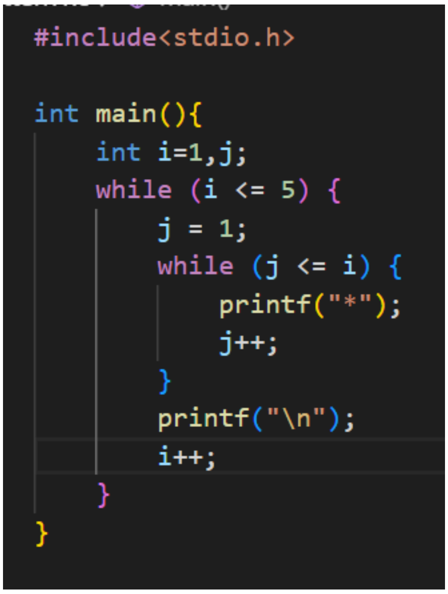
DAY 6-DAILY ASSIGNMENTS

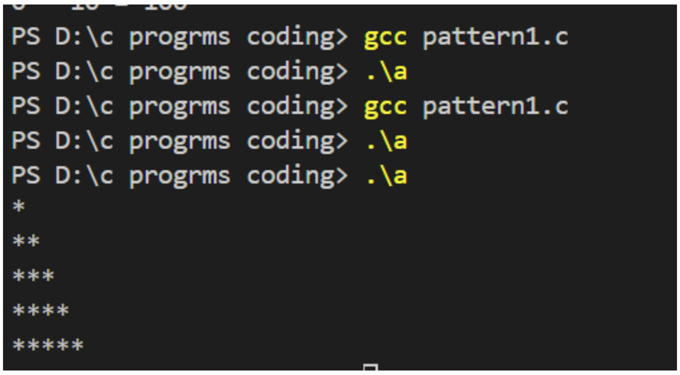
ANSU MARIUM SHIBU

1. Patterns using while

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

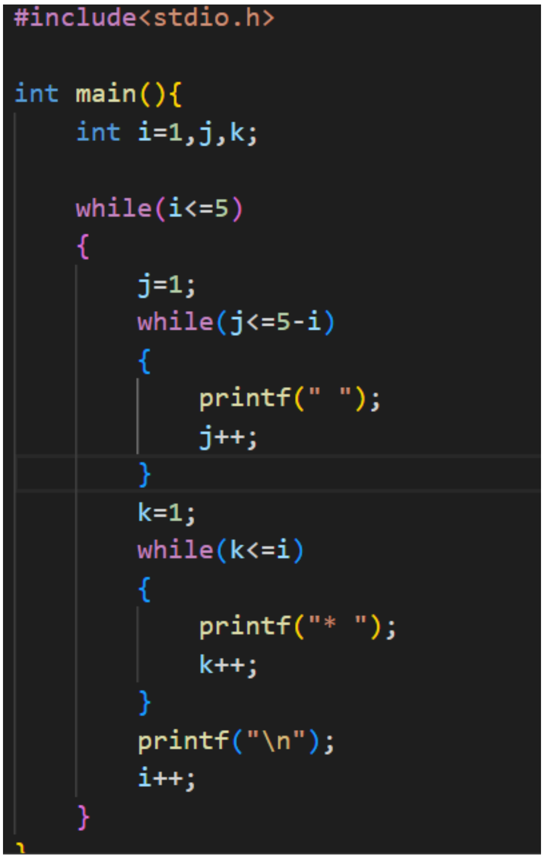


```c
#include<stdio.h>

int main(){
    int i=1,j;
    while (i <= 5) {
        j = 1;
        while (j <= i) {
            printf("*");
            j++;
        }
        printf("\n");
        i++;
    }
}
```

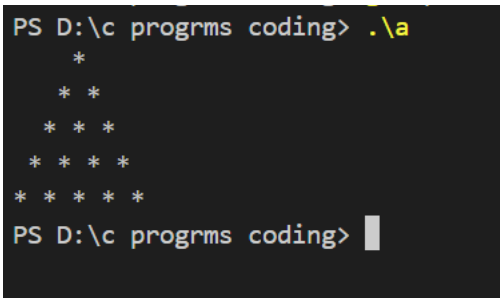


```
PS D:\c progrms coding> gcc pattern1.c
PS D:\c progrms coding> .\a
PS D:\c progrms coding> gcc pattern1.c
PS D:\c progrms coding> .\a
PS D:\c progrms coding> .\a
*
**
***
****
*****
```

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

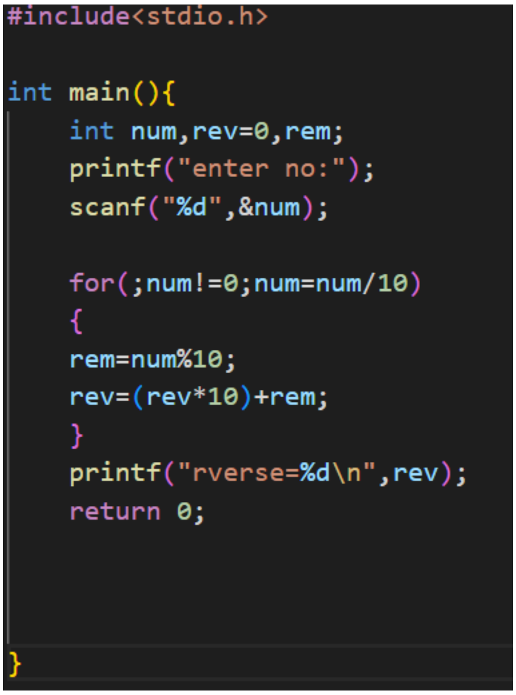


```c
#include<stdio.h>

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

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

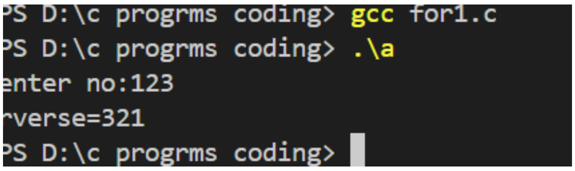


```
PS D:\c progrms coding> .\a
    *
   * *
  * * *
 * * * *
* * * * *
PS D:\c progrms coding> 
```

2.Reverse number using for loop

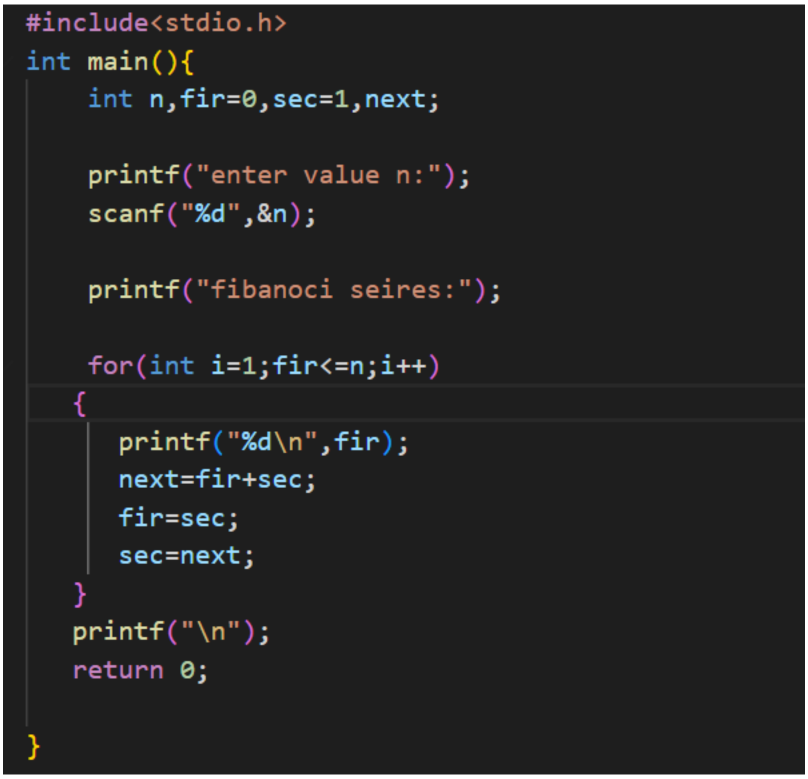


```c
#include<stdio.h>

int main(){
    int num,rev=0,rem;
    printf("enter no:");
    scanf("%d",&num);

    for(;num!=0;num=num/10)
    {
    rem=num%10;
    rev=(rev*10)+rem;
    }
    printf("rverse=%d\n",rev);
    return 0;



}
```

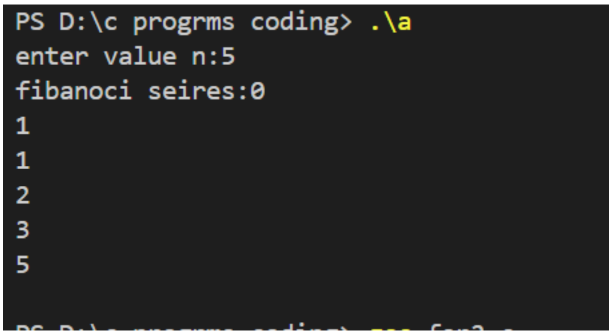


```
PS D:\c progrms coding> gcc for1.c
PS D:\c progrms coding> .\a
enter no:123
rverse=321
PS D:\c progrms coding> 
```

3. Fibannocci series using for loop

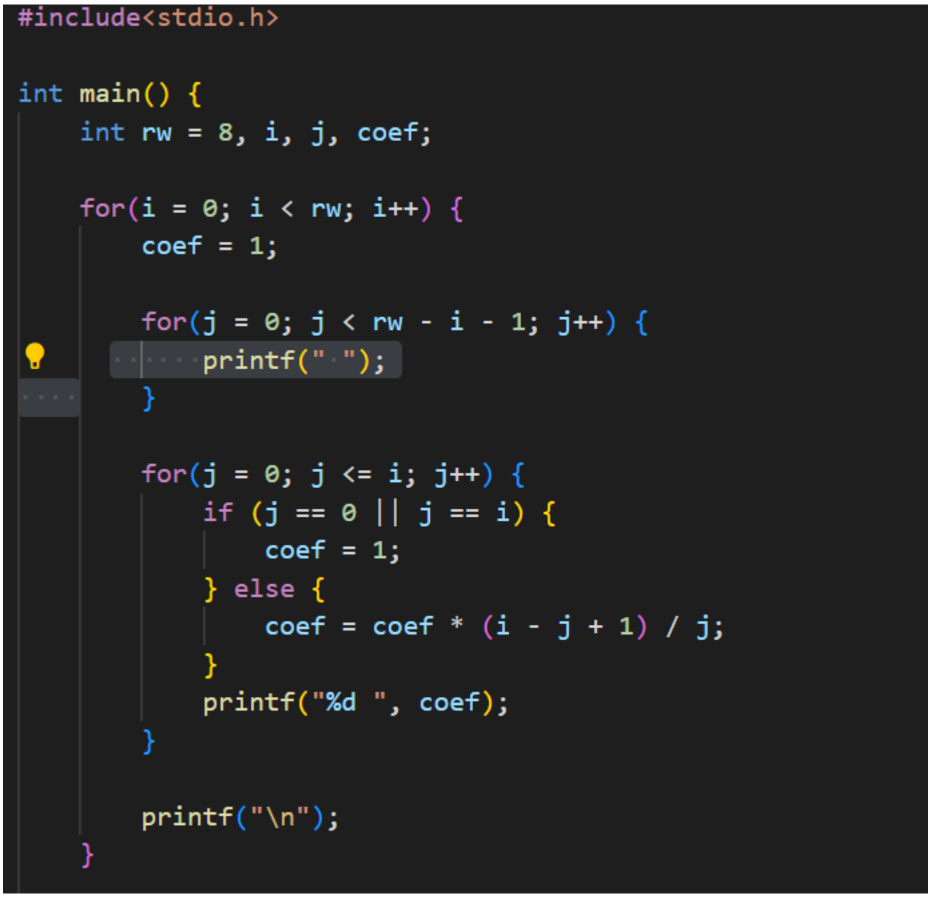


