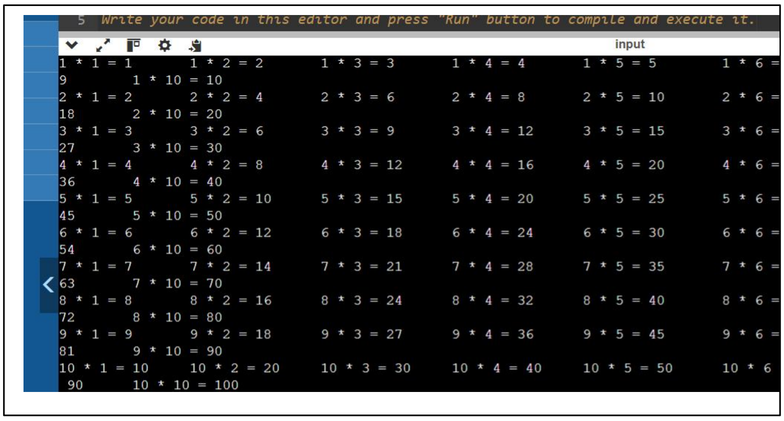
ASSIGNMENT

Nested While Loop

Write a program to print the multiplication tables from 1 to 10

```
#include <stdio.h>
int main()
{
   int a=1,b;
   while(a<=10)
   {
      b=1;
      while(b<=10){
         printf("%d * %d = %d\t",a,b,a*b);
         b++;
      }
      printf("\n");
      a++;

   }
   return 0;
}
```



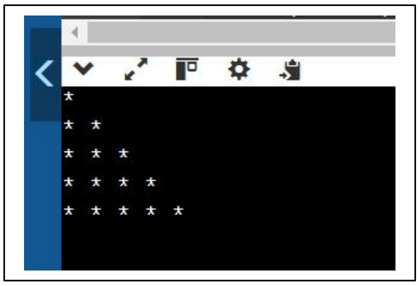
```
Write your code in this editor and press "Run" button to compile and execute it.
```

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

Print the pattern

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

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

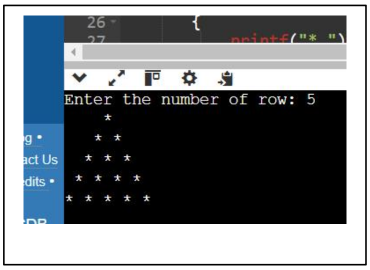


```
*
*  *
*  *  *
*  *  *  *
*  *  *  *  *
```

Print the pattern

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

```
#include <stdio.h>
int main()
{
   int n, i=1, j=1, k=1;
   printf("Enter the number of row: ");
   scanf("%d", &n);
   while(i <= n)
   {
      j=1, k=1;
      while(j <= n-i)
      {
         printf(" ");
         j++;
```



```
{
    printf("* ")
```

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

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

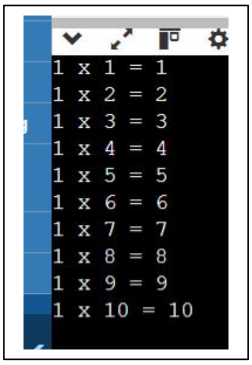
Do while

Write a program to print the number between 1 to 10 using do while

```
#include<stdio.h>
int main(){
    int i=1;
    do{
        printf("%d \n",i);
        i++;
    }while(i<=10);
    return 0;
}
```

Write a program to print multiplication table from 1 to 10

```
#include <stdio.h>
int main() {
    int i = 1, j;

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

    return 0;
}
```

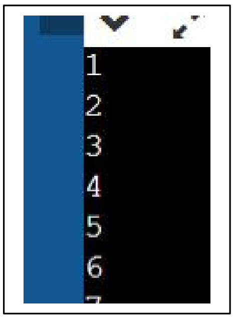


```
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9
1 x 10 = 10
```

For Loop

Write a program to print the numbers between 1 to 10.

