

1.WAP to display multiplication table from 1 to 10

```
#include<stdio.h>

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

2.WAP to display following pattern mentioned below

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

```
#include<stdio.h>

int main(){
    int i=0,j;

    while(i<=5){
        j=1; //j ivide aahn initialise cheyyandat
        while(j<=i){
```

```

        printf("* ");
        j++;
    }
    printf("\n");
    i++;

}

}

```

3. WAP to display following pattern mentioned below

```

*
* *
* * *
* * * *
* * * * *

```

```

#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++;

}
}
```

5. WAP To print numbers from 1 to 10 using do while loop

```
#include<stdio.h>

int main(){
    int i=1;
    do
    {
        printf("%d ->",i);
        i++;
    } while (i<=10);

}
```

6.WAP to multiplication table using do while loop

```
#include<stdio.h>

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

```

{

    printf("%d * %d = %d\n",j,i,i*j);

    j++;

} while (j<=10);

printf("\n");

i++;

} while (i<=10);

}

```

7. WAP to print the numbers between 1 to 10 using for loop

```

#include<stdio.h>

int main(){

    int i;

    for(i=1;i<=10;i++){

        printf("%d\n",i);

    }

}

```

8. WAP to calculate sum of first n natural numbers using for loop

```

#include<stdio.h>

int main(){

    int limit,sum=0;

    printf("enter the limit");

    scanf("%d",&limit);

```

```
for(int i=1;i<=limit;i++){  
    sum=sum+i;  
}  
printf("summation=%d",sum);  
return 0;  
  
}
```

9. WAP to reverse number using for loop

```
#include<stdio.h>  
  
int main(){  
    int num,rev=0;  
    printf("enter the number");  
    scanf("%d",&num);  
  
    for(num=num;num>0;num/=10){  
        int rem=num%10;  
        rev=rev*10+rem;  
    }  
    printf("reverse= %d",rev);  
}
```

10. WAP to print fibonacci series upto n times

```
#include<stdio.h>  
  
int main(){  
    int n,num1=0,num2=1;  
    printf("enter the limit");  
    scanf("%d",&n);
```

```

for(int i=0;i<=n;i++){
    printf("%d-->",num1);
    int nth=num1;
    num1=num2;
    num2=nth+num1;
}
}

```

11. WAP to print infinite for loop

```

#include<stdio.h>
int main(){
    int i=1;
    for(;;)
    return 0;
}

```

12. WAP to print pascals triangle using for loop till 8 rows

```

#include <stdio.h>

int main() {
    int row,i,j,k,p=1;
    printf("Enter 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 || k==i){
            p=1;
        }else{
            p=p*(i-k+1)/k;
        }
        printf("%d ",p);
    }
    printf("\n");
}

}

```

13. Program for break and continue statement

```

#include<stdio.h>

int main(){
    int i;
    printf("BREAK\n");
    for(i=0;i<10;i++){
        if(i==5){
            break;

        }
        printf("%d-->",i);
    }

    printf("CONTINUE\n");
    for(i=0;i<10;i++){
        if(i==5){
            continue;

```

```

    }
    printf("%d-->",i);
}

}

```

14. Problem statement : Guess The Number Game

Requirements:

- 1.In this challenge, you are going to create a "Guess the Number" Cprogram**
- 2.Your program will generate a random number from 0-20**
- 3.You will then ask user to guess it**
- 4.User shouls only able to enter the number from 0-20**
- 5.The program will indicate to the user if each guess is too high or too low**
- 6.The player win the game if they can guess the number within 5 tries**

```

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

int main(){
    int random_number,num;

    int limit=5;

    srand(time(NULL));

    random_number=rand()%21;
    printf("%d\n",random_number);


    printf("This is a guessing game\nI have choosen a number between 0 and 20 which you must
    guess.\nYou have 5 tries left\n\n");


    for(int i=0;i<=5;i++){

```



```

printf("Enter a guess\t");
scanf("%d",&num);

if(num==random_number){
    printf("Congratulation you guessed it");
    break;
}else if(num>random_number && num<20){
    printf("Sorry,%d is wrong.My number is less than that\n",num);
    printf("you have %d tries left\n\n",(limit-1)-i);

}else if(num<random_number && num>0){
    printf("Sorry,%d is wrong.My number is greater than that\n",num);
    printf("you have %d tries left\n\n",(limit-1)-i);

}else{
    printf("invalid number enter number between 0 and 20\n");
    printf("you have %d tries left\n\n",(limit-1)-i);
}

if((limit-1)-i == 0){
    printf("sorry your tries are over");
    break;
}

}

}

```

15. Problem statement :Filter even numbers with continue.

Description

Write a c program that prompts the user to enter a series of integers (upto a maximum of 20). The program should calculate and display the sum of

all even number entered while skipping any negative numbers. Use the continue statement to skip processing for negative numbers.