```c
#include<stdio.h>
int main(){
    int n,fir=0,sec=1,next;

    printf("enter value n:");
    scanf("%d",&n);

    printf("fibanoci seires:");

    for(int i=1;fir<=n;i++)
   {
      printf("%d\n",fir);
      next=fir+sec;
      fir=sec;
      sec=next;
   }
   printf("\n");
   return 0;

}
```

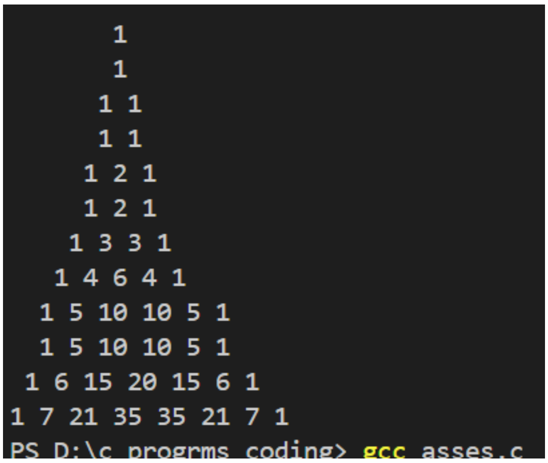


```
PS D:\c progrms coding> .\a
enter value n:5
fibanoci seires:0
1
1
2
3
5

PS D:\c progrms coding> gcc for2.c
```

4.Pascal Triangle using loop



```c
#include<stdio.h>

int main() {
    int rw = 8, i, j, coef;

    for(i = 0; i < rw; i++) {
        coef = 1;

        for(j = 0; j < rw - i - 1; j++) {
            printf(" ");
        }

        for(j = 0; j <= i; j++) {
            if (j == 0 || j == i) {
                coef = 1;
            } else {
                coef = coef * (i - j + 1) / j;
            }
            printf("%d ", coef);
        }

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



```
       1
       1
      1 1
      1 1
     1 2 1
     1 2 1
    1 3 3 1
   1 4 6 4 1
  1 5 10 10 5 1
  1 5 10 10 5 1
 1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
PS D:\c progrms coding> gcc asses.c
```

5. • In this challenge, you are going to create a "Guess the Number C program

•Your program will generate a random number from 0 to 20 •

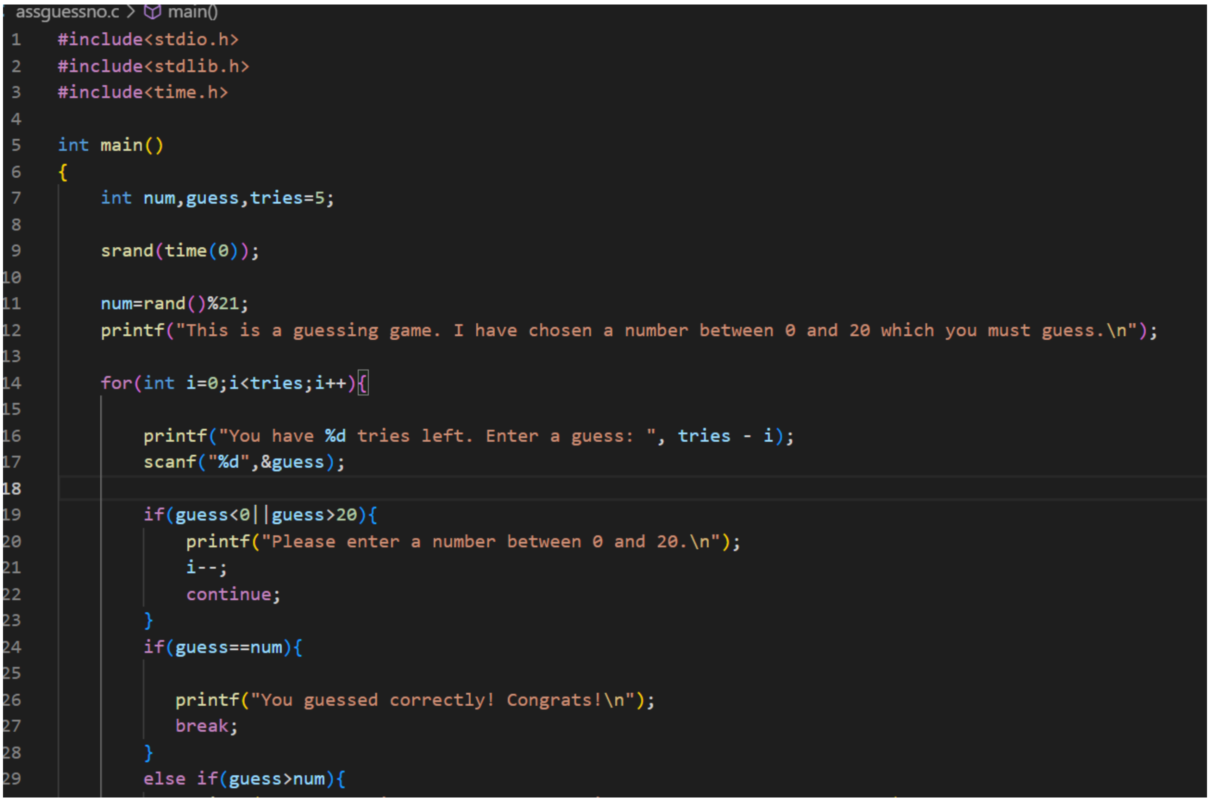
You will then ask the user to guess it •

User should only be able to enter numbers from 0-20

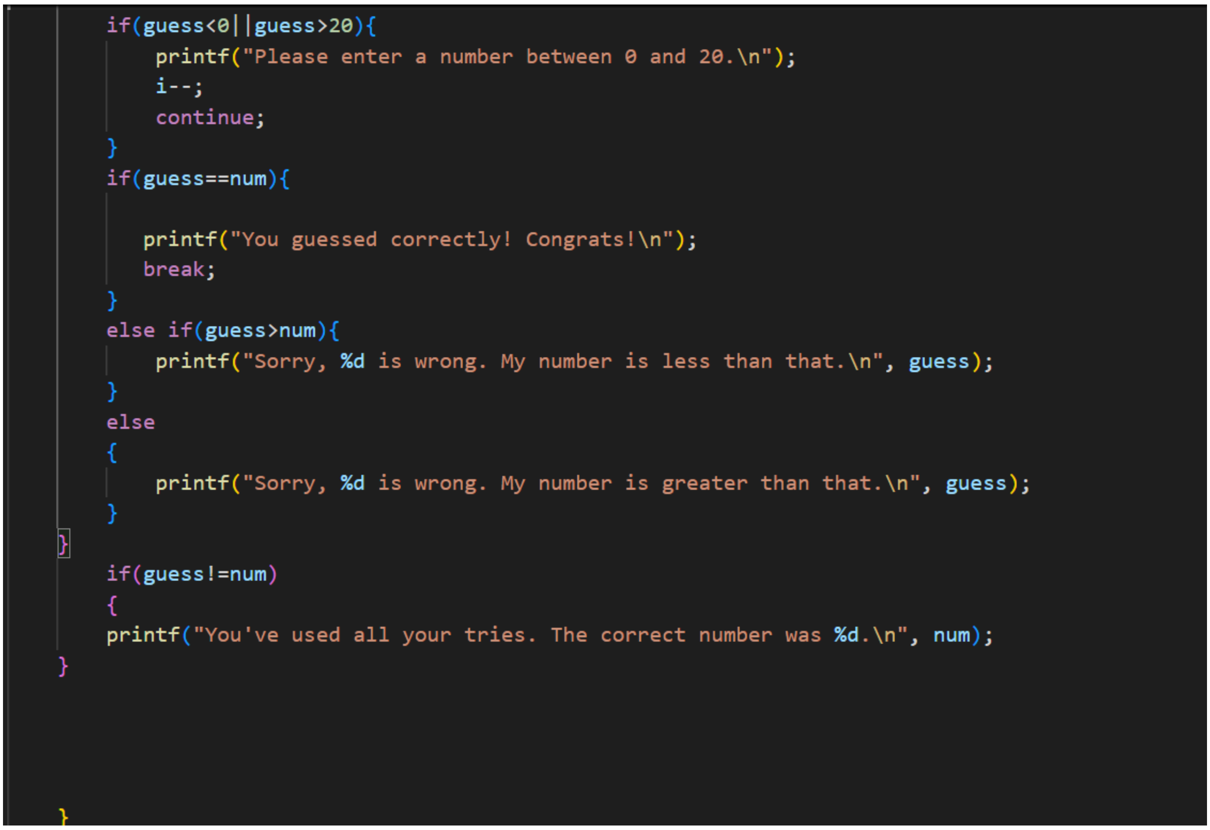
• The program will indicate to the user if each guess is too high or too low

• The player wins the game if they can guess the number within five tries Sample Output This is a guessing game

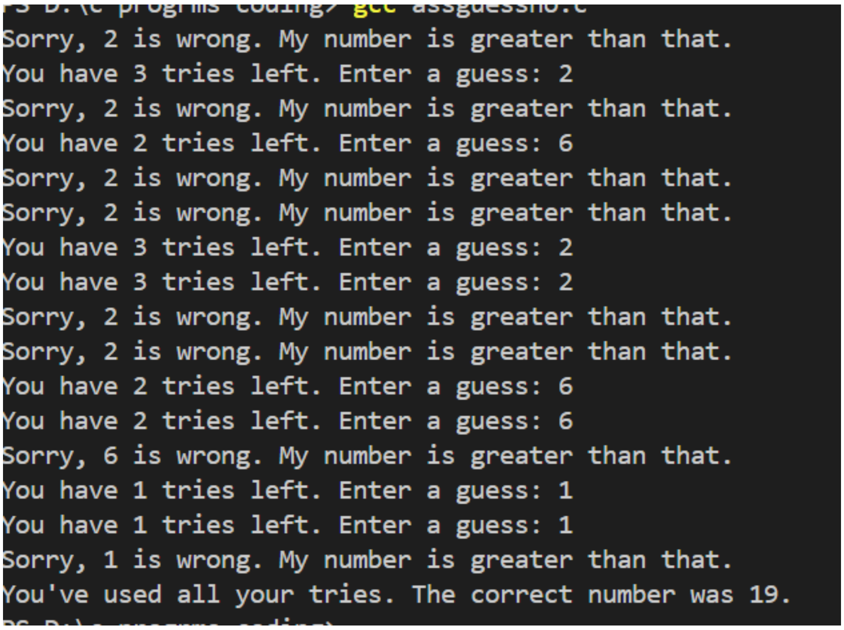
. I have chosen a number between 0 and 20 which you must guess. You have 5 tries left. Enter a guess: 12 Sorry, 12 is wrong. My number is less than that. You have 4 tries left. Enter a guess: 8 Sorry, 8 is wrong. My number is less than that. You have 3 tries left. Enter a guess: 4 Sorry, 4 is wrong. My number is less than that. You have 2 tries left. Enter a guess: 2 you guessed correct! Congrats.



