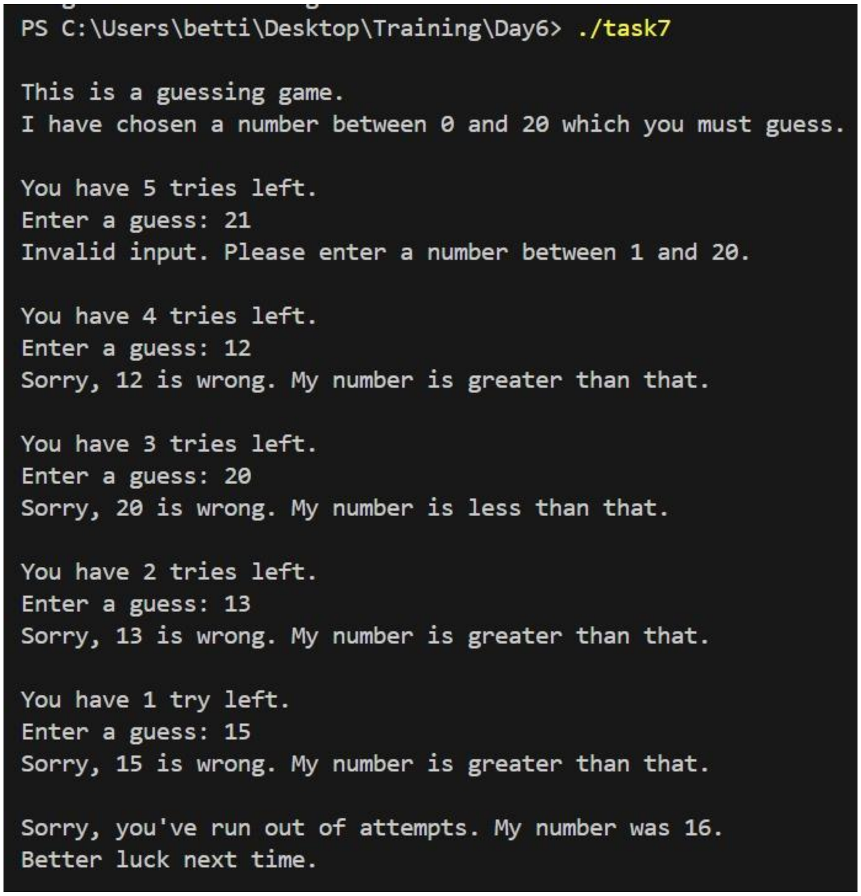
7.



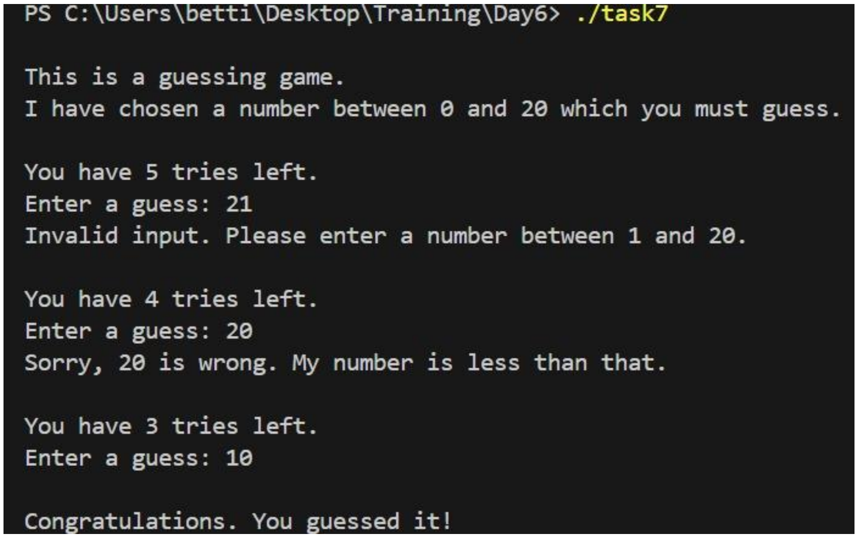
```c
/*Create a "Guess the number" program
Requirements:
1. Generate a random number between 0 and 20
2. Prompt the user to enter their guess (number must be between 0 and 20)
3. Indicate the user's guess is too high or too low
4. Player wins if they guess within 5 tries
```



```
*/

#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main() {
    int guess, attempts = 5;
    srand(time(0));
    int random_number = rand() % 21; // Generates a random number between 1 and
20

    printf("\nThis is a guessing game.\n");
    printf("I have chosen a number between 0 and 20 which you must guess.\n");

    while (attempts > 0) {
        printf("\nYou have %d %s left.\n", attempts, attempts == 1 ? "try" :
"tries");
        printf("Enter a guess: ");
        scanf("%d", &guess);

        if (guess < 0 || guess > 20) {
            printf("Invalid input. Please enter a number between 1 and 20.\n");
        } else if (random_number == guess) {
            printf("\nCongratulations. You guessed it!\n");
            break;
        } else if (random_number > guess) {
            printf("Sorry, %d is wrong. My number is greater than that.\n",
guess);
        } else {
            printf("Sorry, %d is wrong. My number is less than that.\n", guess);
        }

        attempts--;

        if (attempts == 0) {
            printf("\nSorry, you've run out of attempts. My number
was %d.\nBetter luck next time.\n\n", random_number);
        }
    }

    return 0;
}
```



