1-Wap to print multiplication table from 1 to 10

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

}
```

2-print pattern

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

```
#include<stdio.h>
int main()
```

```
{
   int i=0;

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

3- print pattern

```
   *
  * *
 * * *
* * * *
* * * * *
#include<stdio.h>
int main()
{

   int i=5;
```

```
    int a=1;

    while(i>0)
    {
        int j=0;
        while(j<i-1)
        {
            printf(" ");
            j=j+1;
        }
        int k=0;
        while(k<a)
        {
            printf("* ");
            k=k+1;
        }
        a=a+1;
        i=i-1;
        printf("\n");
    }
    return 0;
}
```

4-WAP to reverse a number using for loop

```
#include <stdio.h>

int main()
```

```
{
   int n,rem,rev=0;
   printf("Enter the number: ");
   scanf("%d",&n);
   int i;
   for(i=n;i>0;i=i/10)
   {
      rem=i%10;
      rev=rev*10+rem;
   }
   printf("%d",rev);
   return 0;
}
```

5-)WAP to print fibinocci series upto n terms using for loop

```
#include<stdio.h>
int main()
{
   int number, first = 0, second = 1,next;

   printf("Enter the number: ");
   scanf("%d", &number);
   if (number>= 0)
      printf("%d\n", first);
   if  (number>= 1)
      printf("%d\n", second);
   next=first+second;
```

```
  for(;next<=number;)
  {
     printf("%d\n",next);
     first=second;
     second=next;
     next=first+second;
  }
  return 0;
}
```

6- WAP to print Pascal’s triangle

```
#include <stdio.h>
int main()
{
  int n;
  printf("Enter noof 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;
```

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

7-guess game

```
#include<stdio.h>
#include <stdlib.h>
#include<time.h>
int main()
{
  srand(time(0));
  int random = rand() % 20 + 1;
  int a=0;
  printf("This is a guessing game\n");
  printf("Guess the number that i have choosen from 1 to 20\n");
  int guess;

  for(int i=5;i>0;i--)
  {
    printf("\n");
    printf("You have %d tries left\n",i);
    printf("Enter a guess: ");
    scanf("%d",&guess);
    if(guess==random)
    {
```

```
        a=1;
        printf("Congrats\n");
        break;
      }

      else
      {
        if(random>guess)
        {
          printf("Sorry %d is wrong, my number is bigger\n",guess);
          printf("\n");
        }
        else
        {
          printf("Sorry %d is wrong, my number is lesser\n",guess);
          printf("\n");
        }
      }
    }

    if(a==0)
    {
      printf("You lost the game");
    }

    return 0;
}
```

8- 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.

```
#include<stdio.h>
int main()
{
   int number,a=0,s=0;
   printf("Enter upto 20 integers.Enter -1 to stop\n");
   while(a<20)
   {
      a=a+1;
      scanf("%d",&number);
      if(number%2 ==0 && number>0)
      {
         s=s+number;
      }
      else if(number==-1)
      {
         break;
      }
      else
      {
         continue;
      }
   }
   printf("\n");
   printf("the sum of even numbers are %d",s);
```

```
    return 0;

}
```

9- 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 op,money,orginal,balance=0;
   char name[]="";
   int number;
   int a=0;
   printf("Welcome to banking System\n");
```

```
   printf("1.Create account\n2.Deposit money\n3.Withdraw money\n4.Check
balance\n5.Exit");
   while(a!=1)
   {
      printf("\nChoose the option: ");
      scanf("%d",&op);
      switch(op)
      {
         case 1:
            printf("\n");
            printf("Enter account name: ");
            scanf("%s",name);
            printf("Enter account number: ");
            scanf("%d",&orginal);
            printf("Account created\n");
            break;

         case 2:
            printf("\n");
            printf("Enter account number: ");
            scanf("%d",&number);
            if(number==orginal)
            {
               printf("Enter the amount to deposit: ");
               scanf("%d",&money);
               balance=balance+money;
               printf("New account balance is %d\n",balance);
               break;
```

```
    }
    else
    {
       printf("Invalid account number\n");
       break;
    }


case 3:
    printf("\n");
    printf("Enter account number: ");
    scanf("%d",&number);
    if(number==orginal)
    {
       printf("Enter the amount to withdraw: ");
       scanf("%d",&money);
       if(money>balance)
       {
          printf("Insufficient balance\n");
          break;
       }
       else
       {
          balance=balance-money;
          printf("New account balance is %d\n",balance);
          break;
       }
    }
```

```
    else
    {
       printf("Invalid account number\n");
       break;
    }




case 4:
    printf("\n");
    printf("Enter account number: ");
    scanf("%d",&number);
    if(number==orginal)
    {
       printf("Account balance is %d\n",balance);
       break;
    }
    else
    {
       printf("Invalid account number\n");
       break;
    }


case 5:
    a=1;
    printf("\n");
```

```
            printf("Thank You\n");
            break;
        }
    }
    return 0;
}
```

10- 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()
{
   float arr[30];
   float s=0;
   printf("Enter temperatures for each day of the month (30 days)");
   printf("\n");
   for(int i=0;i<30;i++)
   {
      printf("Day %d temp: ",i+1);
      scanf("%f",&arr[i]);
      s=s+arr[i];
   }
   int cnt=0;
   float avg=s/30;
   float small=arr[0];
   float large=arr[0];
   for(int i=0;i<30;i++)
   {
      if (arr[i]<small)
      {
         small=arr[i];
      }

      if(arr[i]>large)
      {
         large=arr[i];
      }
```

```
      if(arr[i]>avg)
      {
         cnt=cnt+1;
      }
   }
   printf("\n");
   printf("The average temperature is %f\n",avg);
   printf("The highest temperature is %f\n",large);
   printf("The lowest temperature is %f\n",small);
   printf("Number of Days Above Average Temperature is %d",cnt);
   return 0;
}
```

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. 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()

{

   char name[]="";

   int qnty,op,a=0;
```

```
  float price;

  while(a!=1)
  {

    printf("Inventory Management System\n1. Add Product\n2. Update Product Quantity\n3. Delete Product\n4. View Products\n5. Exit\n");
    printf("Choose an option: ");
    scanf("%d", &op);

    switch(op)
    {
    case 1:
      printf("Enter product name: ");
      scanf("%s",name);
      printf("Enter product quantity: ");
      scanf("%d", &qnty);
      printf("Enter product price: ");
      scanf("%f", &price);
      printf("Detailes entered successfully!\n");
      break;


    case 2:
      printf("Updated product quatity:");
      scanf("%d",&qnty);
      printf("Product quantity updated successfully\n");
      break;
```

```
case 3:
   name[0]='\0';
   qnty = 0;
   price = 0;
   printf("Product deleted successfully\n");
   break;



case 4:
   if(name[0] != '\0')
   {
      printf("Product Name: %s | Quantity: %d | Price: %f \n",name,qnty,price);
   }
   else
   {
      printf("Empty\n");
   }

   break;



case 5:
   a=1;
   printf("\n");
   printf("Thank You\n");
   break;
```

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