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

int main()
{
    int num,guess,tries=5;

    srand(time(0));

    num=rand()%21;
    printf("This is a guessing game. I have chosen a number between 0 and 20 which you must guess.\n");

    for(int i=0;i<tries;i++){

        printf("You have %d tries left. Enter a guess: ", tries - i);
        scanf("%d",&guess);

        if(guess<0||guess>20){
            printf("Please enter a number between 0 and 20.\n");
            i--;
            continue;
        }
        if(guess==num){

           printf("You guessed correctly! Congrats!\n");
           break;
        }
        else if(guess>num){
```



```
        if(guess<0||guess>20){
            printf("Please enter a number between 0 and 20.\n");
            i--;
            continue;
        }
        if(guess==num){

           printf("You guessed correctly! Congrats!\n");
           break;
        }
        else if(guess>num){
            printf("Sorry, %d is wrong. My number is less than that.\n", guess);
        }
        else
        {
            printf("Sorry, %d is wrong. My number is greater than that.\n", guess);
        }
    }
        if(guess!=num)
        {
        printf("You've used all your tries. The correct number was %d.\n", num);
    }




    }
```



```
PS D:\c programs coding> gcc assguessno.c
Sorry, 2 is wrong. My number is greater than that.
You have 3 tries left. Enter a guess: 2
Sorry, 2 is wrong. My number is greater than that.
You have 2 tries left. Enter a guess: 6
Sorry, 2 is wrong. My number is greater than that.
Sorry, 2 is wrong. My number is greater than that.
You have 3 tries left. Enter a guess: 2
You have 3 tries left. Enter a guess: 2
Sorry, 2 is wrong. My number is greater than that.
Sorry, 2 is wrong. My number is greater than that.
You have 2 tries left. Enter a guess: 6
You have 2 tries left. Enter a guess: 6
Sorry, 6 is wrong. My number is greater than that.
You have 1 tries left. Enter a guess: 1
You have 1 tries left. Enter a guess: 1
Sorry, 1 is wrong. My number is greater than that.
You've used all your tries. The correct number was 19.
PS D:\c programs coding>
```

6. Problem Statement 1: Banking System Simulation

Description: Create a simple banking system simulation that allows users to create an account, deposit money, withdraw money, and check their balance. The program should handle multiple accounts and 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).

2.user inputs(account number, account holder name, balance).

3. Use control statements to navigate through the menu options:

i. Create Account

ii. Deposit Money

iii. Withdraw Money

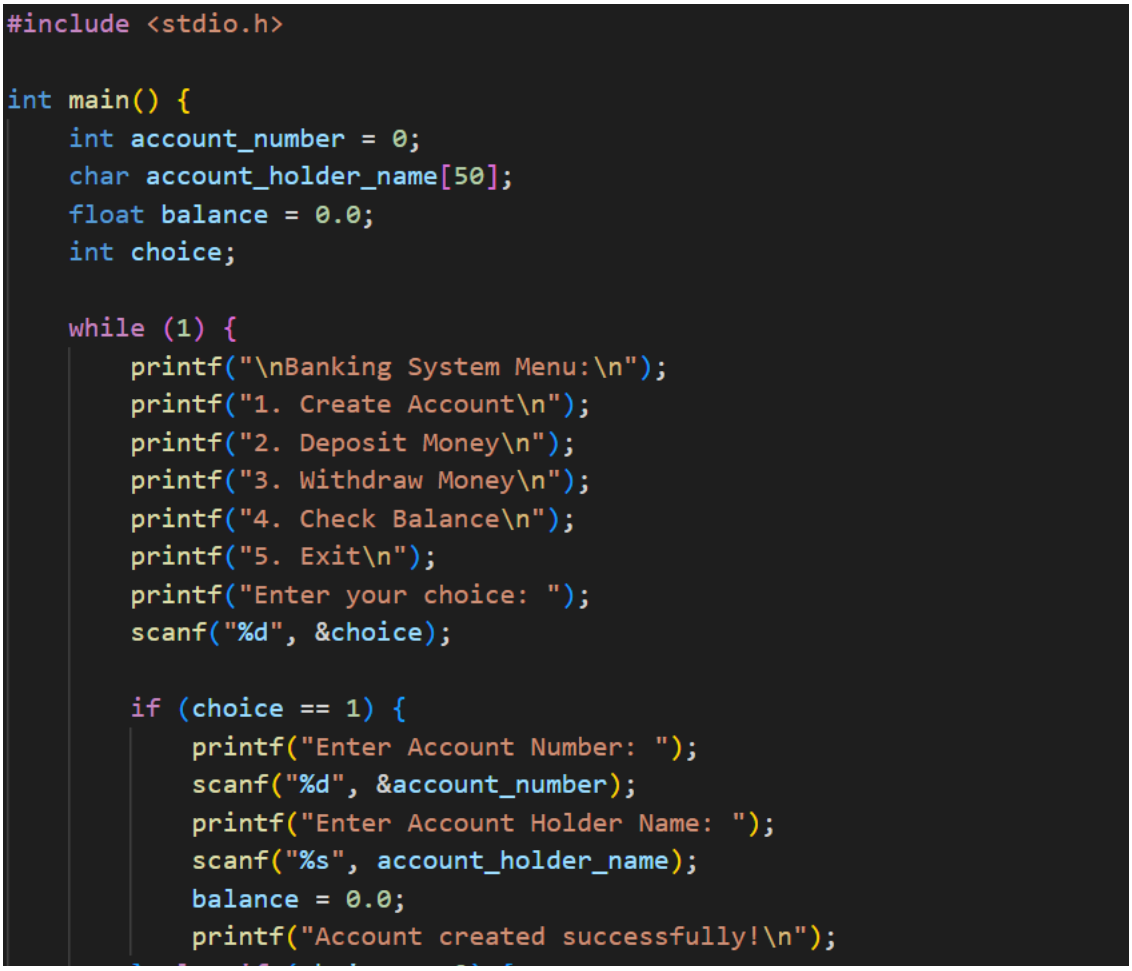
iv. Check Balance

4. Ensure that the withdrawal does not exceed the available balance and handle invalid inputs gracefully.

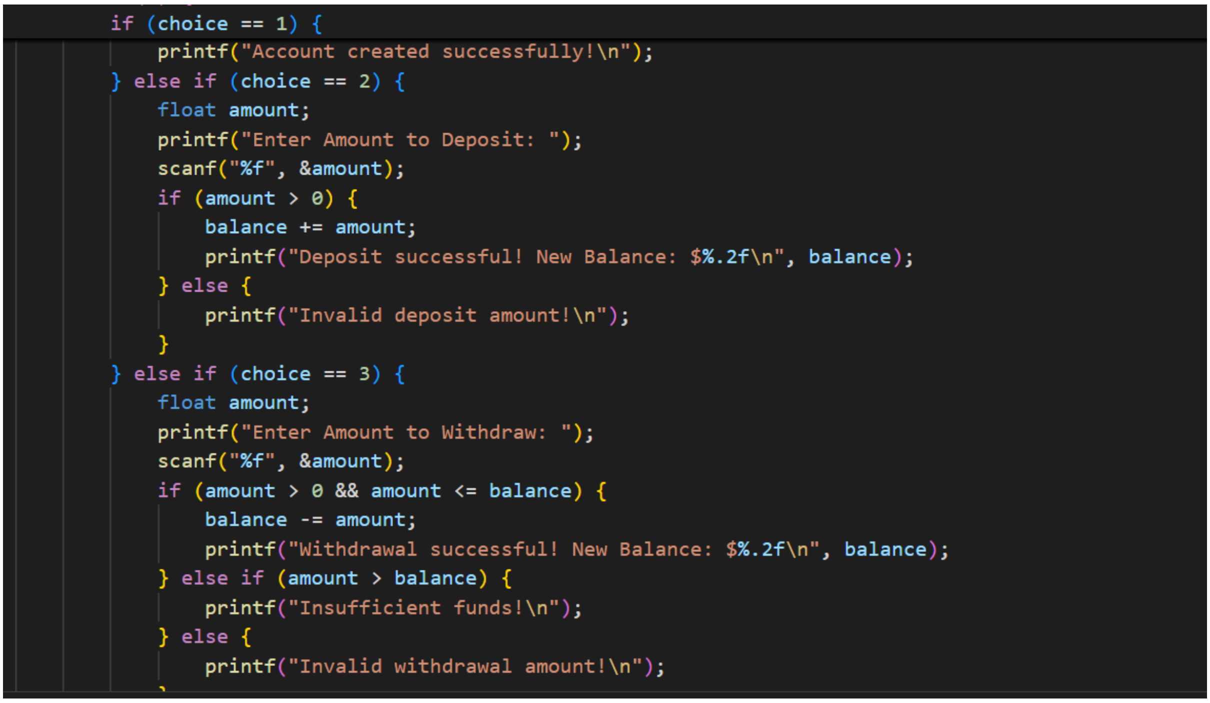
Example Input/Output:

```
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
```

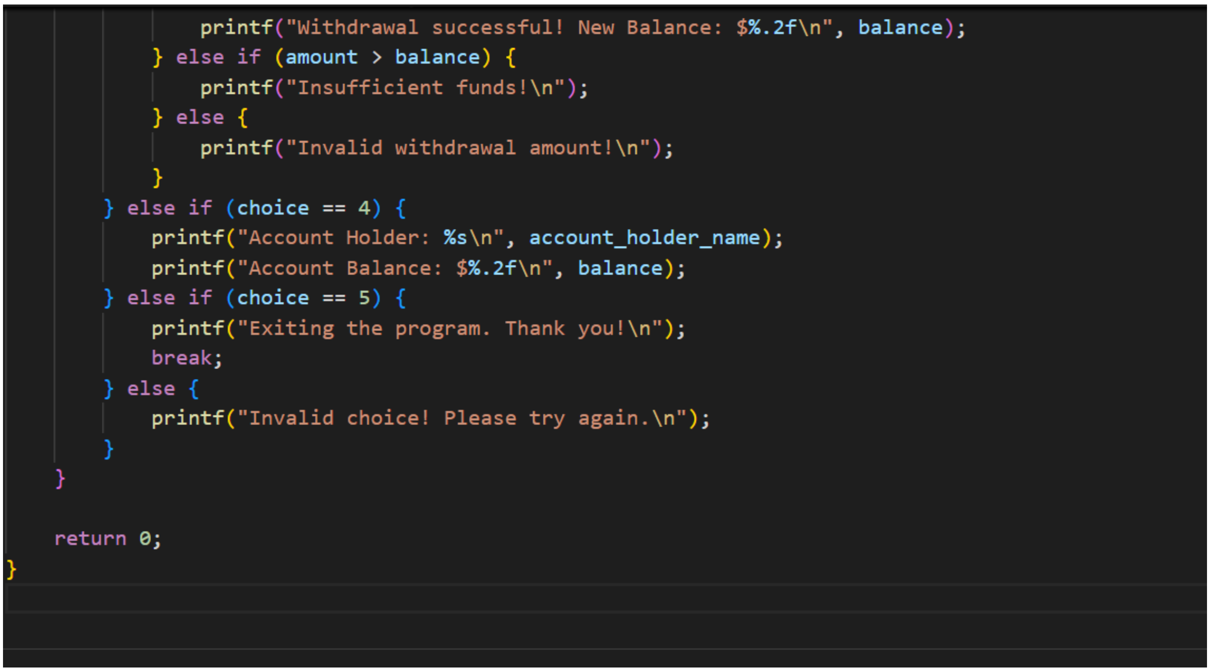
```
Choose an option: 3

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.
```



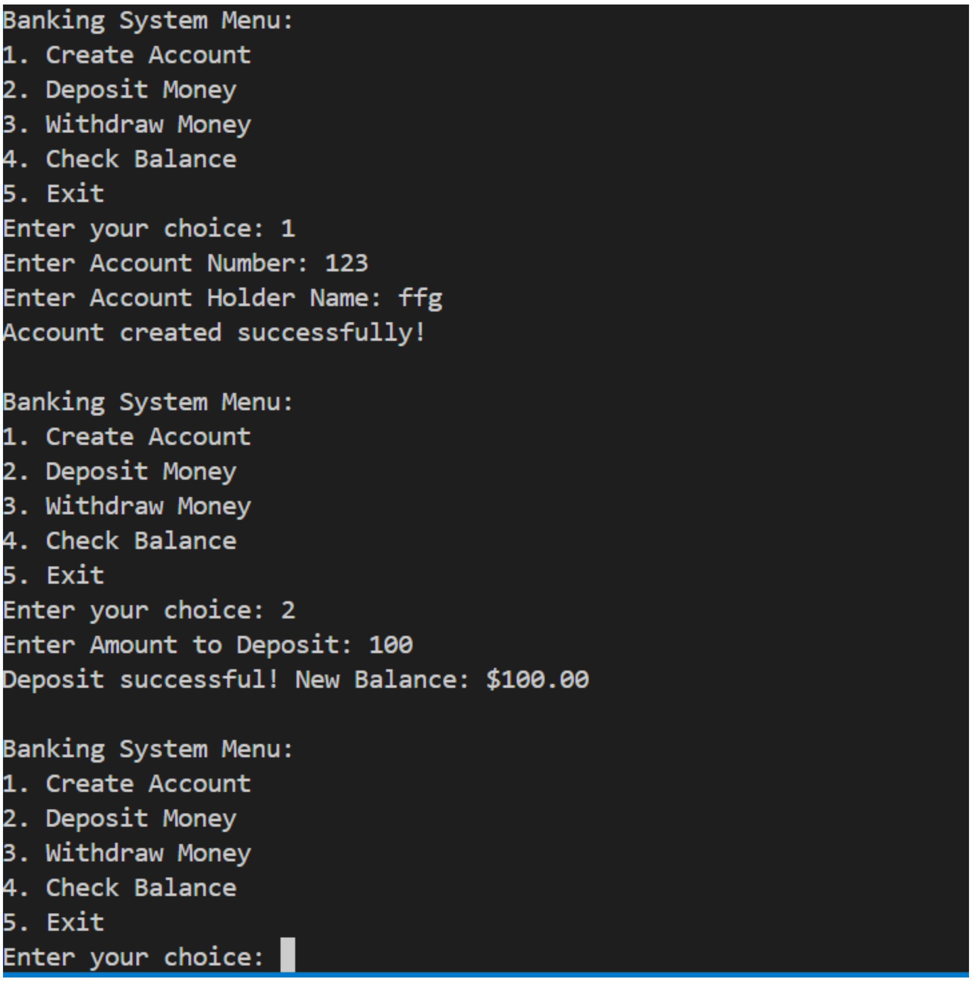
```c
#include <stdio.h>

int main() {
    int account_number = 0;
    char account_holder_name[50];
    float balance = 0.0;
    int choice;

    while (1) {
        printf("\nBanking System Menu:\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");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        if (choice == 1) {
            printf("Enter Account Number: ");
            scanf("%d", &account_number);
            printf("Enter Account Holder Name: ");
            scanf("%s", account_holder_name);
            balance = 0.0;
            printf("Account created successfully!\n");
```



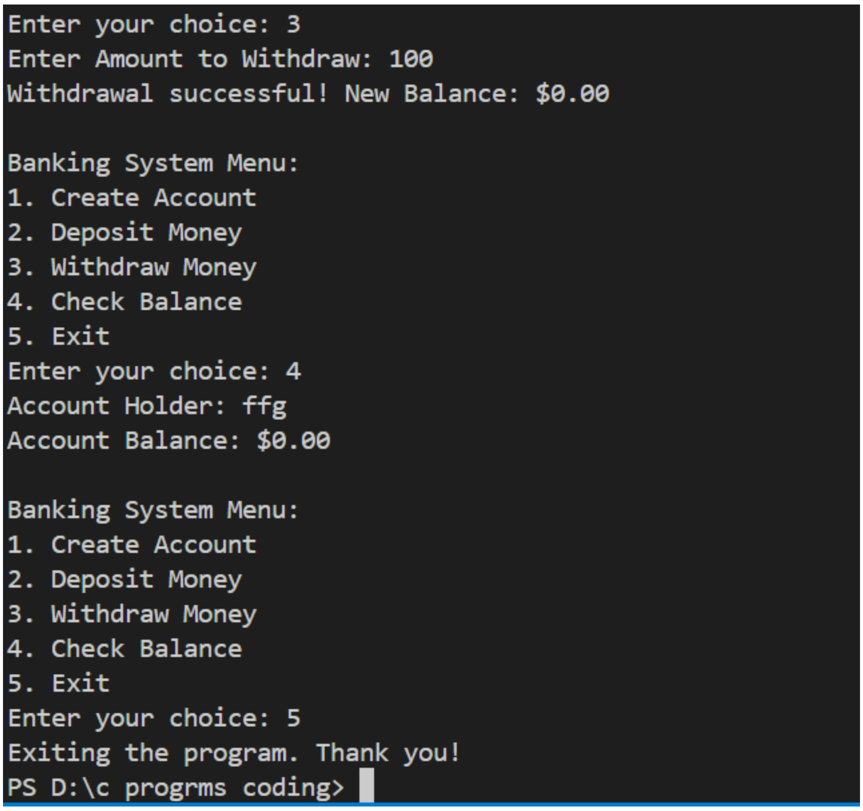
```c
        if (choice == 1) {
            printf("Account created successfully!\n");
        } else if (choice == 2) {
            float amount;
            printf("Enter Amount to Deposit: ");
            scanf("%f", &amount);
            if (amount > 0) {
                balance += amount;
                printf("Deposit successful! New Balance: $%.2f\n", balance);
            } else {
                printf("Invalid deposit amount!\n");
            }
        } else if (choice == 3) {
            float amount;
            printf("Enter Amount to Withdraw: ");
            scanf("%f", &amount);
            if (amount > 0 && amount <= balance) {
                balance -= amount;
                printf("Withdrawal successful! New Balance: $%.2f\n", balance);
            } else if (amount > balance) {
                printf("Insufficient funds!\n");
            } else {
                printf("Invalid withdrawal amount!\n");
```



```c
                printf("Withdrawal successful! New Balance: $%.2f\n", balance);
            } else if (amount > balance) {
                printf("Insufficient funds!\n");
            } else {
                printf("Invalid withdrawal amount!\n");
            }
        } else if (choice == 4) {
            printf("Account Holder: %s\n", account_holder_name);
            printf("Account Balance: $%.2f\n", balance);
        } else if (choice == 5) {
            printf("Exiting the program. Thank you!\n");
            break;
        } else {
            printf("Invalid choice! Please try again.\n");
        }
    }

    return 0;
}
```



```
Banking System Menu:
1. Create Account
2. Deposit Money
3. Withdraw Money
4. Check Balance
5. Exit
Enter your choice: 1
Enter Account Number: 123
Enter Account Holder Name: ffg
Account created successfully!

Banking System Menu:
1. Create Account
2. Deposit Money
3. Withdraw Money
4. Check Balance
5. Exit
Enter your choice: 2
Enter Amount to Deposit: 100
Deposit successful! New Balance: $100.00

Banking System Menu:
1. Create Account
2. Deposit Money
3. Withdraw Money
4. Check Balance
5. Exit
Enter your choice: 
```



```
Enter your choice: 3
Enter Amount to Withdraw: 100
Withdrawal successful! New Balance: $0.00

Banking System Menu:
1. Create Account
2. Deposit Money
3. Withdraw Money
4. Check Balance
5. Exit
Enter your choice: 4
Account Holder: ffg
Account Balance: $0.00