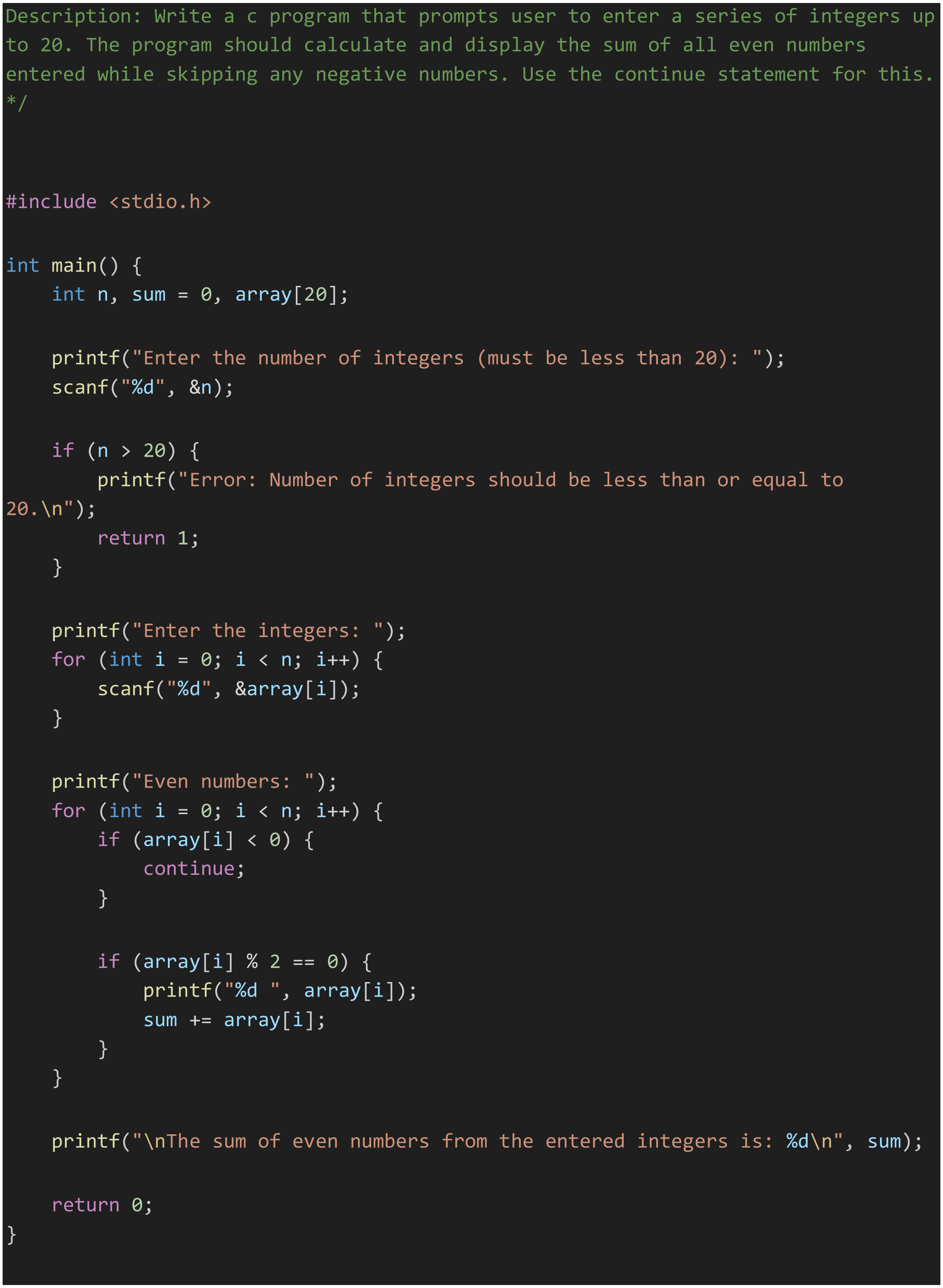
```
PS C:\Users\betti\Desktop\Training\Day6> ./task7

This is a guessing game.
I have chosen a number between 0 and 20 which you must guess.

You have 5 tries left.
Enter a guess: 21
Invalid input. Please enter a number between 1 and 20.

You have 4 tries left.
Enter a guess: 12
Sorry, 12 is wrong. My number is greater than that.

You have 3 tries left.
Enter a guess: 20
Sorry, 20 is wrong. My number is less than that.

You have 2 tries left.
Enter a guess: 13
Sorry, 13 is wrong. My number is greater than that.

You have 1 try left.
Enter a guess: 15
Sorry, 15 is wrong. My number is greater than that.

Sorry, you've run out of attempts. My number was 16.
Better luck next time.
```



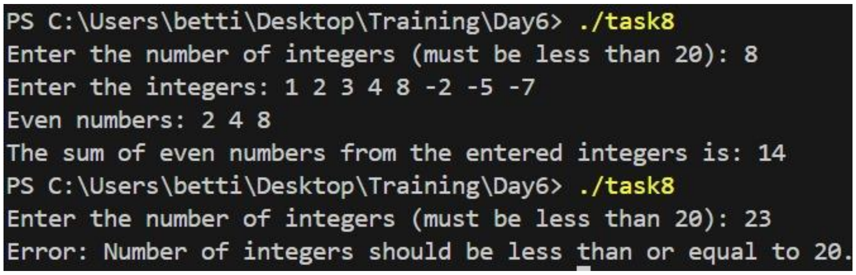
```
PS C:\Users\betti\Desktop\Training\Day6> ./task7

This is a guessing game.
I have chosen a number between 0 and 20 which you must guess.

You have 5 tries left.
Enter a guess: 21
Invalid input. Please enter a number between 1 and 20.

You have 4 tries left.
Enter a guess: 20
Sorry, 20 is wrong. My number is less than that.

You have 3 tries left.
Enter a guess: 10

Congratulations. You guessed it!
```

