1.Print Table from 1 to 10 using nested while.

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

    }
    return 0;

}
```

2.Print pattern

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

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

        }
        printf("\n");
        i++;
    }
    return 0;
}
```

3.Print Pattern

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

```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++;
    }
    return 0;
}
```

4.Print table using do while.

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

5.Sum of n natural numbers using for.

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

```c
int main(){
    int i,n,sum=0;
    printf("Enter a Number::");
    scanf("%d",&n);
    for(i=0;i<=n;i++)
    {
        sum+=i;
    }
    printf("\nSum is::%d",sum);
}
```

6.Reverse a number using a for loop.

```c
#include<stdio.h>
int main(){
    int n,rev=0,rem;
    printf("Enter a Number::");
    scanf("%d",&n);
    for(n=n;n>0;n/=10)
    {
        rem=n%10;
        rev=rev*10+rem;
    }
    printf("\nREVERSE is::%d",rev);
}
```

7.Fibonacci series using a for loop.

```c
#include<stdio.h>
int main(){
    int i,n,t1=0,t2=1,next;
    printf("Enter value of n::");
    scanf("%d",&n);
    for(i=1;t1<=n;i++)
    {
        printf("%d ",t1);
        next=t1+t2;
        t1=t2;
        t2=next;
    }
    return 0;
}
```

8.Pascal Triangle

```
#include<stdio.h>
int main(){
    int row,i,j,k,p=1;
    printf("Enter No:of rows::");
    scanf("%d",&row);
    for(i=0;i<row;i++){
        for(j=0;j<row-i-1;j++){
            printf(" ");
        }
        for(k=0;k<=i;k++){
            if(k==0||i==0)
            {
                p=1;
            }
            else{
                p=p*(i-k+1)/k;
            }
            printf("%4d",p);
        }
        printf("\n");
    }
}
```

9.Guessing game

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
int main(){
    srand(time(0));
    int random=rand()%21;
    printf("Random selected is %d\n",random);
    printf("This is a guessing game.\n");
    printf("I have chosen a number between 0&20 which you must
guess\n");
    int i=5,g;
    do{
        printf("\nYou have %d tries left.\n",i);
        printf("Enter a Guess::");
        scanf("%d",&g);
        if(0<=g<=20){
```

```
            if(g>random){
                printf("\nSorry %d is wrong,My guess is lesser than
that",g);
            }
            else if(g<random){
                printf("\nSorry %d is wrong,My guess is greater than
that",g);
            }
            else{
                printf("\nCongrats you guess it");
                break;
            }


        }
        else{
            printf("\nChoose between 0&20");
        }
        i--;

    }while(i>0);
}
```

10.Filter Even Numbers.

```
#include<stdio.h>
int main(){
    int a[20],i=0,j,sum=0;
    printf("Enter up to 20 integers(enter -1 to stop)::\n");
    do{
        if(i<=20){
            scanf("%d",&a[i]);
            if(a[i]==-1){
                break;
            }
        }
        else{
            printf("\nOnly 20 integers allowed");
        }
        i++;
    }
    while(i<20);
    for(j=0;j<=i-1;j++){
```

```
        if(a[j]>0){
            if(0==a[j]%2){
                sum+=a[j];
            }
            else{
                continue;
            }

        }
        else{
            continue;
        }

    }
    printf("\nSum of Even numbers::%d",sum);
    return 0;
}
```

11.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(){

    int ac_c,ac_n,balance=0,deposit,op,withd;

    char a;

    printf(" Welcome to the Banking System\n1. Create Account \n2.
Deposit Money \n3. Withdraw Money \n4. Check Balance \n5. Exit");

    do{

    printf("\nChoose an option::");

    scanf("%d",&op);

    switch (op)

    {

    case 1:

        printf("\nEnter account holder name::");

        scanf("%s",&a);

        printf("\nAccount created successfully!");

        printf("Account Number::");
```

```
        scanf("%d",&ac_n);
        break;
    case 2:
        printf("\nEnter Account Number::");
        scanf("%d",&ac_c);
        if(ac_c==ac_n){
            printf("\nEnter amount to deposit::");
            scanf("%d",&deposit);
            balance+=deposit;
            printf("\nDeposit Success! New Balance::%d",balance);


        }
        else{
            printf("\nWrog Account NUmber");
        }
        break;
    case 3:
        printf("\nEnter Account Number::");
        scanf("%d",&ac_c);
        if(ac_c==ac_n){
            printf("\nEnter amount to withdraw::");
            scanf("%d",&withd);
            if(balance<withd){
                printf("\nInsufficient Balance");
            }
```

```
            else{
                balance-=withd;
                printf("\nWithdrawal Success! New
Balance::%d",balance);
            }

        }
        else{
            printf("\nWrog Account NUmber");
        }
        break;
    case 4:
        printf("\nEnter Account Number::");
        scanf("%d",&ac_c);
        if(ac_c==ac_n){
            printf("\nCurrent Balance ::%d",balance);

        }
        else{
            printf("\nWrog Account NUmber");
        }
        break;
    case 5:
        printf("\nExiting...");
        return 0;
```

```
    default:

        break;

    }

    }while(1);



}
```

12.Problem Statement 4: Weather Data Analysis

Description: Write a program that collects daily temperature data for a month and analyses 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 day = 30;
```

```
    float temp[30];

    float sum = 0.0, avg, high, low;

    int count = 0;

    printf("Enter temperatures for each day of the month (30
days):\n");

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

        printf("Day %d temperature: ", i + 1);

        scanf("%f", &temp[i]);

        sum += temp[i];

    }

    avg = sum / day;

    high = temp[0];

    low = temp[0];



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

        if (temp[i] > high) {

            high = temp[i];

        }

        if (temp[i] < low) {

            low = temp[i];

        }

        if (temp[i] > avg) {

            count++;

        }

    }
```

```c
    printf("\nAverage Temperature :: %.1f\n", avg);

    printf("Highest Temperature :: %.1f\n", high);

    printf("Lowest Temperature :: %.1f\n", low);

    printf("No:of Days with Above Average Temperature: %d\n", count);



    return 0;

}
```

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>
#include<string.h>
int main(){
    int q,q_n,op;
    float p;
    char a[10],d[10];
    printf("Inventory Management System \n1. Add Product \n2. Update
Product Quantity \n3. Delete Product \n4. View All Products in
Inventory \n5. Exit");
    do{
    printf("\nChoose an option::");
    scanf("%d",&op);
    switch (op)
    {
    case 1:
        printf("\nEnter product name::");
        scanf("%s",&a);
        printf("\nEnter Quantity::");
        scanf("%d",&q);
        printf("\nEnter Price::");
        scanf("%f",&p);
        break;
    case 2:
        printf("\nEnter New Quantity::");
        scanf("%d",&q_n);
        q=q_n;
        printf("\nUpdated Quantity! New Quantity::%d",q);
        break;
    case 3:
        printf("\nEnter Product name to delete::");
        scanf("%s",&d);
        if(strcmp(d,a)==0){
            strcpy(a," ");
            q=0;
```

```
            p=0;
        }
        else{
            printf("\nWrong item ");
        }
        break;
    case 4:
        printf("\n Product Name:%s, Quantity:%d, Price: $%.2f",a,q,p);
        break;
    case 5:
        printf("\nExiting...");
        return 0;
    default:
        break;
    }
    }while(1);

}
```