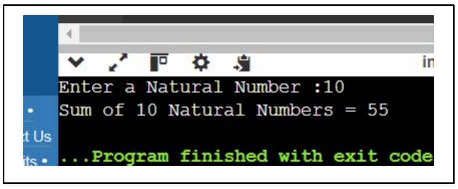
```
#include <stdio.h>
int main()
{
    int i;
    for(i=1;i<=10;i++){
        printf("%d\n",i);
    }
    return 0;
}
```



```
1
2
3
4
5
6
```

Write a program to get sum of n natural numbers

```
#include <stdio.h>
int main()
{
    int n,sum=0;
    printf("Enter a Natural Number :");
```

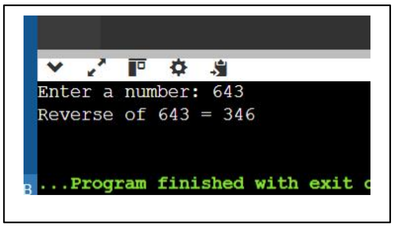


```
Enter a Natural Number :10
Sum of 10 Natural Numbers = 55

...Program finished with exit code
```

```
    scanf("%d",&n);
    for(int i=0;i<=n;i++){
        sum+=i;
    }
    printf("Sum of %d Natural Numbers = %d",n,sum);

    return 0;
}
```

Write a program to reverse a number

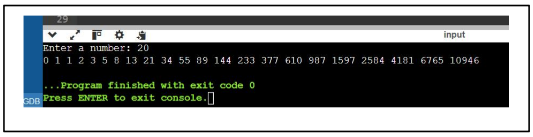
```
#include <stdio.h>
int main() {
    int num,reversed_num = 0,original_num;

    printf("Enter a number: ");
    scanf("%d", &num);
    original_num=num;
    for (num; num != 0; num /= 10) {
        reversed_num = reversed_num * 10 + num % 10;
    }

    printf("Reverse of %d = %d\n", original_num,reversed_num);
    return 0;
}
```



```
Enter a number: 643
Reverse of 643 = 346

...Program finished with exit c
```

Print Fibanocci Series of n natural numbers

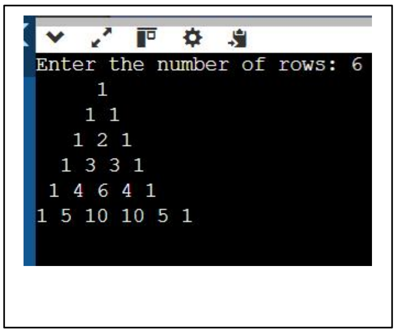
```
#include <stdio.h>

int main() {
    int i, num;
    int previous = 0, next = 1, next_term;

    printf("Enter a number: ");
    scanf("%d", &num);

    printf("%d %d ", previous, next);

    for (i = 1; i <= num; i++) {
        next_term = previous + next;
        printf("%d ", next_term);
        previous = next;
        next = next_term;
    }

    return 0;
}
```



```
Enter a number: 20
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946

...Program finished with exit code 0
Press ENTER to exit console.
```