8.

```
/*Filter even numbers with continue
```



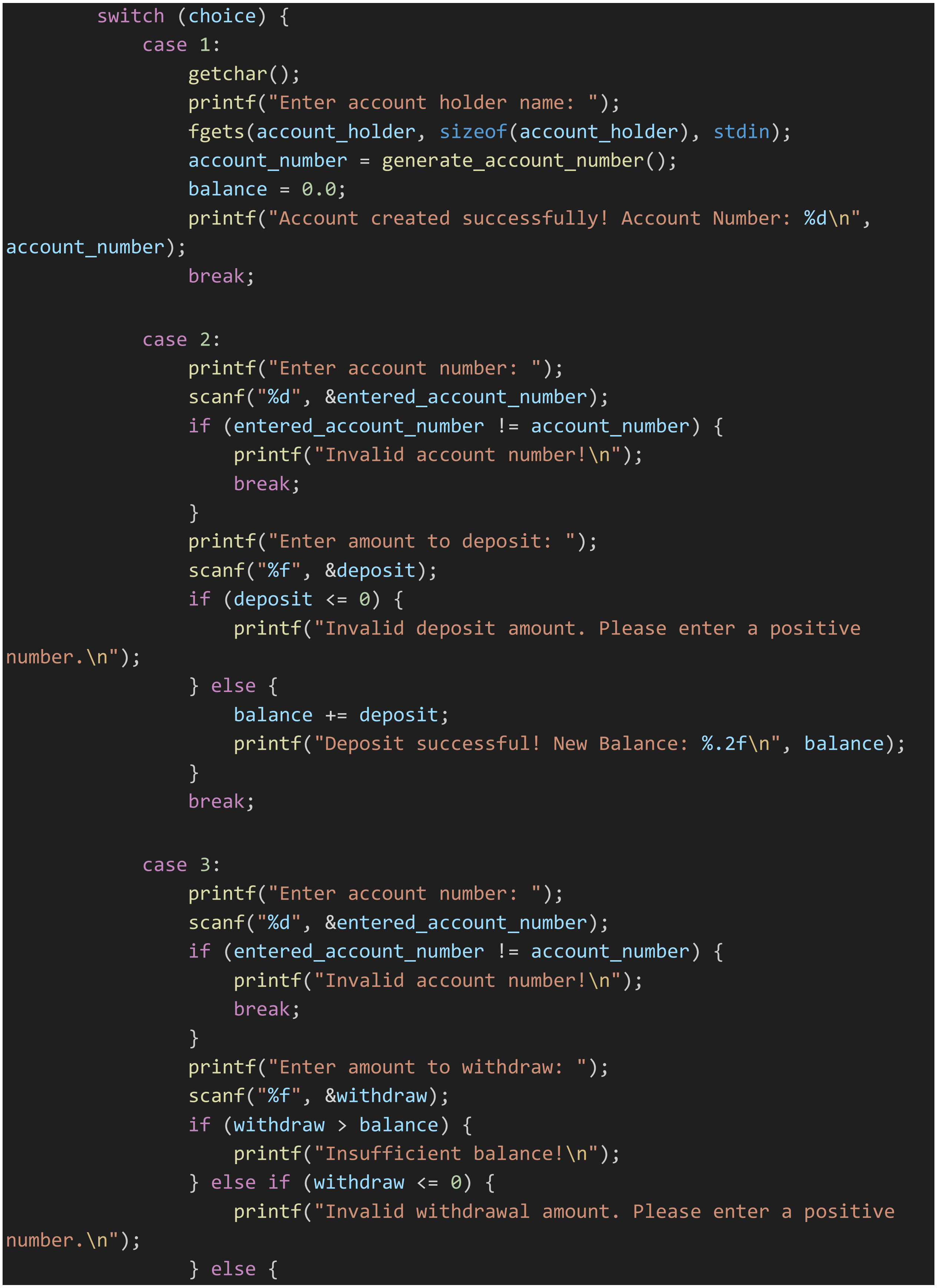
```c
Description: Write a c program that prompts user to enter a series of integers up
to 20. The program should calculate and display the sum of all even numbers
entered while skipping any negative numbers. Use the continue statement for this.
*/


#include <stdio.h>

int main() {
    int n, sum = 0, array[20];

    printf("Enter the number of integers (must be less than 20): ");
    scanf("%d", &n);

    if (n > 20) {
        printf("Error: Number of integers should be less than or equal to
20.\n");
        return 1;
    }

    printf("Enter the integers: ");
    for (int i = 0; i < n; i++) {
        scanf("%d", &array[i]);
    }

    printf("Even numbers: ");
    for (int i = 0; i < n; i++) {
        if (array[i] < 0) {
            continue;
        }

        if (array[i] % 2 == 0) {
            printf("%d ", array[i]);
            sum += array[i];
        }
    }

    printf("\nThe sum of even numbers from the entered integers is: %d\n", sum);

    return 0;
}
```



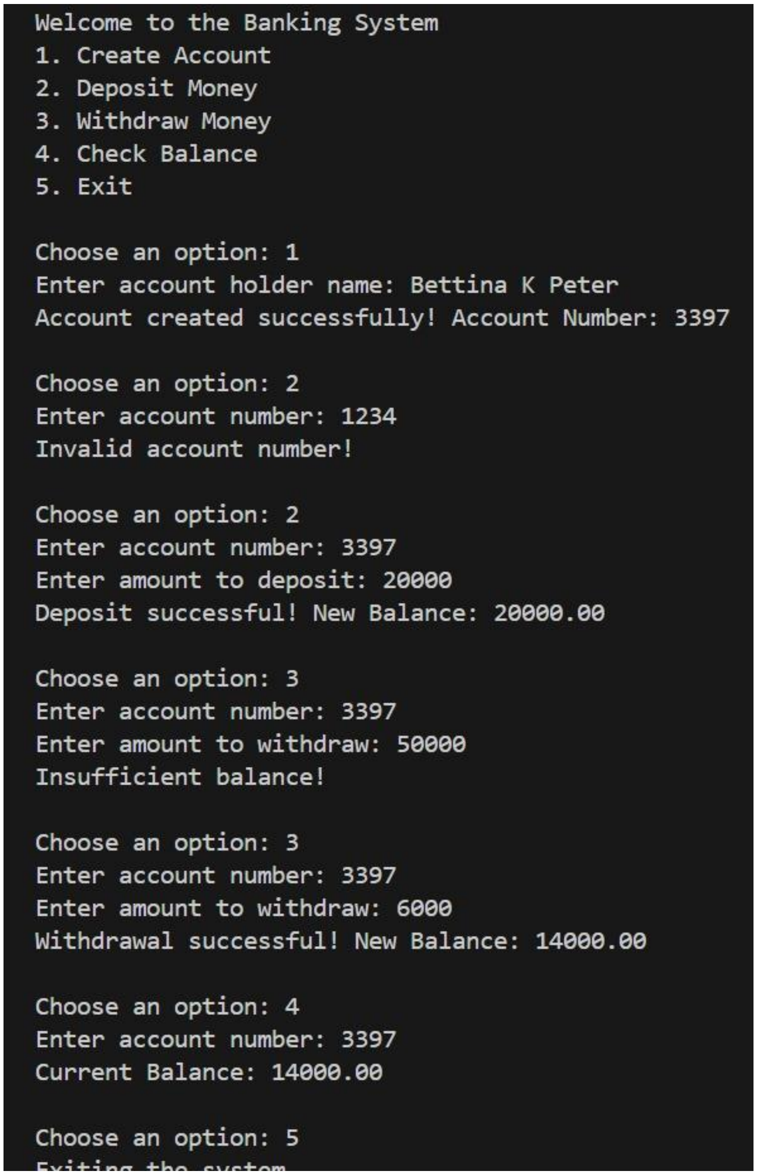
```
PS C:\Users\betti\Desktop\Training\Day6> ./task8
Enter the number of integers (must be less than 20): 8
Enter the integers: 1 2 3 4 8 -2 -5 -7
Even numbers: 2 4 8
The sum of even numbers from the entered integers is: 14
PS C:\Users\betti\Desktop\Training\Day6> ./task8
Enter the number of integers (must be less than 20): 23
Error: Number of integers should be less than or equal to 20.
```

