

```
//infinte loop
#include <stdio.h>
int main()
{
    char a=1;
    while(a<=10);
    {
        printf("a=%d",a);
        a++;
    }
    return 0;
}
```

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

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

/*
WAP to print pattern

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

```
*/
#include<stdio.h>
void main(){
    int num=1,limit;
    printf("enter limit:");
    scanf("%d",&limit);
    while(num<=limit){
        int i=1;
        while(i<=num){
            printf("*");
            i++;
        }
        printf("\n");
        num++;
    }
}
```

```

/*
WAP to print pattern
*
* *
* * *
* * * *
* * * * *
*/
#include<stdio.h>
void main(){
    int num=1,limit;
    printf("enter limit:");
    scanf("%d",&limit);
    while(num<=limit){
        int i=1;
        while(i<=limit-num){
            printf(" ");
            i++;
        }
        int j=1;
        while(j<=num-1){
            printf("* ");
            j++;
        }
        printf("\n");
        num++;
    }
}

```

do while

```

/*
WAP to print number from 1-10
*/
#include<stdio.h>
void main(){
    int num=11;
    do{
        printf("%d\n",num);
        num++;
    }while(num<=10);
}

```

```

/*
WAP to reverse a number using for loop
*/
#include<stdio.h>
void main(){
    int num,rev=0;
    printf("enter number\n");
    scanf("%d",&num);
    for(;num!=0;){
        rev=(rev*10)+(num%10);
        num=num/10;
    }
    printf("reversed number is %d",rev);
}

```

```

/*
WAP to print fibonacci series using for loop;
*/
#include<stdio.h>
void main(){
    int limit,digit0=0,digit1=1,next;

    printf("enter limit\n");
    scanf("%d",&limit);
    printf("%d %d",digit0,digit1);
    for(int i=0;i<limit;i++){
        next=digit0+digit1;
        printf(" %d",next);
        digit0=digit1;
        digit1=next;
    }
}

```

```

/*
WAP to print pascals triangle using for loop;
*/
#include<stdio.h>
void main(){
    int row,num=1;
    printf("enter number of rows\n");
    scanf("%d",&row);

    for(int i=0;i<row;i++){
        for(int j=0;j<row-i-1;j++){
            printf(" ");
        }
        num=1;
        for(int k=0;k<=i;k++){
            printf("%d ",num);
            num=num*(i-k)/(k+1);
        }
        printf("\n");
    }
}

```

- 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 t 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 ties left. Enter a guess: 12

Sorry, 12 le wrong. My number is less than that.

You have 4 tries left Enter a guess: 0

Sorry, 8 is wrong. My number is less than that

You have 3 bries left. Enter a quess:4

Sorry, 4 is wrong. My number is less than that.

You have 2 les loft. Enter a guess: 2

Congratulations. You guessed it

```
#include <stdio.h>
#include <stdlib.h>
int random_fn();
int main() {
    int count=5,guess;
    int rand_nnum=random();
    printf("This is gussing game\n");
    printf("I have chosen a number between 0 and 20 which you must guess.\n");
    while(count!=0){
        printf("you have %d chances to win\n",count);
        printf("enter the guess:");
        scanf("%d",&guess);
        if(guess==rand_nnum){
            printf("congragulations.you gussed it\n");
        }
        else if(guess<rand_nnum){
            printf("sorry %d is wrong. My number is greater than that\n",guess);
        }
        else if(guess>rand_nnum){
            printf("sorry %d is wrong. My number is less than that\n",guess);
        }
        count--;
        printf("\n");
    }
}

int random_fn(){

    // Generate and print a random number between 1 and 20
    int random_number = (rand() % 20) + 1;
    printf("Random number between 1 and 20: %d\n", random_number);
    return random_number;
}
```

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.

Requirements:

Prompt the user for up to 20 integers.

Use a loop to read each integer.

If an integer is negative, use continue to skip adding it to the sum.

If an integer is even, add it to a running total sum.

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

25

Sum of even numbers: 24

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

```
#define limit 20
```

```
int main() {
```

```
    int n,sum=0;
```

```
    printf("enter 20 numbers\n");
```

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

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

```
        if(n<-1){
```

```
            continue;
```

```
        }
```

```
        else if(n%2==0){
```

```
            sum=sum+n;
```

```
        }
```

```
        else if(n==-1){
```

```
            break;
```

```
        }
```

```
    }
```

```
    printf("sum is %d",sum);
```

```
}
```

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.