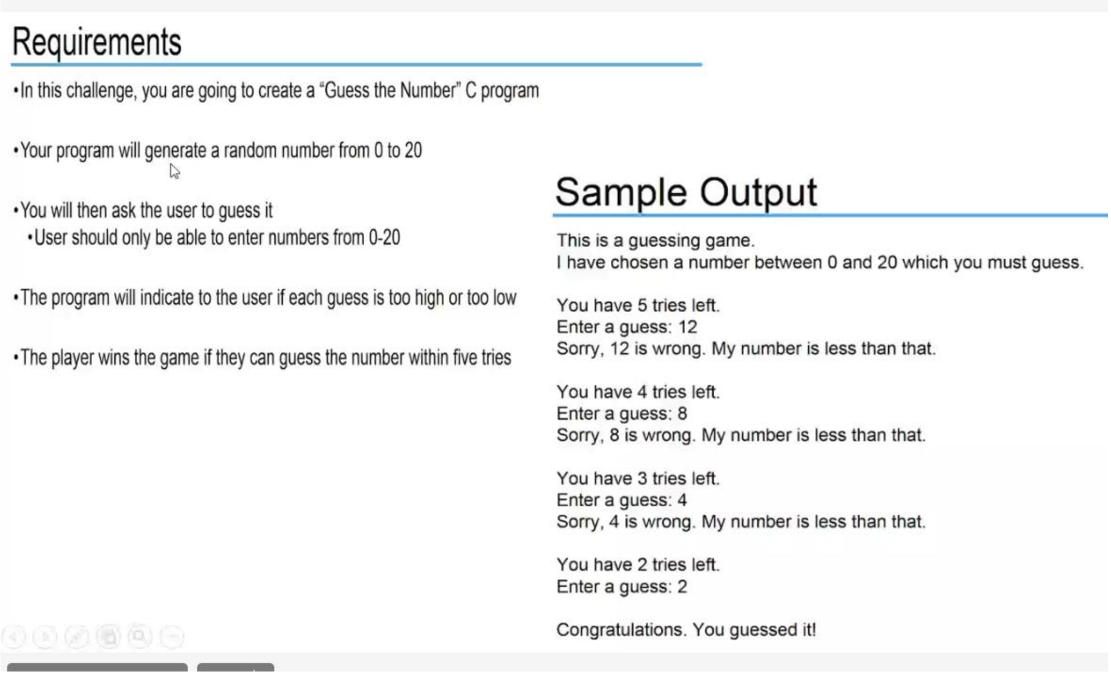
Pascal Triangle

```
#include <stdio.h>
int main() {
    int n;
    printf("Enter the number of rows: ");
    scanf("%d", &n);
    for(int i = 1; i <= n; i++) {
        for(int j = 1; j <= n - i; j++) {
            printf(" ");
        }

        int a = 1;
        for(int k = 1; k <= i; k++) {
            printf("%d ", a);
            a = a * (i - k) / k;
```



```
Enter the number of rows: 6
     1
    1 1
   1 2 1
  1 3 3 1
 1 4 6 4 1
1 5 10 10 5 1
```

```
        }

        printf("\n");
    }

    return 0;
}
```



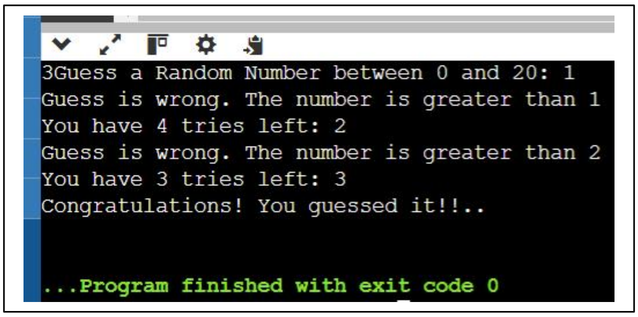
Requirements

- 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

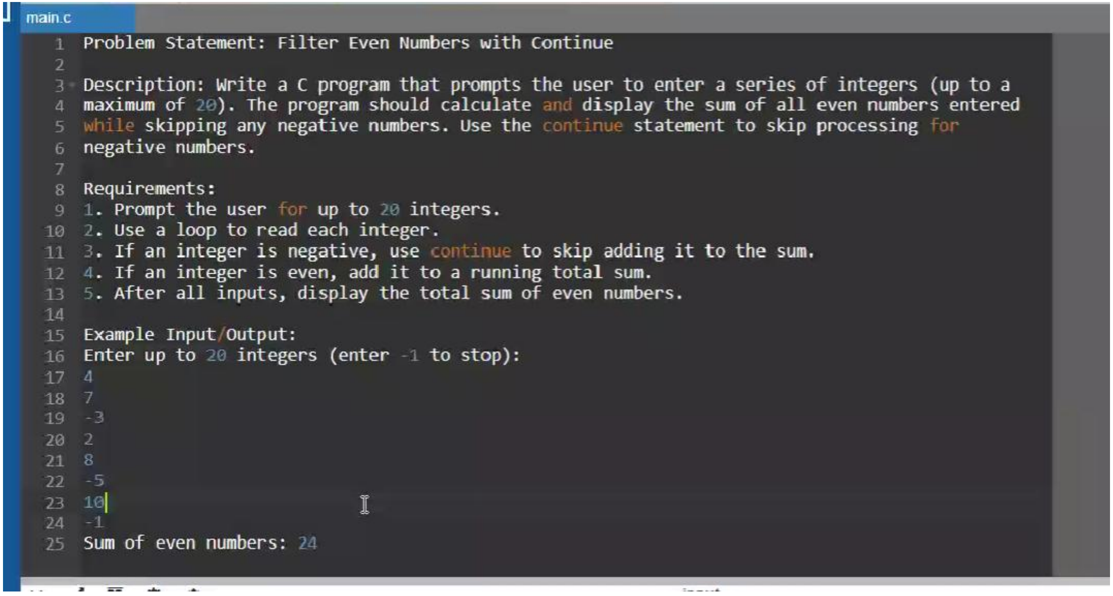
```
This is a guessing game.
I have chosen a number between 0 and 20 which you must guess.

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

Congratulations. You guessed it!
```