9.

```
/*Problem Statement: Banking System Simulation

Description: Create a simple banking system simulation that allows user to create
an
account, deposit money, withdraw money, and check their balance. The program
should provide a menu-driven interface.

Requirements:
1. Use appropriate data types for account balance (e.g., float for monetary
values)
and user input (e.g., int for account numbers).
3. Use control statements to navigate through the menu options:
  i.    Create Account
  ii.   Deposit Money
  iii.  Withdraw Money
  iv.   Check Balance
4. Ensure that the withdrawal does not exceed the available balance and handle
invalid inputs gracefully.

Example Input/Output:
Welcome to the Banking System
1. Create Account
2. Deposit Money
3. Withdraw Money
4. Check Balance
5. Exit
Choose an option: 1
Enter account holder name: John Doe
Account created successfully! Account Number: 1001

Choose an option: 2
Enter account number: 1001
Enter amount to deposit: 500
Deposit successful! New Balance: 500.0

Choose an option: 3
Enter account number: 1001
Enter amount to withdraw: 200
Withdrawal successful! New Balance: 300.0

Choose an option: 4
Enter account number: 1001
Current Balance: 300.0

Choose an option: 5
Exiting the system.
*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

int generate_account_number() {
    return (rand() % 9000) + 1000;
}

int main() {
    char account_holder[50];
    int choice, account_number = 0, entered_account_number;
    float balance = 0.0, withdraw, deposit;
    srand(time(0));

    printf("Welcome to the Banking System\n");
    printf("1. Create Account\n");
    printf("2. Deposit Money\n");
    printf("3. Withdraw Money\n");
    printf("4. Check Balance\n");
    printf("5. Exit\n");

    while (1) {
        printf("\nChoose an option: ");
        scanf("%d", &choice);
```

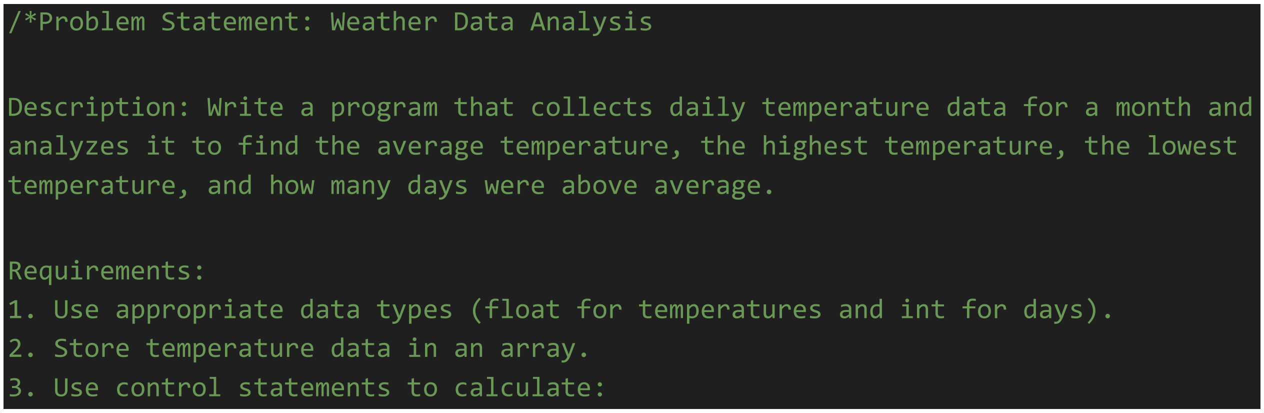


```
        switch (choice) {
            case 1:
                getchar();
                printf("Enter account holder name: ");
                fgets(account_holder, sizeof(account_holder), stdin);
                account_number = generate_account_number();
                balance = 0.0;
                printf("Account created successfully! Account Number: %d\n",
account_number);
                break;

            case 2:
                printf("Enter account number: ");
                scanf("%d", &entered_account_number);
                if (entered_account_number != account_number) {
                    printf("Invalid account number!\n");
                    break;
                }
                printf("Enter amount to deposit: ");
                scanf("%f", &deposit);
                if (deposit <= 0) {
                    printf("Invalid deposit amount. Please enter a positive
number.\n");
                } else {
                    balance += deposit;
                    printf("Deposit successful! New Balance: %.2f\n", balance);
                }
                break;

            case 3:
                printf("Enter account number: ");
                scanf("%d", &entered_account_number);
                if (entered_account_number != account_number) {
                    printf("Invalid account number!\n");
                    break;
                }
                printf("Enter amount to withdraw: ");
                scanf("%f", &withdraw);
                if (withdraw > balance) {
                    printf("Insufficient balance!\n");
                } else if (withdraw <= 0) {
                    printf("Invalid withdrawal amount. Please enter a positive
number.\n");
                } else {
                    balance -= withdraw;
                    printf("Withdrawal successful! New Balance: %.2f\n", 
balance);
                }
                break;

            case 4:
                printf("Enter account number: ");
                scanf("%d", &entered_account_number);
                if (entered_account_number != account_number) {
                    printf("Invalid account number!\n");
                } else {
                    printf("Current Balance: %.2f\n", balance);
                }
                break;

            case 5:
                printf("Exiting the system.\n");
                exit(0);

            default:
                printf("Invalid option! Please choose a number between 1 and 
5.\n");
                break;
        }
    }

    return 0;
}
```