Banking System Menu:
1. Create Account
2. Deposit Money
3. Withdraw Money
4. Check Balance
5. Exit
Enter your choice: 5
Exiting the program. Thank you!
PS D:\c progrms coding> 
```

7. Problem Statement 4: 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.

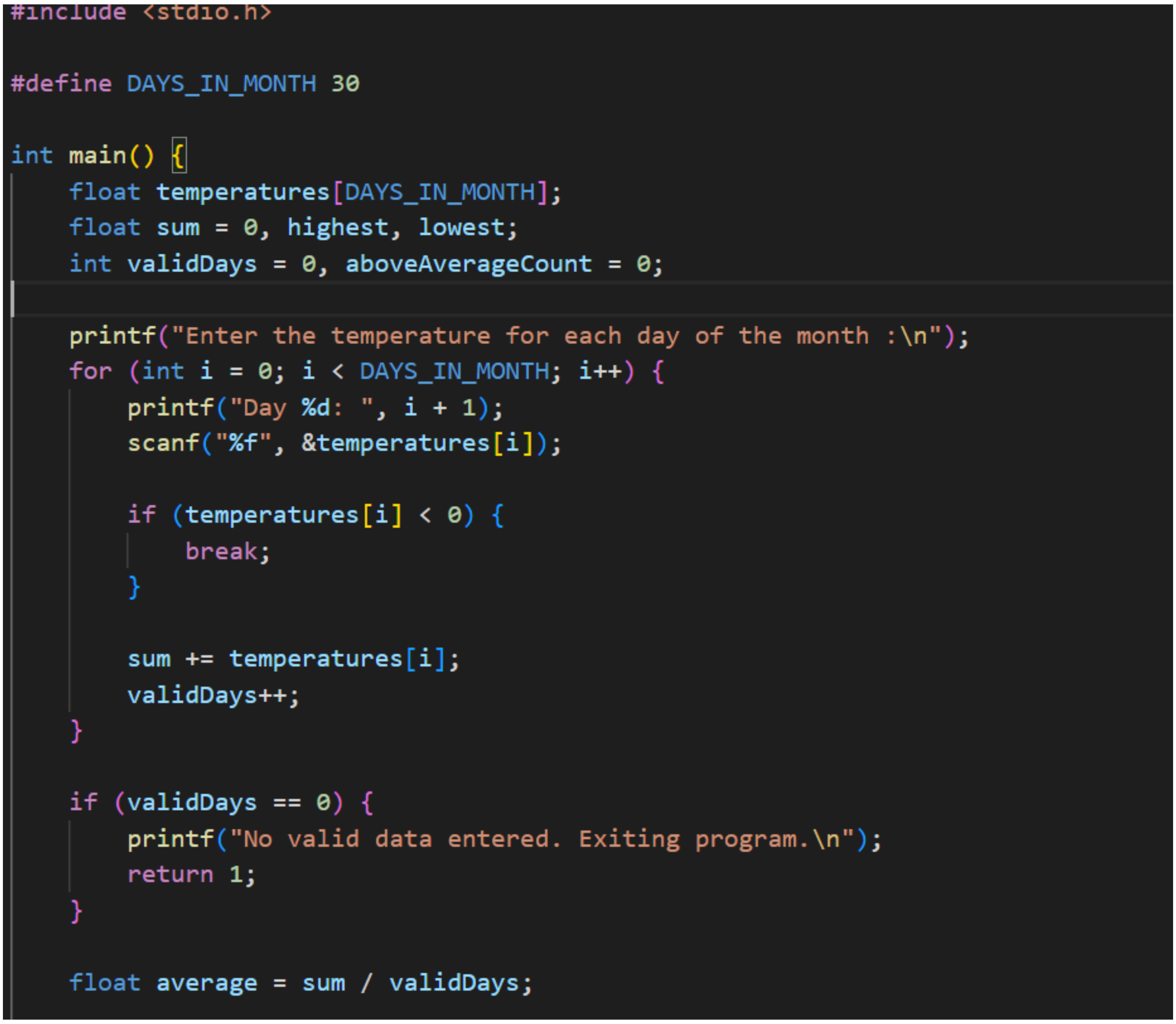
    iii. Lowest Temperature recorded.

    iv. Count of days with temperatures above average.

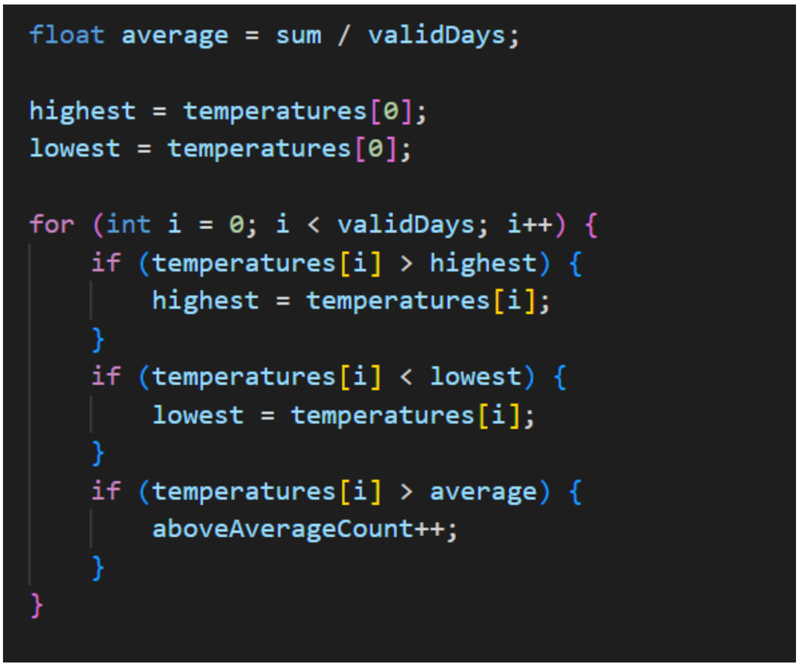
4. Handle cases where no data is entered.

Example Input/Output:

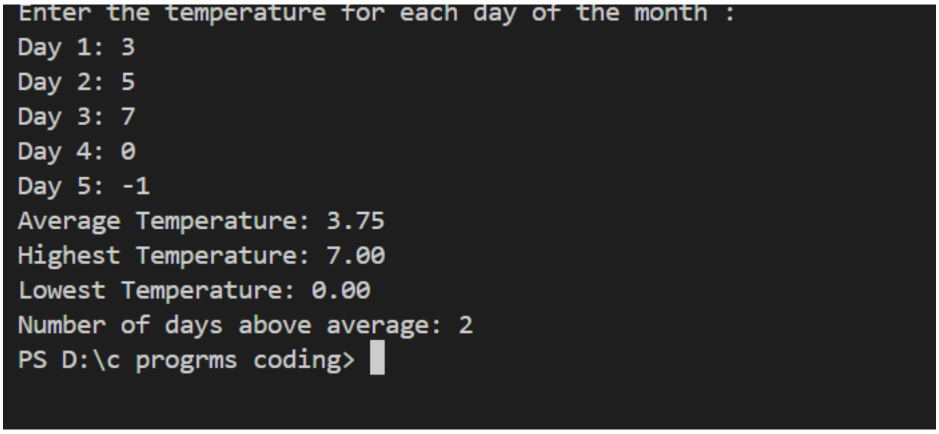
```
Enter temperatures for each day of the month (30 days):

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
```



```c
#include <stdio.h>

#define DAYS_IN_MONTH 30

int main() {
    float temperatures[DAYS_IN_MONTH];
    float sum = 0, highest, lowest;
    int validDays = 0, aboveAverageCount = 0;

    printf("Enter the temperature for each day of the month :\n");
    for (int i = 0; i < DAYS_IN_MONTH; i++) {
        printf("Day %d: ", i + 1);
        scanf("%f", &temperatures[i]);

        if (temperatures[i] < 0) {
            break;
        }

        sum += temperatures[i];
        validDays++;
    }

    if (validDays == 0) {
        printf("No valid data entered. Exiting program.\n");
        return 1;
    }

    float average = sum / validDays;
```



```c
float average = sum / validDays;

highest = temperatures[0];
lowest = temperatures[0];

for (int i = 0; i < validDays; i++) {
    if (temperatures[i] > highest) {
        highest = temperatures[i];
    }
    if (temperatures[i] < lowest) {
        lowest = temperatures[i];
    }
    if (temperatures[i] > average) {
        aboveAverageCount++;
    }
}
```