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

int main() {
   int num, count = 0,chances=5;

   // Seed the random number generator
   srand(time(0));

   // Generate a random number between 0 and 20
   int random_number = rand() % 21;
   // printf("%d",random_number);

   printf("Guess a Random Number between 0 and 20: ");


   while (count < chances) {
      scanf("%d", &num);

      if (random_number > num) {
         printf("Guess is wrong. The number is greater than %d\n", num);
      } else if (random_number < num) {
         printf("Guess is wrong. The number is less than %d\n", num);
      } else {
         printf("Congratulations! You guessed it!!..\n");
         break;
      }

      count++;

      if (count < chances) {
         printf("You have %d tries left: ",chances-count);
      }
```



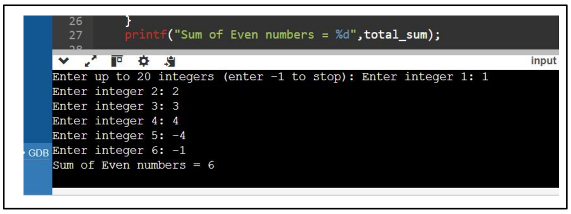
```
3Guess a Random Number between 0 and 20: 1
Guess is wrong. The number is greater than 1
You have 4 tries left: 2
Guess is wrong. The number is greater than 2
You have 3 tries left: 3
Congratulations! You guessed it!!..


...Program finished with exit code 0
```

```
    }

    if (count == chances && num != random_number) {
        printf("Sorry, you've used all your guesses. The correct number was %d\n", random_number);
    }

    return 0;
}
```



```
Problem Statement: Filter Even Numbers with Continue

Description: Write a C program that prompts the user to enter a series of integers (up to a
maximum of 20). The program should calculate and display the sum of all even numbers entered
while skipping any negative numbers. Use the continue statement to skip processing for
negative numbers.

Requirements:
1. Prompt the user for up to 20 integers.
2. Use a loop to read each integer.
3. If an integer is negative, use continue to skip adding it to the sum.
4. If an integer is even, add it to a running total sum.
5. After all inputs, display the total sum of even numbers.

Example Input/Output:
Enter up to 20 integers (enter -1 to stop):
4
7
-3
2
8
-5
10
-1
Sum of even numbers: 24
```

```
#include <stdio.h>

int main() {
    int numbers[20];
    int count = 0;
    int input,total_sum;

    printf("Enter up to 20 integers (enter -1 to stop): ");

    for (int i = 0; i < 20; i++) {
        printf("Enter integer %d: ",i+1);
        scanf("%d", &input);

        if (input == -1) {
            break;
        }
        if (input<0){
            continue;
        }

        numbers[count++] = input;
        if (input%2==0){
            total_sum+=input;
        }
    }
    printf("Sum of Even numbers = %d",total_sum);


}
```



```
        }
        printf("Sum of Even numbers = %d",total_sum);
```

```
Enter up to 20 integers (enter -1 to stop): Enter integer 1: 1
Enter integer 2: 2
Enter integer 3: 3
Enter integer 4: 4
Enter integer 5: -4
Enter integer 6: -1
Sum of Even numbers = 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. Implement a structure to hold account details (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