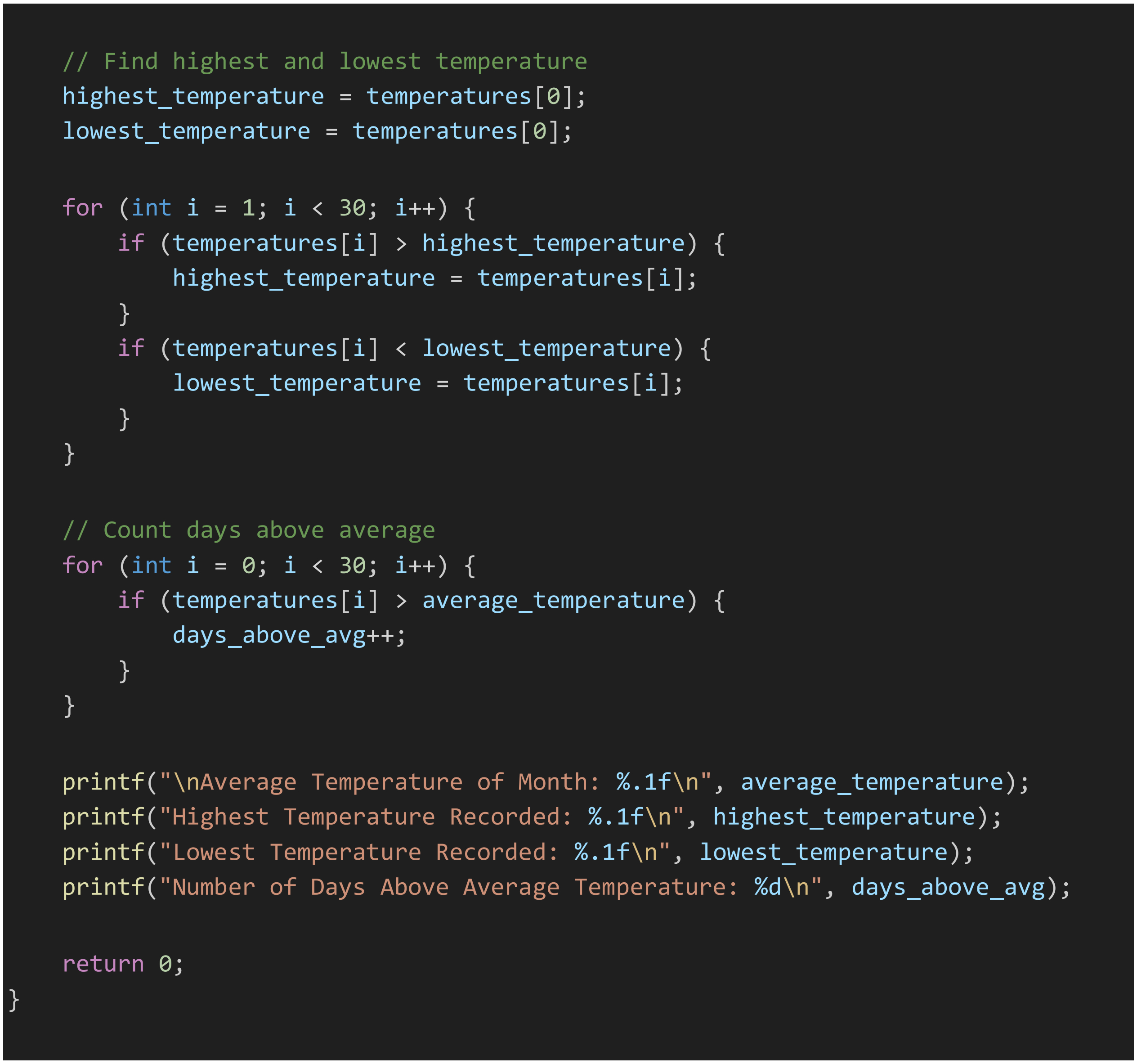


```
Welcome to the Banking System
1. Create Account
2. Deposit Money
3. Withdraw Money
4. Check Balance
5. Exit

Choose an option: 1
Enter account holder name: Bettina K Peter
Account created successfully! Account Number: 3397

Choose an option: 2
Enter account number: 1234
Invalid account number!

Choose an option: 2
Enter account number: 3397
Enter amount to deposit: 20000
Deposit successful! New Balance: 20000.00

Choose an option: 3
Enter account number: 3397
Enter amount to withdraw: 50000
Insufficient balance!

Choose an option: 3
Enter account number: 3397
Enter amount to withdraw: 6000
Withdrawal successful! New Balance: 14000.00

Choose an option: 4
Enter account number: 3397
Current Balance: 14000.00

Choose an option: 5
Exiting the system
```

