Assignment

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

Output:

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

2.

```
// #        *
// #      *   *
// #     *   *   *
// #   *   *   *   *
// # *   *   *   *   *
#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;
}
```

Output:

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

```
//WAP to display multiplication table from 1 to 10
```

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

Output:

```
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
```

```
//wap to reverse a number using for loop
#include<stdio.h>
int main(){
   int n,rev=0,rem;
   printf("Enter the number:");
   scanf("%d",&n);
   for(int i=0;i<=n;i++){
      rem=n%10;
      rev=rev*10+rem;
      n=n/10;
   }
   printf("\n");
   printf("Reverse number is=%d \n",rev);
   return 0;
}
```

Output:

```
Enter the number:234
Reverse number is=432
```

```
//wap to print fibonacci series upto n terms
#include<stdio.h>
int main(){
   int n,a=0,b=1,next;
   printf("Enter the number:");
   scanf("%d",&n);
   for(int i=1;i<=n;i++){
      printf("%d \n",a);
      next=a+b;
```

```
        a=b;
        b=next;
    }
    return 0;
}
```

Output:

```
Enter the number:10
0
1
1
2
3
5
8
13
21
34
55
```

```
// 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,guess,tries=5;
    srand(time(0));
    num=rand()%21;
    printf("This is a guessing game.\n");
    printf("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.\n", tries);
        printf("Enter a guess: ");
        scanf("%d", &guess);
        if(guess<0 || guess>20){
            printf("Please enter a number between 0 and 20.\n");
            continue;
        }
        if(guess==num){
            printf("Congratulations. You guessed it!.\n");
            break;
        }
        else if (guess < num) {
            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);
        }
        if(i==tries-1){
            printf("Sorry, you've used all your tries. The number was %d.\n", num);
        }
    }
    return 0;
}
```

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: 5
Sorry, 5 is wrong. My number is greater than that.
You have 5 tries left.
Enter a guess: 7
Sorry, 7 is wrong. My number is greater than that.
You have 5 tries left.
Enter a guess: 18
Sorry, 18 is wrong. My number is less than that.
You have 5 tries left.
Enter a guess: 20
Sorry, 20 is wrong. My number is less than that.
You have 5 tries left.
Enter a guess: 5
Sorry, 5 is wrong. My number is greater than that.
Sorry, you've used all your tries. The number was 9.
```

// 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 4 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.

```c
// 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 num,i,sum=0;
    printf("Enter up to 20 integers (enter 1 to stop):\n");
    for(i=0;i<20;i++){
        printf("Enter number %d:",i+1);
        scanf("%d",&num);
        if(num==1){
            break;
        }
        if(num<0){
            continue;
        }
        if(num%2==0){
            sum=sum+num;
        }
    }
    printf("Sum of even numbers:%d \n",sum);
    return 0;
}
```

Output:

```
Enter up to 20 integers (enter 1 to stop):
Enter number 1:4
Enter number 2:7
Enter number 3:-3
Enter number 4:2
Enter number 5:8
Enter number 6:5
Enter number 7:10
Enter number 8:-1
Enter number 9:1
Sum of even numbers:24
```

```
// 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).//dont give structure make it as inputs

// 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()
{
   int choice;
   int acc_num=0,entered_acc_num;
   char acc_holder_name[20];
   float balance=0.0,deposit_amount,withdraw_amount;
   while(1){
      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");
      printf("Choose an option: \n");
      scanf("%d",&choice);
      switch (choice){
         case 1:
```

```
    printf("Enter the account holder name:");
    scanf("%s",acc_holder_name);
    printf("Enter a new account number: ");
    scanf("%d", &acc_num);
    printf("Account created successfully! Account Number: %d \n",acc_num);
    break;
  case 2:
  printf("Enter account number:");
  scanf("%d",&entered_acc_num);
  if(entered_acc_num==acc_num){
    printf("Enter amount to deposit:");
    scanf("%f",&deposit_amount);
    if(deposit_amount<=0){
      printf("Deposit must be positive \n");
    }else{
      balance+=deposit_amount;
      printf("Deposit successful! New Balance: %f\n",balance);
    }
  }else{
    printf("Invalid account number.\n");
  }
  break;
case 3:
  if (entered_acc_num == acc_num) {
      printf("Enter amount to withdraw: ");
      scanf("%f", &withdraw_amount);

      if (withdraw_amount <= 0) {
        printf("Withdrawal amount must be positive.\n");
      } else if (withdraw_amount > balance) {
        printf("Insufficient balance.\n");
```

```
                    } else {
                        balance -= withdraw_amount;
                        printf("Withdrawal successful! New Balance: %f\n", balance);
                    }
                } else {
                    printf("Invalid account number.\n");
                }
                break;
        case 4:
                printf("Enter account number: ");
                scanf("%d", &entered_acc_num);

                if (entered_acc_num == acc_num) {
                    printf("Current Balance: %f\n", balance);
                } else {
                    printf("Invalid account number.\n");
                }
                break;

        case 5:
            printf("Exiting the system.\n");
            return 0;
        default:
            printf("Invalid option. Please choose again.\n");
        }
    }
    return 0;
}
```

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 the account holder name:John
Enter a new account number: 100
Account created successfully! Account Number: 100
Welcome to the Banking System
1. Create Account
2. Deposit Money
3. Withdraw Money
4. Check Balance
5. Exit
Choose an option:
2
Enter account number:100
Enter amount to deposit:500
Deposit successful! New Balance: 500.000000
Welcome to the Banking System
1. Create Account
2. Deposit Money
3. Withdraw Money
4. Check Balance
5. Exit
Choose an option:
3
Enter amount to withdraw: 200
Withdrawal successful! New Balance: 300.000000
Welcome to the Banking System
```

```
1. Create Account
2. Deposit Money
3. Withdraw Money
4. Check Balance
5. Exit
Choose an option:
4
Enter account number: 100
Current Balance: 300.000000
Welcome to the Banking System
1. Create Account
2. Deposit Money
3. Withdraw Money
4. Check Balance
5. Exit
Choose an option:
5
Exiting the system.
```