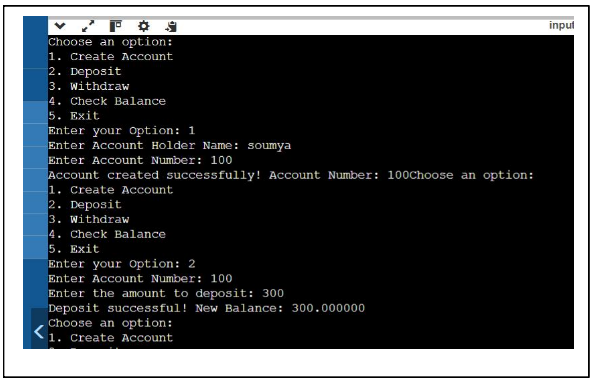
Choose an option: 2
Enter account number: 1001
Enter amount to deposit: 500
Deposit successful! New Balance: 500.0

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.

```
#include <stdio.h>

int main() {
   char name[50];
   int account_number, option;
   float deposit, withdraw, balance = 0;

   while (1) {
      printf("Choose an option:\n");
      printf("1. Create Account\n");
      printf("2. Deposit\n");
      printf("3. Withdraw\n");
      printf("4. Check Balance\n");
      printf("5. Exit\n");
      printf("Enter your Option: ");
      scanf("%d", &option);

      switch (option) {
         case 1:
            printf("Enter Account Holder Name: ");
            scanf("%s", name);  //
            printf("Enter Account Number: ");
            scanf("%d", &account_number);
            printf("Account created successfully! Account Number: %d", account_number);
            break;

         case 2:
            printf("Enter Account Number: ");
            scanf("%d", &account_number);
            printf("Enter the amount to deposit: ");
            scanf("%f", &deposit);
            balance += deposit;
            printf("Deposit successful! New Balance: %f\n", balance);
            break;
```



```
Choose an option:
1. Create Account
2. Deposit
3. Withdraw
4. Check Balance
5. Exit
Enter your Option: 1
Enter Account Holder Name: soumya
Enter Account Number: 100
Account created successfully! Account Number: 100Choose an option:
1. Create Account
2. Deposit
3. Withdraw
4. Check Balance
5. Exit
Enter your Option: 2
Enter Account Number: 100
Enter the amount to deposit: 300
Deposit successful! New Balance: 300.000000
Choose an option:
1. Create Account
```

```
            case 3:
                printf("Enter Account Number: ");
                scanf("%d", &account_number);
                printf("Enter the amount to withdraw: ");
                scanf("%f", &withdraw);

                if (withdraw > balance) {
                    printf("Withdrawal amount cannot exceed balance.\n");
                } else {
                    balance -= withdraw;
                    printf("Withdrawal successful! New Balance: %f\n", balance);
                }
                break;

            case 4:
                printf("Enter Account Number: ");
                scanf("%d", &account_number);
                printf("Current Balance: %f\n", balance);
                break;

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

            default:
                printf("Invalid option. Please try again.\n");
        }
    }

    return 0;
}
```

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.

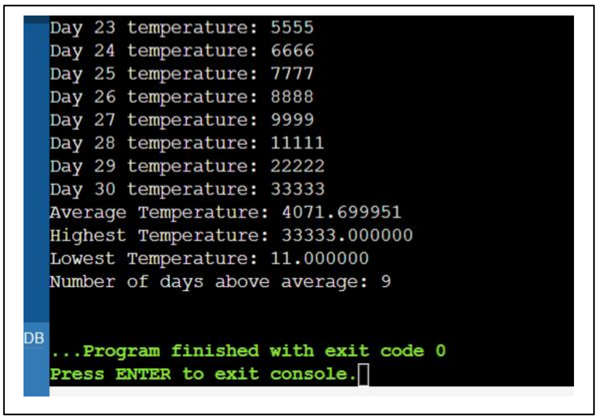
Requirements:
1. Use appropriate data types (float for temperatures and int for days).
2. Store temperature data in an array.
3. Use control statements to calculate:
  i. Average Temperature of the month.
  ii. Highest Temperature recorded.
  iii. Lowest Temperature recorded.
  iv. Count of days with 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
```