10.



```
/*Problem Statement: Weather Data Analysis

Description: Write a program that collects daily temperature data for a month and
analyzes it to find the average temperature, the highest temperature, the lowest
temperature, and how many days were above average.

Requirements:
1. Use appropriate data types (float for temperatures and int for days).
2. Store temperature data in an array.
3. Use control statements to calculate:
  i.   Average Temperature of the month.
  ii.  Highest Temperature recorded.
  iii. Lowest Temperature recorded.
  iv.  Count of days with temperatures above average.
4. Handle cases where no data is entered.

Example Input/Output:
Enter temperatures for each day of the month (30 days):
Day 1 temperature: 72.5
Day 2 temperature: 68.0
...
Day 30 temperature: 75.0

Average Temperature of Month: XX.X
Highest Temperature Recorded: YY.Y
Lowest Temperature Recorded: ZZ.Z
Number of Days Above Average Temperature: N
*/

#include <stdio.h>

int main()
{
    float temperatures[30], total_temperature = 0.0, highest_temperature,
lowest_temperature, average_temperature;
    int days_above_avg = 0;

    printf("Enter temperatures for each day of the month (30 days):\n");
    for (int i = 0; i < 30; i++) {
        printf("Day %d temperature: ", i + 1);

        // Check for valid temperature input
        while (scanf("%f", &temperatures[i]) != 1) {
            printf("Invalid input. Please enter a valid temperature for Day %d:
", i + 1);
            while (getchar() != '\n'); // Clear invalid input from the buffer
        }

        total_temperature += temperatures[i];
    }

    // Calculate average temperature
    average_temperature = total_temperature / 30;
```



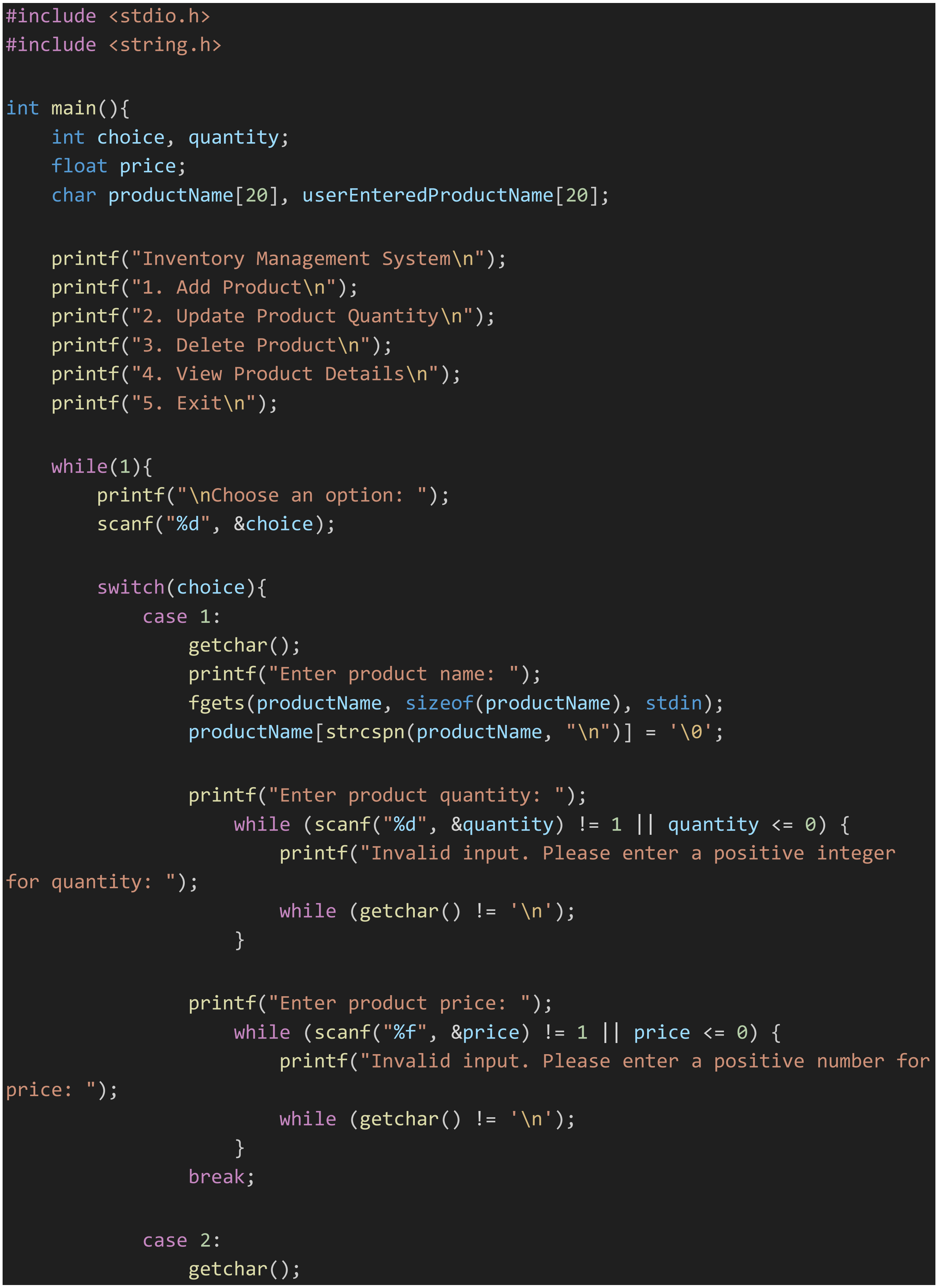
```c
    // Find highest and lowest temperature
    highest_temperature = temperatures[0];
    lowest_temperature = temperatures[0];

    for (int i = 1; i < 30; i++) {
        if (temperatures[i] > highest_temperature) {
            highest_temperature = temperatures[i];
        }
        if (temperatures[i] < lowest_temperature) {
            lowest_temperature = temperatures[i];
        }
    }

    // Count days above average
    for (int i = 0; i < 30; i++) {
        if (temperatures[i] > average_temperature) {
            days_above_avg++;
        }
    }

    printf("\nAverage Temperature of Month: %.1f\n", average_temperature);
    printf("Highest Temperature Recorded: %.1f\n", highest_temperature);
    printf("Lowest Temperature Recorded: %.1f\n", lowest_temperature);
    printf("Number of Days Above Average Temperature: %d\n", days_above_avg);

    return 0;
}
```

```
PS C:\Users\betti\Desktop\Training\Day6> ./task10

Enter temperatures for each day of the month (30 days):

Day 1 temperature: 15

Day 2 temperature: 17

Day 3 temperature: 10

Day 4 temperature: 0

Day 5 temperature: -4

Day 6 temperature: -10

Day 7 temperature: -6
Day 8 temperature: 1
Day 9 temperature: 8
Day 10 temperature: 12
Day 11 temperature: 16
Day 12 temperature: 18
Day 13 temperature: 20
Day 14 temperature: -=
Invalid input. Please enter a valid temperature for Day 14: 21
Day 15 temperature: 22
Day 16 temperature: 14
Day 17 temperature: 27
Day 18 temperature: 24
Day 19 temperature: 22
Day 20 temperature: 21
Day 21 temperature: 27
Day 22 temperature: 28
Day 23 temperature: 22
Day 24 temperature: 21
Day 25 temperature: 27
Day 26 temperature: 22
Day 27 temperature: 24
Day 28 temperature: 28
Day 29 temperature: 22
Day 30 temperature: 26

Average Temperature of Month: 16.5
Highest Temperature Recorded: 28.0
Lowest Temperature Recorded: -10.0
```