// 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 i,days;
    float sum=0,avg=0;
    float highest,lowest,above_avg_temp=0;
    printf("Enter the number of days in the month:");
    scanf("%d",&days);
    float temp[days];
    for (i=0;i<days;i++){
        printf("Day %d temperature:",i+1);
        scanf("%f",&temp[i]);
        sum=sum+temp[i];
    }
    avg=sum/days;
    highest=lowest=temp[0];
    for(i=0;i<days;i++){
        if(temp[i]>highest){
            highest=temp[i];
        }
        if(temp[i]<lowest){
            lowest=temp[i];
        }
        if(temp[i]>avg){
            above_avg_temp++;
        }
    }
    printf("Average Temperature of Month:%f\n",avg);
    printf("Highest Temperature Recorded:%f\n",highest);
    printf("Lowest Temperature Recorded:%f\n",lowest);
    printf("Number of Days Above Average Temperature:%f\n",above_avg_temp);
    return 0;
}
```

Output:

```
Enter the number of days in the month:30
Day 1 temperature:77
Day 2 temperature:34
Day 3 temperature:67
Day 4 temperature:88
Day 5 temperature:36
Day 6 temperature:97
Day 7 temperature:36
Day 8 temperature:65
Day 9 temperature:92
Day 10 temperature:20
Day 11 temperature:35
Day 12 temperature:38

Day 13 temperature:45

Day 14 temperature:57

Day 15 temperature:89

Day 16 temperature:30

Day 17 temperature:24

Day 18 temperature:66

Day 19 temperature:89

Day 20 temperature:47

Day 21 temperature:83

Day 22 temperature:95

Day 23 temperature:62

Day 24 temperature:64

Day 25 temperature:74

Day 26 temperature:73

Day 27 temperature:26

Day 28 temperature:59

Day 29 temperature:80

Day 30 temperature:75

Average Temperature of Month:60.766666

Highest Temperature Recorded:97.000000

Lowest Temperature Recorded:20.000000

Number of Days Above Average Temperature:17.000000
```

// 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.


#include<stdio.h>
int main()
{
    int choice,quantity,new_quantity,product_exists=0;
    char name[20];
    float price;
    while(1){
        printf("\n Inventory Management System\n");
        printf("1.Add Product\n");
        printf("2.Update Product Quantity\n");
        printf("3.Delete Product\n");
        printf("4.View Product in Inventory\n");
        printf("5.Exit\n");
        printf("Choose an option:");
        scanf("%d",&choice);
        if(choice==1){
            printf("Enter product name:");
            scanf("%s",name);
            printf("Enter product quantity:");
            scanf("%d",&quantity);
            printf("Enter product price:");
            scanf("%f",&price);
            product_exists=1;
            printf("Product added successfully!\n");
        }
        else if(choice==2){
            if(!product_exists){
                printf("No product in Inventory.Please add a product first.\n");
            }
            else{
                printf("Enter new quantity for %s:",name);
                scanf("%d",&new_quantity);
                quantity=new_quantity;
                printf("Product quantity updated successfully!\n");
            }
        }
        else if(choice==3){
            if(!product_exists){
                printf("No product found in inventory to delete.\n");
            }
            else
            {
                product_exists=0;
                printf("Product deleted successfully!\n");
            }
        }
        else if(choice==4){
            if(!product_exists){
                printf("No product found in inventory.\n");
            }
            else
            {
                printf("Product Name:%s, Quantity:%d, Price:%f\n",name,quantity,price);
            }
        }
        else if(choice==5)
        {
            printf("Exiting the system.\n");
            break;
        }
```

```
        else
        {
            printf("Invalid\n");
        }
    }
    return 0;
}
```

Output:

```
 Inventory Management System
1.Add Product
2.Update Product Quantity
3.Delete Product
4.View Product in Inventory
5.Exit
Choose an option:
1
Enter product name:
widget
Enter product quantity:
10
Enter product price:
500
Product added successfully!

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

```
Choose an option:
2
Enter new quantity for widget:
20
Product quantity updated successfully!

 Inventory Management System
1.Add Product
2.Update Product Quantity
3.Delete Product
4.View Product in Inventory
5.Exit
Choose an option:
3
Product deleted successfully!

 Inventory Management System
1.Add Product
2.Update Product Quantity
3.Delete Product
4.View Product in Inventory
5.Exit
Choose an option:
1
Enter product name:
gadget
Enter product quantity:
20
Enter product price:
1000
Product added successfully!
```

```
 Inventory Management System
1.Add Product
2.Update Product Quantity
3.Delete Product
4.View Product in Inventory
5.Exit
Choose an option:
2
Enter new quantity for gadget:
30
Product quantity updated successfully!

 Inventory Management System
1.Add Product
2.Update Product Quantity
3.Delete Product
4.View Product in Inventory
5.Exit
Choose an option:
4
Product Name:gadget, Quantity:30, Price:1000.000000

 Inventory Management System
1.Add Product
2.Update Product Quantity
3.Delete Product
4.View Product in Inventory
5.Exit
Choose an option:
5
```

Exiting the system.