```
#include <stdio.h>

int main() {

    int days=30,days_above_average=0,i;
```

```c
    float temperatures[days];
    float sum = 0.0, average;
    printf("Enter temperatures for each day of the month (%d days):\n", days);
    for (i = 0; i < days; i++) {
        printf("Day %d temperature: ", i+1);
        scanf("%f", &temperatures[i]);
        sum += temperatures[i];
    }
    average = sum / days;
    printf("Average Temperature: %f\n", average);
        float highest,lowest;
    highest = temperatures[0];
    lowest = temperatures[0];
    for (int i = 0; i < days; i++) {
        if (temperatures[i] > highest) {
            highest = temperatures[i];
        }
        if (temperatures[i] < lowest) {
            lowest = temperatures[i];
        }
        if (temperatures[i] > average) {
            days_above_average++;
        }
    }
    printf("Highest Temperature: %f\n", highest);
    printf("Lowest Temperature: %f\n", lowest);
    printf("Number of days above average: %d\n", days_above_average);
    return 0;
}
```



```
Day 23 temperature: 5555
Day 24 temperature: 6666
Day 25 temperature: 7777
Day 26 temperature: 8888
Day 27 temperature: 9999
Day 28 temperature: 11111
Day 29 temperature: 22222
Day 30 temperature: 33333
Average Temperature: 4071.699951
Highest Temperature: 33333.000000
Lowest Temperature: 11.000000
Number of days above average: 9


...Program finished with exit code 0
Press ENTER to exit console.
```

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**

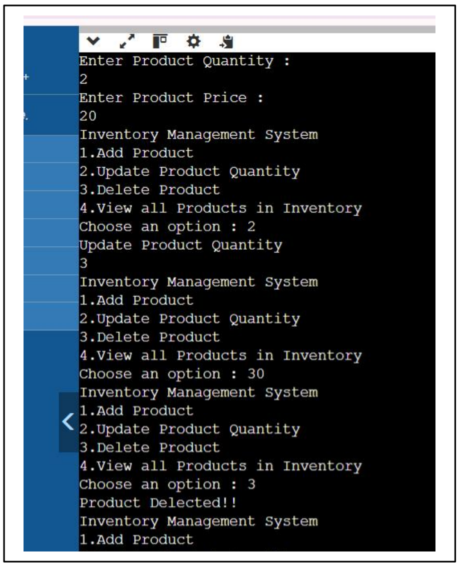
4. Ensure that the program handles ipr

nvalid 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.
```

```
#include<stdio.h>

int main(){

  char product_name[50];

  int quantities,options;

  float price;

  while(1){

    printf("Inventory Management System\n");

    printf("1.Add Product\n");

    printf("2.Update Product Quantity\n");

    printf("3.Delete Product\n");

    printf("4.View all Products in Inventory\n");
printf("Choose an option : ");
scanf("%d",&options);


switch(options){
  case 1:
    printf("Enter Product Name :\n ");
    scanf("%s",product_name);
    printf("Enter Product Quantity :\n");
    scanf("%d",&quantities);
    printf("Enter Product Price :\n");
    scanf("%f",&price);


    break;


  case 2:
    printf("Update Product Quantity\n");
    scanf("%d",&quantities);
    break;
  case 3:
    quantities=0;
    price=0;
    printf("Product Delected!!\n");
    break;
  case 4:
    if(quantities!=0){
      printf("Product Name : %s Quantity : %d Price :%f\n",product_name,quantities,price);
    }
    else{
```



```
Enter Product Quantity :
2
Enter Product Price :
20
Inventory Management System
1.Add Product
2.Update Product Quantity
3.Delete Product
4.View all Products in Inventory
Choose an option : 2
Update Product Quantity
3
Inventory Management System
1.Add Product
2.Update Product Quantity
3.Delete Product
4.View all Products in Inventory
Choose an option : 30
Inventory Management System
1.Add Product
2.Update Product Quantity
3.Delete Product
4.View all Products in Inventory
Choose an option : 3
Product Delected!!
Inventory Management System
1.Add Product
```

```
                printf("No Products Present\n");

            }

            break;

        default:

            break;

        }

    }

}
```