Sum of even numbers: 24

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

```
#define limit 20
```

```
int main() {
```

```
    printf("enter the banking system\n");
```

```
    printf("1.create account\n2. Deposit Money\n3. Withdraw Money\n4. Check Balan\n5. Exit\n");
```

```
    int i;
```

```
    char name[20];
```

```
    int acct_number;
```

```
    float deposit,balance=0,withdraw;
```

```
    while(1){
```

```
        printf("choose action\n");
```

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

```

switch(i){
    case 1:
        printf("enter acct holder name\t:");
        scanf("%s",name);
        printf("account created successfully\n");
        printf("account number is \t:1001\n");
        break;
    case 2:
        printf("enter account number\t:");
        scanf("%d",&acct_number);
        if(1001==acct_number){
            printf("amt to deposit\t:");
            scanf("%f",&deposit);
            balance=balance+deposit;
            printf("Deposit successfull!! New balance is %f\n",balance);
        }
        else{
            printf("wrong acct number\n");
        }
        break;
    case 3:
        printf("enter account number\t:");
        scanf("%d",&acct_number);
        if(1001==acct_number){
            printf("amt to withdraw\t:");
            scanf("%f",&withdraw);
            if(balance<withdraw){
                printf("not enough balance");
            }
            else{
                balance=balance-withdraw;
                printf("Withdraw successfull!! New balance is %f\n",balance);
            }
        }
        else{
            printf("wrong acct number\n");
        }
        break;
    case 4:
        printf("enter account number:\t");
        scanf("%d",&acct_number);
        if(1001==acct_number){
            printf("current acct balance is %f\n",balance);
        }
        else{
            printf("wrong acct number\n");
        }
        break;
    case 5:
        printf("exiting system\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>
```

```
#define limit 10
```

```
int count=0,j;
```

```
float sort[limit];
```

```
void main(){
```

```
    float temp[limit],sum=0,avg,t;
```

```
    for(int day=0;day<limit;day++){
```

```
        printf("enter temp of day %d : ",day+1);
```

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

```
        sort[day]=temp[day];
```

```
    }
```

```
    for(int i=0;i<limit;i++){
```

```
        for(int l=0;l<limit-1-i;l++){
```

```
            if(sort[l]>sort[l+1]){
```

```
                t=sort[l];
```

```
                sort[l]=sort[l+1];
```

```
                sort[l+1]=t;
```

```
            }
```

```
        }
```

```
    }
```

```
    printf("1:avg temprature\n2:highest temprature\n3:lowest temprature\n4:number of days\nabove avg\n");
```

```
    while(1){
```

```
        int action;
```

```
        printf("choose action\t:");
```



```

scanf("%d",&action);
switch(action){
    case 1:
        for(int i=0;i<limit;i++){
            sum=sum+temp[i];
        }
        avg=sum/limit;
        printf("average value is %f\n", avg);
        break;
    case 2:
        printf("highest temp is %f\n",sort[limit-1]);
        break;
    case 3:
        printf("lowest temp is %f\n",sort[0]);
        break;
    case 4:
        for(int i=0;i<limit;i++){
            sum=sum+temp[i];
        }
        avg=sum/limit;
        while(temp[j]>avg){
            count++;
            j++;
        }
        printf("no days above avg is %d\n",count);
        break;
}

}

}

```

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 pdt_name[20];
    int pdt_no,action;
    float pdt_price;
    printf("Inventory managemnet system\n");
    printf("1.add product\n2. Update Product Quantity\n3. Delete Product\n4. View All Products
in Inventory\n5. Exit\n"); while(1){
        printf("enter action\n");
        scanf("%d",&action);
        switch(action){
            case 1:
                printf("enter pdt name\t:");
                scanf("%s",pdt_name);
                printf("enter no of products\t:");
                scanf("%d",&pdt_no);
                printf("enter price\t:");
                scanf("%f",&pdt_price);
                break;
            case 2:
                printf("enter updated Quantity\n");
                scanf("%d",&pdt_no);
                printf("product quantity updated\n");
                break;
            case 3:
                pdt_name[0]='\0';
                pdt_no='\0';
                pdt_price='\0';
                printf("product details deleted\n");
                break;
            case 4:
                printf("pdt name\t:%s\n",pdt_name);
                printf("pdt quantity\t:%d\n",pdt_no);
                printf("pdt price\t:%f\n",pdt_price);
                break;
            case 5:
                printf("exiting system\n");
                return 0;
        }
    }
}
```