```
Enter the temperature for each day of the month :
Day 1: 3
Day 2: 5
Day 3: 7
Day 4: 0
Day 5: -1
Average Temperature: 3.75
Highest Temperature: 7.00
Lowest Temperature: 0.00
Number of days above average: 2
PS D:\c progrms coding> 
```

8.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. Implement a structure to hold product information.

3. Use control statements for menu-driven operations:

    i. Add Product

    ii. Update Product Quantity

    iii. Delete Product

    iv. View All Products in Inventory

4. 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 All Products in Inventory

5. Exit

Choose an option: 1

Enter product name: Widget A

Enter product quantity: 50

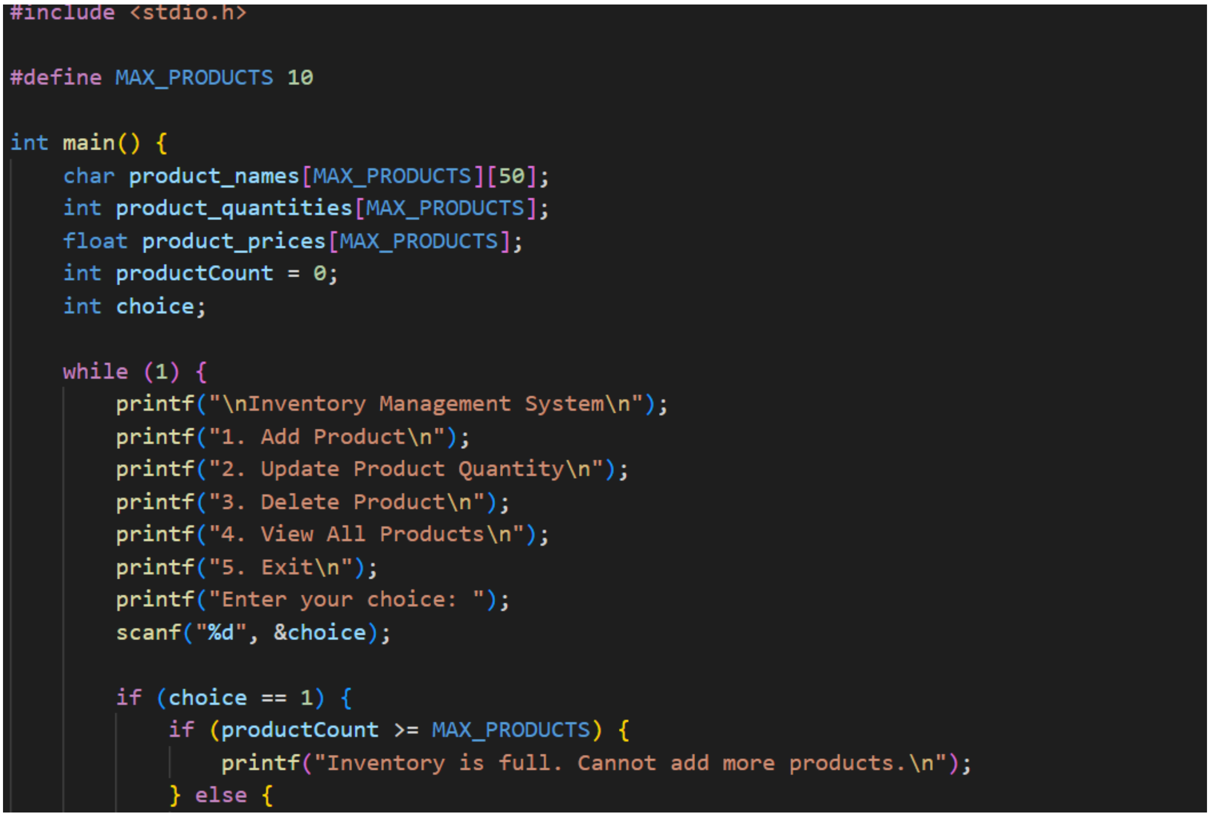
Enter product price: 19.99

Choose an option: 4

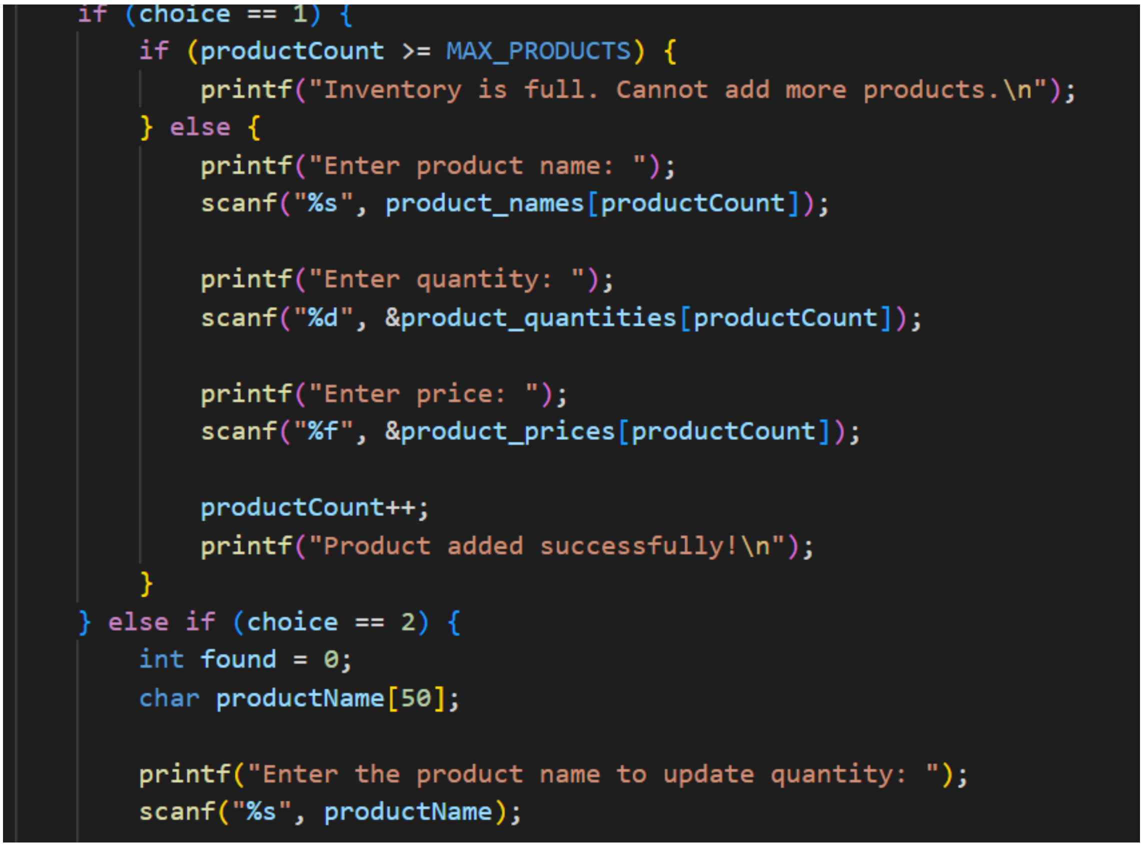
Product Name: Widget A, Quantity: 50, Price: $19.99

Choose an option: 5

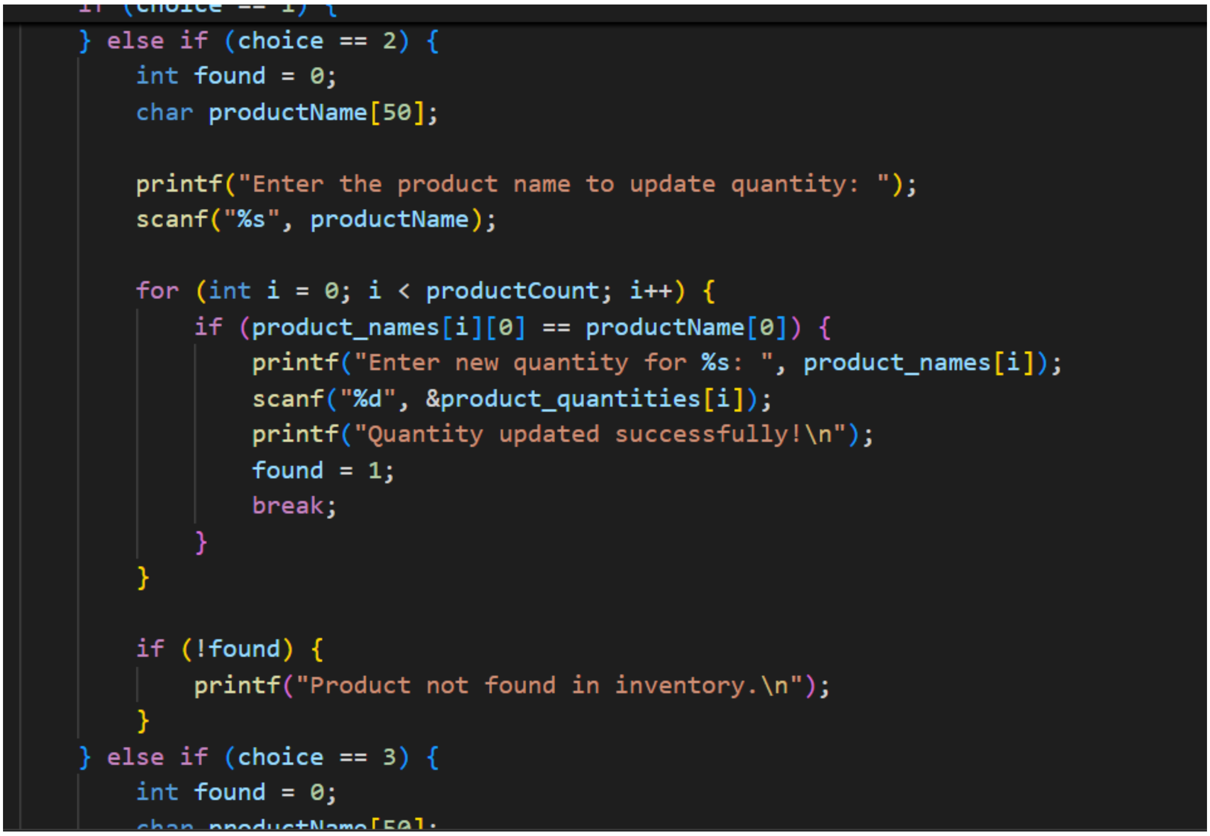
Exiting the system.



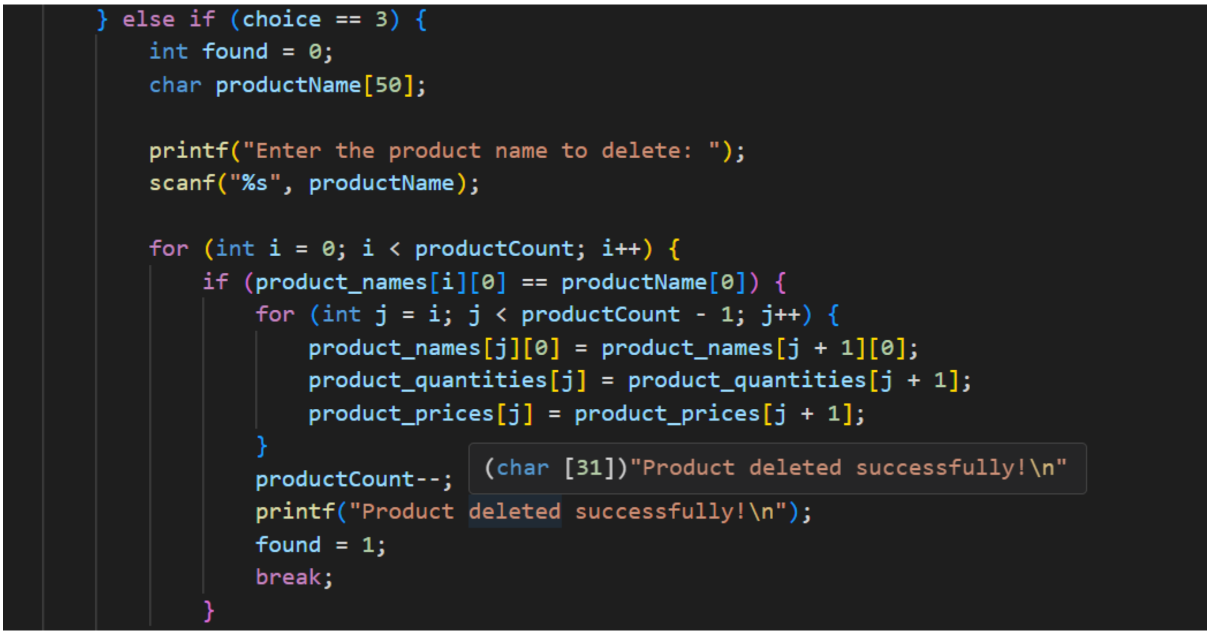
```c
#include <stdio.h>

#define MAX_PRODUCTS 10

int main() {
    char product_names[MAX_PRODUCTS][50];
    int product_quantities[MAX_PRODUCTS];
    float product_prices[MAX_PRODUCTS];
    int productCount = 0;
    int choice;

    while (1) {
        printf("\nInventory Management System\n");
        printf("1. Add Product\n");
        printf("2. Update Product Quantity\n");
        printf("3. Delete Product\n");
        printf("4. View All Products\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        if (choice == 1) {
            if (productCount >= MAX_PRODUCTS) {
                printf("Inventory is full. Cannot add more products.\n");
            } else {
```



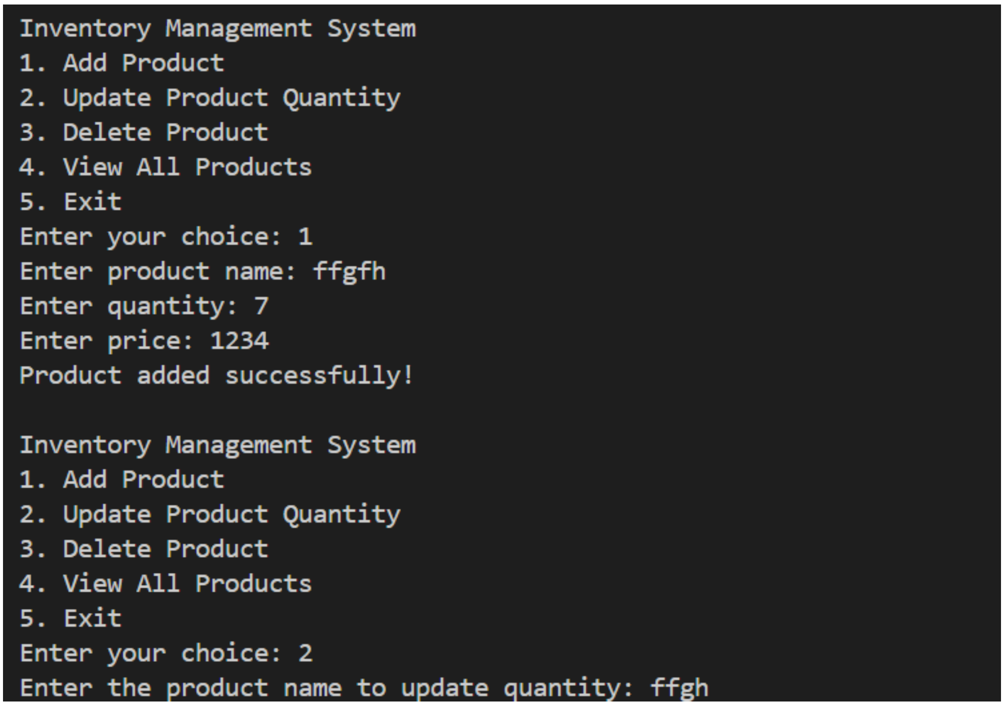
```
if (choice == 1) {
    if (productCount >= MAX_PRODUCTS) {
        printf("Inventory is full. Cannot add more products.\n");
    } else {
        printf("Enter product name: ");
        scanf("%s", product_names[productCount]);