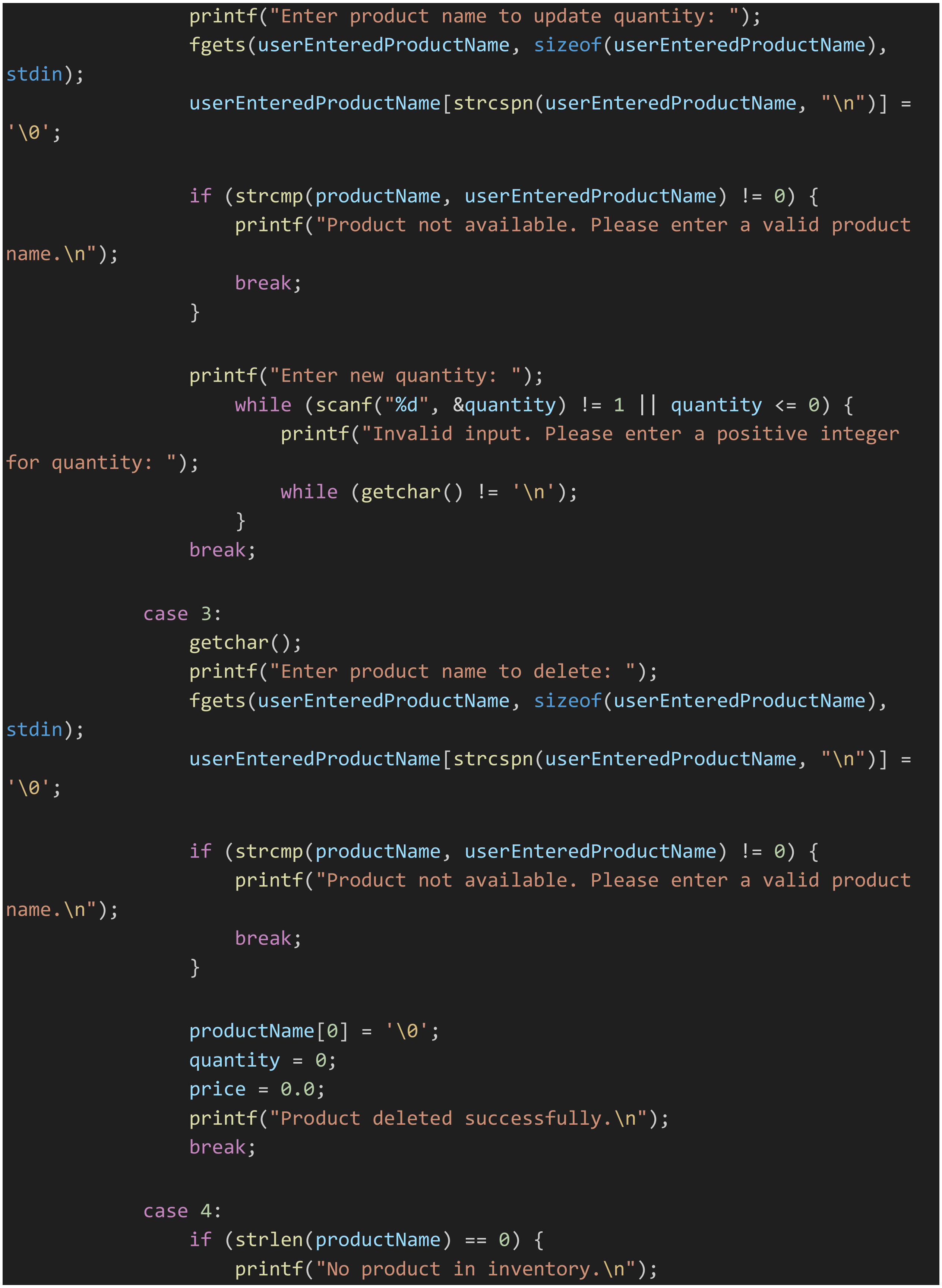
Number of Days Above Average Temperature: 19

11.