Requirements:

1. Prompt the user for upto 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 the running toatal sum.
5. After all inputs display the total sum of even numbers

Example Input/Output:

Enter upto 20 integer:

4

7

-3

2

8

-5

10

-1

the sum of even numbers= 24

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

```
int main(){
```

```
    int num_arr[20],n,sum=0;
```

```
    printf("enter the number of elements.You can enter upto 20\n");
```

```

scanf("%d",&n);

for(int i=1;i<=n;i++){
    printf("Enter %d th element: ",i);
    scanf("%d",&num_arr[i]);
}
printf("array elements=");
for(int i=1;i<=n;i++){
    printf("%d,",num_arr[i]);

}
printf("\n");


for(int i=1;i<=n;i++){
    if(num_arr[i]>=0 && num_arr[i]%2==0){
        sum=sum+num_arr[i];

    }
    continue;
}

printf("summation=%d",sum);
}
/*

```

16.Problem Statement : 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 option;
```

```
    int accountnum,ac_no;
```

```
    char name[10];
```

```
    float deposit,withdraw;
```

```
    float balance=0.0;
```

```
    printf("welcome to the banking system \n");
```

```
    while (1)
```

```
    {
```

```
        printf("1.Create Account \n");
```

```
        printf("2.Deposite Money \n");
```

```
        printf("3.withdraw money \n");
```

```
        printf("4.check balance \n");
```

```
        printf("5.exit \n");
```

```
        printf("choose an option:");
```

```
        scanf("%d",&option);
```

switch (option)

{

case 1:

printf("enter account holder name: \n");

scanf(" %[^\\n]*%c", name);

printf("Enter account number: ");

scanf("%d",&accountnum);

printf("Hello %s your account has been sucessfully created ! account number:%d\\n\\n",name,accountnum);

break;

case 2:

printf("enter account number \n");

scanf("%d",&ac_no);

if(ac_no==accountnum){

printf("Enter the amount to deposit:");

scanf("%f",&deposit);

printf("You have Successfully deposited %.2f in your account\\n\\n",deposit);

balance=balance+deposit;

}else{

printf("Invalid account number please try again\\n\\n");

}

break;

case 3:

printf("enter account number \n");

scanf("%d",&ac_no);

if(ac_no==accountnum){

printf("Enter the amount to withdraw:");

scanf("%f",&withdraw);

if(balance>=0 && balance>=withdraw){

```

        printf("Withdraw successfull\n\n");

        balance=balance-withdraw;
    }else{

        printf("Insufficent balance\n\n");

    }
}
else{

    printf("Invalid account number please try again\n\n");

}

break;
case 4:

    printf("enter account number \n");

    scanf("%d",&ac_no);

    if(ac_no==accountnum){

        printf("Account Balance: %.2f\n\n",balance);

    }else{

        printf("Invalid account number please try again\n\n");

    }

    break;
case 5:

    printf("exiting the system");

    return 0;


default:

    printf("Invalid Options check again\n\n");

    break;

}

}

}

```

17. Problem Statement : 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,count=0;
```



```
float temp_data[30],average,temp_sum=0,max,min;
```

```
printf("Enter the days");
```

```
scanf("%d",&days);
```

```
for(int i=1;i<=days;i++){
```

```
    printf("Day %d temperature= ",i);
```

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

```
}
```

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

```
for(int i=1;i<=days;i++){
```

```
    printf("%.2f-->",temp_data[i]);
```

```
}
```

```
for(int i=1;i<=days;i++){
```

```
    temp_sum=temp_sum+temp_data[i];
```

```
}
```

```
average=temp_sum/days;
```

```
max=temp_data[0];
```

```
for(int i=1;i<=days;i++){
```

```
    if(temp_data[i]>max){
```

```
        max=temp_data[i];
```

```
    }
```

```
    if(temp_data[i]==average){
```

```
        count=count+1;
```

```
    }
```

```
}
```

```

min=temp_data[0];
for(int i=0;i<days;i++){
    if(temp_data[i]<min){
        min=temp_data[i];
    }
}

printf("\nAverage Temperature of Month: %.2f\n",average);
printf("Highest Temperature Recorded: %.2f\n",max);
printf("Lowest Temperature Recorded: %.2f\n",min);
printf("Number of Days Above Average Temperature: %d",count);

}

/*

```

18.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 option,quantity,u_quantity;
```

```
    float price;
```

```
    char name[50];
```

```
    printf("Inventory Management System\n");
```

```
    while(1){
```

```
        printf("1. Add Product\n2. Update Product Quantity\n3. Delete Product\n4. View All Products in\nInventory\n5. Exit\n");
```

```
        printf("Enter any option\n");
```

```
        scanf("%d",&option);
```

```
switch (option)
{
case 1:
printf("Enter the product name: ");
scanf("%s",name);
printf("Enter the quantity: ");
scanf("%d",&quantity);
printf("Enter the Price: ");
scanf("%f",&price);

printf("\n NAME:%s\tQUANTITY:%d\tPRICE:%.2f\nProduct is Successfully
added\n\n",name,quantity,price);

break;

case 2:

printf("You can update the product quantity\n");
printf("Enter updated Quantity: ");
scanf("%d",&u_quantity);
printf("Successfully Updated\n\n");
quantity=u_quantity;
break;

case 3:

quantity=0;
price=0.00;
printf("Deleted succesfully\n\n");


break;
```

case 4:

```
    if( quantity==0 && price==0.00){  
        printf("No products in the list \n");  
    }else{  
        printf("NAME:%s\tQUANTITY:%d\tPRICE:%.2f\n\n",name,quantity,price);  
    }
```

case 5:

```
return 0;
```

```
    break;
```

default:

```
    printf("Invalid option check and retry\n\n");
```

```
    break;
```

```
}
```

```
}
```

```
}
```