        printf("Enter quantity: ");
        scanf("%d", &product_quantities[productCount]);

        printf("Enter price: ");
        scanf("%f", &product_prices[productCount]);

        productCount++;
        printf("Product added successfully!\n");
    }
} else if (choice == 2) {
    int found = 0;
    char productName[50];

    printf("Enter the product name to update quantity: ");
    scanf("%s", productName);
```



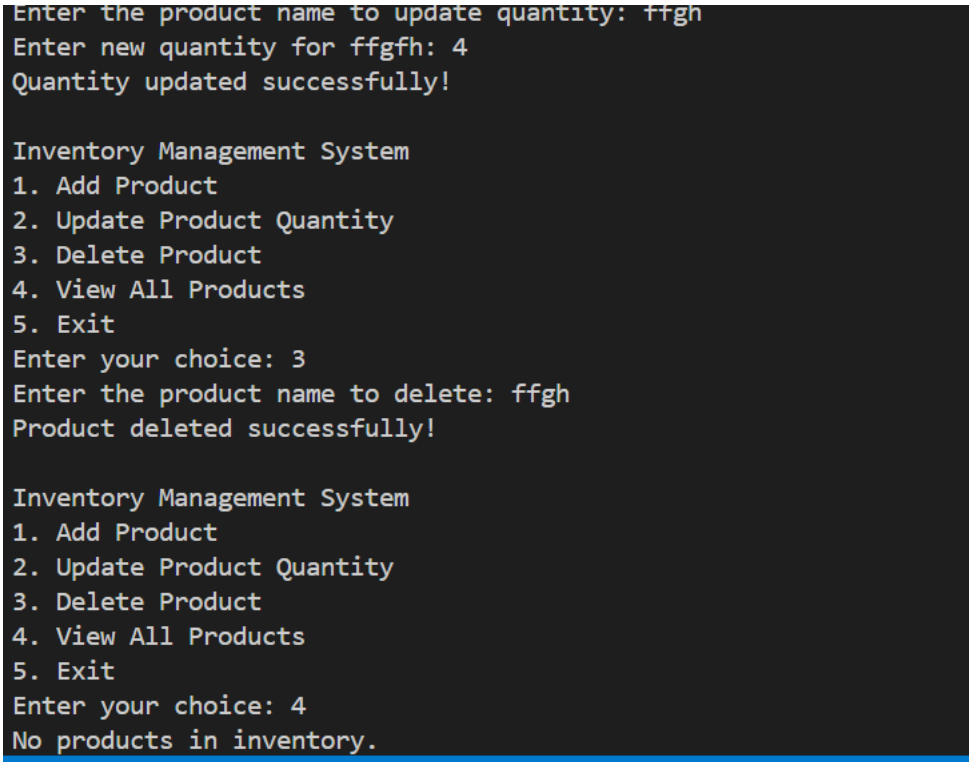
```
if (choice == 1) {
} else if (choice == 2) {
    int found = 0;
    char productName[50];

    printf("Enter the product name to update quantity: ");
    scanf("%s", productName);

    for (int i = 0; i < productCount; i++) {
        if (product_names[i][0] == productName[0]) {
            printf("Enter new quantity for %s: ", product_names[i]);
            scanf("%d", &product_quantities[i]);
            printf("Quantity updated successfully!\n");
            found = 1;
            break;
        }
    }

    if (!found) {
        printf("Product not found in inventory.\n");
    }
} else if (choice == 3) {
    int found = 0;
    char productName[50];
```



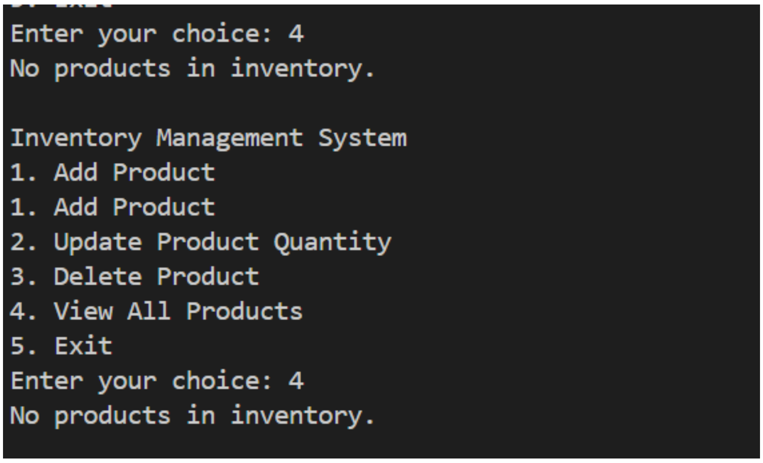
```
} else if (choice == 3) {
    int found = 0;
    char productName[50];

    printf("Enter the product name to delete: ");
    scanf("%s", productName);

    for (int i = 0; i < productCount; i++) {
        if (product_names[i][0] == productName[0]) {
            for (int j = i; j < productCount - 1; j++) {
                product_names[j][0] = product_names[j + 1][0];
                product_quantities[j] = product_quantities[j + 1];
                product_prices[j] = product_prices[j + 1];
            }
            productCount--;  (char [31])"Product deleted successfully!\n"
            printf("Product deleted successfully!\n");
            found = 1;
            break;
        }
```



```
Inventory Management System
1. Add Product
2. Update Product Quantity
3. Delete Product
4. View All Products
5. Exit
Enter your choice: 1
Enter product name: ffgfh
Enter quantity: 7
Enter price: 1234
Product added successfully!

Inventory Management System
1. Add Product
2. Update Product Quantity
3. Delete Product
4. View All Products
5. Exit
Enter your choice: 2
Enter the product name to update quantity: ffgh
```

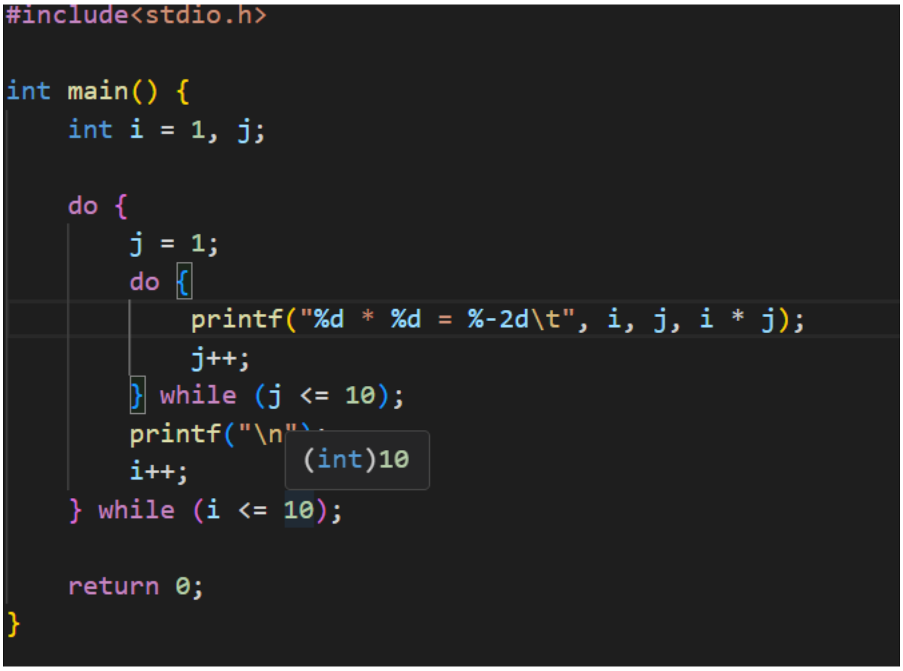


```
Enter the product name to update quantity: ffgh
Enter new quantity for ffgfh: 4
Quantity updated successfully!

Inventory Management System
1. Add Product
2. Update Product Quantity
3. Delete Product
4. View All Products
5. Exit
Enter your choice: 3
Enter the product name to delete: ffgh
Product deleted successfully!

Inventory Management System
1. Add Product
2. Update Product Quantity
3. Delete Product
4. View All Products
5. Exit
Enter your choice: 4
No products in inventory.
```

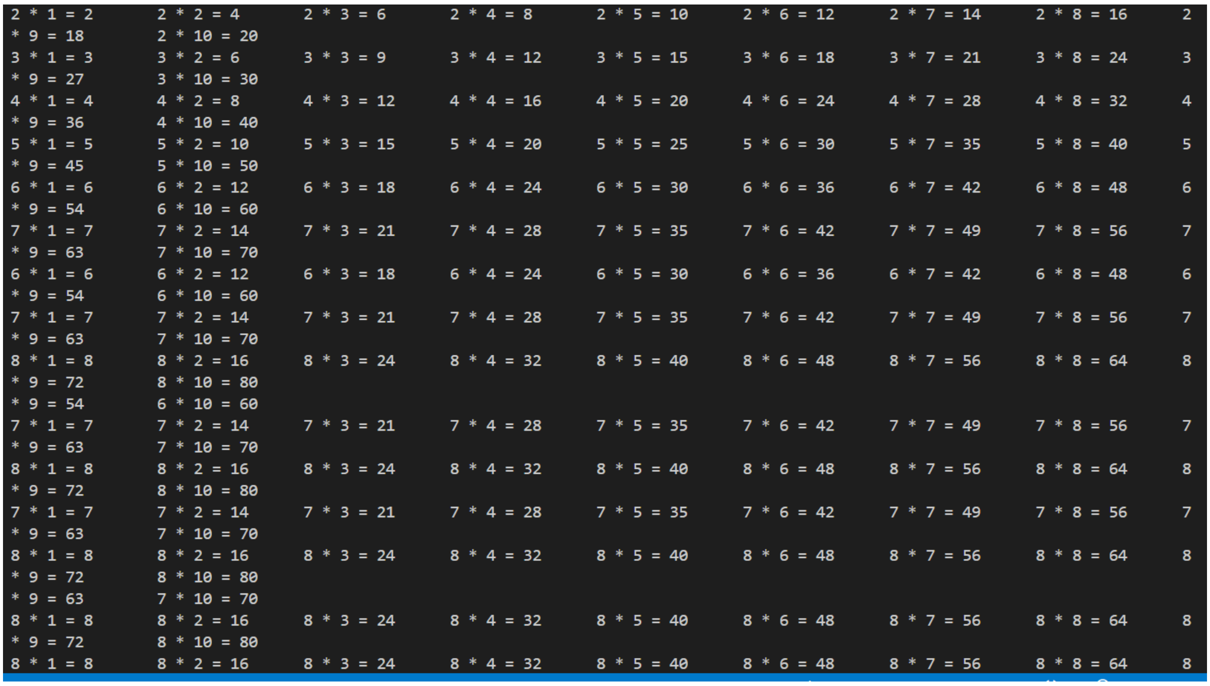


```
5. Exit
Enter your choice: 4
No products in inventory.