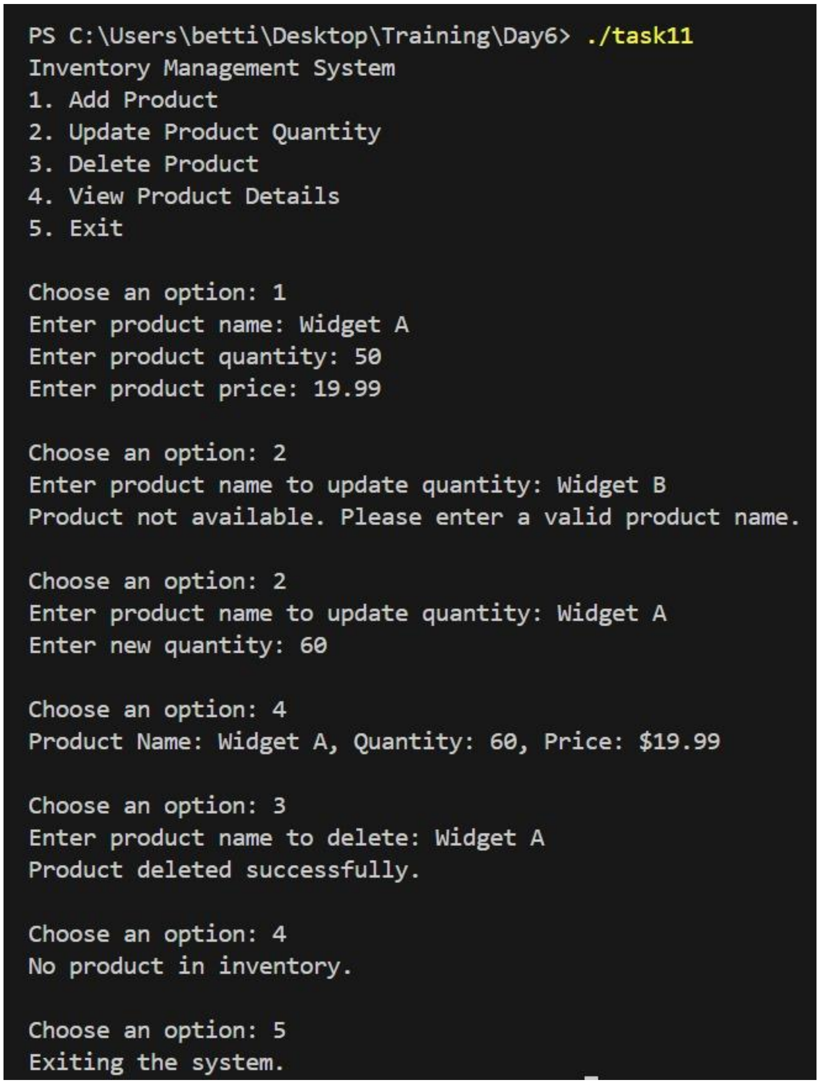
```
/*Problem Statement: Inventory Management System

Description: Create an inventory management system that allows users to manage
products in a store. Users should be able to add new products, update existing
product quantities, delete products, and view inventory details.

Requirements:
1. Use appropriate data types for product details (e.g., char arrays for product
names, int for quantities, float for prices).
2. Use control statements for menu-driven operations:
  i.   Add Product
  ii.  Update Product Quantity
  iii. Delete Product
  iv.  View All Products in Inventory
3. Ensure that the program handles invalid inputs and displays appropriate error
messages.


Example Input/Output:

Inventory Management System
1. Add Product
2. Update Product Quantity
3. Delete Product
4. View product details
5. Exit

Choose an option: 1
Enter product name: Widget A
Enter product quantity: 50
Enter product price: 19.99

Choose an option: 4
Product Name: Widget A, Quantity: 50, Price: $19.99

Choose an option: 5
Exiting the system.*/
```



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

int main(){
    int choice, quantity;
    float price;
    char productName[20], userEnteredProductName[20];

    printf("Inventory Management System\n");
    printf("1. Add Product\n");
    printf("2. Update Product Quantity\n");
    printf("3. Delete Product\n");
    printf("4. View Product Details\n");
    printf("5. Exit\n");

    while(1){
        printf("\nChoose an option: ");
        scanf("%d", &choice);

        switch(choice){
            case 1:
                getchar();
                printf("Enter product name: ");
                fgets(productName, sizeof(productName), stdin);
                productName[strcspn(productName, "\n")] = '\0';

                printf("Enter product quantity: ");
                    while (scanf("%d", &quantity) != 1 || quantity <= 0) {
                        printf("Invalid input. Please enter a positive integer for quantity: ");
                        while (getchar() != '\n');
                    }

                printf("Enter product price: ");
                    while (scanf("%f", &price) != 1 || price <= 0) {
                        printf("Invalid input. Please enter a positive number for price: ");
                        while (getchar() != '\n');
                    }
                break;

            case 2:
                getchar();
```



```
                printf("Enter product name to update quantity: ");
                fgets(userEnteredProductName, sizeof(userEnteredProductName),
stdin);
                userEnteredProductName[strcspn(userEnteredProductName, "\n")] =
'\0';

                if (strcmp(productName, userEnteredProductName) != 0) {
                    printf("Product not available. Please enter a valid product
name.\n");
                    break;
                }

                printf("Enter new quantity: ");
                    while (scanf("%d", &quantity) != 1 || quantity <= 0) {
                        printf("Invalid input. Please enter a positive integer
for quantity: ");
                        while (getchar() != '\n');
                    }
                break;

            case 3:
                getchar();
                printf("Enter product name to delete: ");
                fgets(userEnteredProductName, sizeof(userEnteredProductName),
stdin);
                userEnteredProductName[strcspn(userEnteredProductName, "\n")] =
'\0';

                if (strcmp(productName, userEnteredProductName) != 0) {
                    printf("Product not available. Please enter a valid product
name.\n");
                    break;
                }

                productName[0] = '\0';
                quantity = 0;
                price = 0.0;
                printf("Product deleted successfully.\n");
                break;

            case 4:
                if (strlen(productName) == 0) {
                    printf("No product in inventory.\n");
                } else {
                    printf("Product Name: %s, Quantity: %d, Price: $%.2f\n",
productName, quantity, price);
                }
                break;

            case 5:
                printf("Exiting the system.\n");
                return 0;

            default:
                printf("Invalid option. Please choose a number between 1 and
5.\n");
                break;
        }
    }
    return 0;
}
```



```
PS C:\Users\betti\Desktop\Training\Day6> ./task11
Inventory Management System
1. Add Product
2. Update Product Quantity
3. Delete Product
4. View Product Details
5. Exit

Choose an option: 1
Enter product name: Widget A
Enter product quantity: 50
Enter product price: 19.99

Choose an option: 2
Enter product name to update quantity: Widget B
Product not available. Please enter a valid product name.

Choose an option: 2
Enter product name to update quantity: Widget A
Enter new quantity: 60

Choose an option: 4
Product Name: Widget A, Quantity: 60, Price: $19.99

Choose an option: 3
Enter product name to delete: Widget A
Product deleted successfully.

Choose an option: 4
No product in inventory.

Choose an option: 5
Exiting the system.
```