Inventory Management System
1. Add Product
1. Add Product
2. Update Product Quantity
3. Delete Product
4. View All Products
5. Exit
Enter your choice: 4
No products in inventory.
```

9.multiplication table using do while

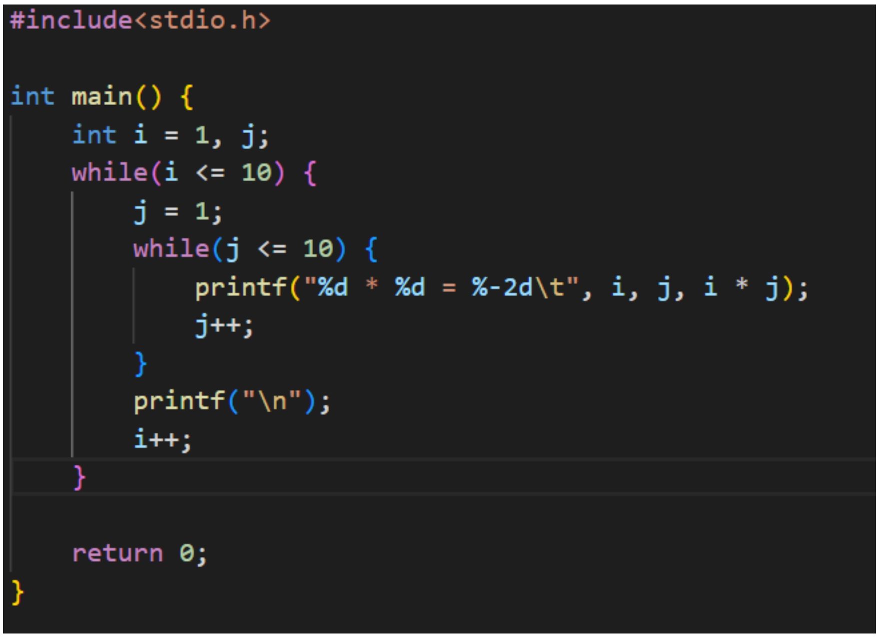


```c
#include<stdio.h>

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

    do {
        j = 1;
        do {
            printf("%d * %d = %-2d\t", i, j, i * j);
            j++;
        } while (j <= 10);
        printf("\n");
        i++;
    } while (i <= 10);

    return 0;
}
```

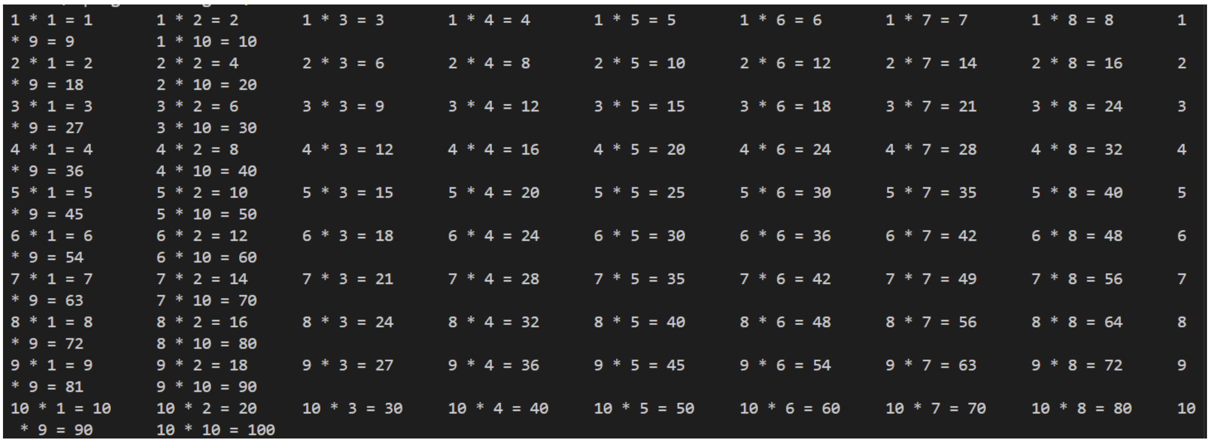


```
2 * 1 = 2       2 * 2 = 4       2 * 3 = 6       2 * 4 = 8       2 * 5 = 10      2 * 6 = 12      2 * 7 = 14      2 * 8 = 16      2
* 9 = 18        2 * 10 = 20
3 * 1 = 3       3 * 2 = 6       3 * 3 = 9       3 * 4 = 12      3 * 5 = 15      3 * 6 = 18      3 * 7 = 21      3 * 8 = 24      3
* 9 = 27        3 * 10 = 30
4 * 1 = 4       4 * 2 = 8       4 * 3 = 12      4 * 4 = 16      4 * 5 = 20      4 * 6 = 24      4 * 7 = 28      4 * 8 = 32      4
* 9 = 36        4 * 10 = 40
5 * 1 = 5       5 * 2 = 10      5 * 3 = 15      5 * 4 = 20      5 * 5 = 25      5 * 6 = 30      5 * 7 = 35      5 * 8 = 40      5
* 9 = 45        5 * 10 = 50
6 * 1 = 6       6 * 2 = 12      6 * 3 = 18      6 * 4 = 24      6 * 5 = 30      6 * 6 = 36      6 * 7 = 42      6 * 8 = 48      6
* 9 = 54        6 * 10 = 60
7 * 1 = 7       7 * 2 = 14      7 * 3 = 21      7 * 4 = 28      7 * 5 = 35      7 * 6 = 42      7 * 7 = 49      7 * 8 = 56      7
* 9 = 63        7 * 10 = 70
6 * 1 = 6       6 * 2 = 12      6 * 3 = 18      6 * 4 = 24      6 * 5 = 30      6 * 6 = 36      6 * 7 = 42      6 * 8 = 48      6
* 9 = 54        6 * 10 = 60
7 * 1 = 7       7 * 2 = 14      7 * 3 = 21      7 * 4 = 28      7 * 5 = 35      7 * 6 = 42      7 * 7 = 49      7 * 8 = 56      7
* 9 = 63        7 * 10 = 70
8 * 1 = 8       8 * 2 = 16      8 * 3 = 24      8 * 4 = 32      8 * 5 = 40      8 * 6 = 48      8 * 7 = 56      8 * 8 = 64      8
* 9 = 72        8 * 10 = 80
* 9 = 54        6 * 10 = 60
7 * 1 = 7       7 * 2 = 14      7 * 3 = 21      7 * 4 = 28      7 * 5 = 35      7 * 6 = 42      7 * 7 = 49      7 * 8 = 56      7
* 9 = 63        7 * 10 = 70
8 * 1 = 8       8 * 2 = 16      8 * 3 = 24      8 * 4 = 32      8 * 5 = 40      8 * 6 = 48      8 * 7 = 56      8 * 8 = 64      8
* 9 = 72        8 * 10 = 80
7 * 1 = 7       7 * 2 = 14      7 * 3 = 21      7 * 4 = 28      7 * 5 = 35      7 * 6 = 42      7 * 7 = 49      7 * 8 = 56      7
* 9 = 63        7 * 10 = 70
8 * 1 = 8       8 * 2 = 16      8 * 3 = 24      8 * 4 = 32      8 * 5 = 40      8 * 6 = 48      8 * 7 = 56      8 * 8 = 64      8
* 9 = 72        8 * 10 = 80
* 9 = 63        7 * 10 = 70
8 * 1 = 8       8 * 2 = 16      8 * 3 = 24      8 * 4 = 32      8 * 5 = 40      8 * 6 = 48      8 * 7 = 56      8 * 8 = 64      8
* 9 = 72        8 * 10 = 80
8 * 1 = 8       8 * 2 = 16      8 * 3 = 24      8 * 4 = 32      8 * 5 = 40      8 * 6 = 48      8 * 7 = 56      8 * 8 = 64      8
```

10. multiplication table using while



```c
#include<stdio.h>

int main() {
    int i = 1, j;
    while(i <= 10) {
        j = 1;
        while(j <= 10) {
            printf("%d * %d = %-2d\t", i, j, i * j);
            j++;
        }
        printf("\n");
        i++;
    }

    return 0;
}
```



```
1 * 1 = 1       1 * 2 = 2       1 * 3 = 3       1 * 4 = 4       1 * 5 = 5       1 * 6 = 6       1 * 7 = 7       1 * 8 = 8       1
* 9 = 9         1 * 10 = 10
2 * 1 = 2       2 * 2 = 4       2 * 3 = 6       2 * 4 = 8       2 * 5 = 10      2 * 6 = 12      2 * 7 = 14      2 * 8 = 16      2
* 9 = 18        2 * 10 = 20
3 * 1 = 3       3 * 2 = 6       3 * 3 = 9       3 * 4 = 12      3 * 5 = 15      3 * 6 = 18      3 * 7 = 21      3 * 8 = 24      3
* 9 = 27        3 * 10 = 30
4 * 1 = 4       4 * 2 = 8       4 * 3 = 12      4 * 4 = 16      4 * 5 = 20      4 * 6 = 24      4 * 7 = 28      4 * 8 = 32      4
* 9 = 36        4 * 10 = 40
5 * 1 = 5       5 * 2 = 10      5 * 3 = 15      5 * 4 = 20      5 * 5 = 25      5 * 6 = 30      5 * 7 = 35      5 * 8 = 40      5
* 9 = 45        5 * 10 = 50
6 * 1 = 6       6 * 2 = 12      6 * 3 = 18      6 * 4 = 24      6 * 5 = 30      6 * 6 = 36      6 * 7 = 42      6 * 8 = 48      6
* 9 = 54        6 * 10 = 60
7 * 1 = 7       7 * 2 = 14      7 * 3 = 21      7 * 4 = 28      7 * 5 = 35      7 * 6 = 42      7 * 7 = 49      7 * 8 = 56      7
* 9 = 63        7 * 10 = 70
8 * 1 = 8       8 * 2 = 16      8 * 3 = 24      8 * 4 = 32      8 * 5 = 40      8 * 6 = 48      8 * 7 = 56      8 * 8 = 64      8
* 9 = 72        8 * 10 = 80
9 * 1 = 9       9 * 2 = 18      9 * 3 = 27      9 * 4 = 36      9 * 5 = 45      9 * 6 = 54      9 * 7 = 63      9 * 8 = 72      9
* 9 = 81        9 * 10 = 90
10 * 1 = 10     10 * 2 = 20     10 * 3 = 30     10 * 4 = 40     10 * 5 = 50     10 * 6 = 60     10 * 7 = 70     10 * 8 = 80     10
 * 9 = 90       10 * 10 = 100
```