

### 1)WAP to create 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 \t",i,j,i*j);
            j++;
        }
        printf("\n");
        i++;
    }

    return 0;
}
```

### 2) WAP to print the pattern

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

```
int main()
{
    int num;
    scanf("%d",&num);

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

### 3)WAP to print the pattern

#include <stdio.h>

```
int main()
{
    int num;
    scanf("%d",&num);
```

```

int i=0,sp,j;
while(i < num)
{
    sp = 0;
    while(sp < num-i)
    {

        printf(" ");
        sp++;
    }
    j=0;
    while(j <= i)
    {

        printf("* ");

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

```

#### 4)WAP to create multiplication table from 1 to 10 using do while loop

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

```

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

        printf("\n");
        i++;
    } while (i <= 10);

    return 0;
}

```

#### 5)WAP to calculate sum of N natural numbers

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

```

int main()
{
    int num,sum=0;
    scanf("%d",&num);
    for(int i=1;i<=num;i++)
    {
        sum += i;
    }
    printf("sum of N numbers: %d",sum);
}

```

```
    return 0;
}
```

#### 6)WAP to reverse a number using for loop

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

```
int main()
{
    int num,rem=0,rev;
    scanf("%d",&num);
    for(int i=1;num != 0;i++)
    {
        rem = num%10;
        rev = rem + rev*10;
        num = num/10;
    }
    printf("Reversed number is: %d",rev);
    return 0;
}
```

#### 7)WAP to print fibinocci series upto n terms

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

```
int main()
{
    int num;
    scanf("%d",&num);
    int first=0,second =1,next =0;
    printf("The fibinocci series is:");
    for(int i=1;next <= num ;i++)
    {
        printf("%d ",next);
        first = second;
        second = next;
        next = first+second;
    }

    return 0;
}
```

#### 8) WAP to print Pascal's triangle ?

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

```
int main()
{
    int n;
    scanf("%d",&n);
    for(int i=1;i<=n;i++)
    {
        for(int sp=1;sp<=n-i;sp++)
        {
            printf(" ");
        }
        int a =1;
        for(int j=1;j<=i;j++)
        {
            printf("%d ",a);
            a=a*(i-j)/j;
        }
    }
}
```

```

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

```

## 9)Assignment

```

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

```

```

int main() {

```

```

    srand(time(0));

```

```

    int random_number = rand() % 21;
    int count = 0, num;

```

```

    printf("This is a guessing game.\n");
    printf("I have chosen a number between 0 and 20 which you must guess.\n");

```

```

    while (count < 20)

```

```

    {
        printf("Enter a guess:\n");
        scanf("%d", &num);
        count++;

```

```

        if (num == random_number)
        {
            printf("Congratulations! You guessed it in %d tries!\n", count);
            return 0;

```

```

        }
        else if (num < random_number)
        {
            printf("Sorry, %d is too low. Try again.\n", num);
        }
        else
        {
            printf("Sorry, %d is too high. Try again.\n", num);
        }

```

```

        printf("You have %d tries left.\n", 5 - count);
    }

```

```

    printf("You've used all your tries. The correct number was %d.\n", random_number);

```

```

    return 0;
}

```

**10)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>
#define LIMIT 20

int main()
{
    int num,sum=0;
    printf("Enter upto 20 integers(enter -1 to stop):\n");
    for(int i=1;i<=LIMIT;i++)
    {
        scanf("%d",&num);
        if(num == -1)
        {
            break;
        }
        else if(num < 0)
        {
            continue;
        }
        else if(num%2 == 0)
        {
            sum += num;
        }
    }
    printf("Sum of even numbers: %d",sum);

    return 0;
}
```

### **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.

15:15has context menu

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

```
int main() {
    char name[30];
```

```

char acc_num[30] = "";
float acc_balance = 0.0;
int op;

while(1) {
    printf("\nWelcome to the Banking System\n");
    printf("1. Create Account\n2. Deposit Money\n3. Withdraw Money\n4. Check Balance\n5.
Exit\n");
    printf("Choose an option: ");
    scanf("%d", &op);

    switch(op) {
        case 1: {
            printf("Enter account holder name: ");
            getchar(); // Consume the newline character left by previous input
            scanf("%s", name);
            printf("Enter Account Number: ");
            scanf("%s", acc_num);

            printf("Account created successfully!\n");
            acc_balance = 0.0; // Initialize balance for new account
            break;
        }

        case 2: {
            char new_acc_num[30];
            float deposit_amount;

            printf("Enter account number: ");
            scanf("%s", new_acc_num);

            if(strcmp(acc_num, new_acc_num) == 0) {
                printf("Enter amount to deposit: ");
                scanf("%f", &deposit_amount);

                acc_balance += deposit_amount;
                printf("Deposit successful! New Balance: %.2f\n", acc_balance);
            } else {
                printf("Wrong account number! Try again!\n");
            }
            break;
        }

        case 3: {
            char new_acc_num[30];
            float withdraw_amount;

            printf("Enter account number: ");
            scanf("%s", new_acc_num);

            if(strcmp(acc_num, new_acc_num) == 0) {
                printf("Enter amount to withdraw: ");
                scanf("%f", &withdraw_amount);

                if(withdraw_amount > acc_balance) {
                    printf("Insufficient balance! Your current balance is %.2f\n", acc_balance);
                } else {

```

```

        acc_balance -= withdraw_amount;
        printf("Withdrawal successful! New Balance: %.2f\n", acc_balance);
    }
} else {
    printf("Wrong account number! Try again!\n");
}
break;
}

case 4: {
    char new_acc_num[30];

    printf("Enter account number: ");
    scanf("%s", new_acc_num);

    if(strcmp(acc_num, new_acc_num) == 0) {
        printf("Current Balance: %.2f\n", acc_balance);
    } else {
        printf("Wrong account number! Try again!\n");
    }
    break;
}

case 5: {
    printf("Exiting the system.\n");
    return 0;
}

default:
    printf("Invalid option! Please choose a valid option.\n");
    break;
}
}

return 0;
}

```

## Problem Statement 2: 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

15:54has context menu

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

```
int main() {
    float temperature;
    int days;
    float data_array[30];
    printf("Enter temperatures for each day of the month (30 days):\n");
    for(int i=0;i<30;i++)
    {
        printf("Day %d temperature:",i+1);
        scanf("%f",&temperature);
        data_array[i] = temperature;
    }
    float sum,avrg;
    float highest_temp,lowest_temp;
    highest_temp=data_array[0];
    lowest_temp = data_array[0];
    for(int i=0;i<30;i++)
    {

        sum += data_array[i];

        if(highest_temp < data_array[i])
        {
            highest_temp = data_array[i];
        }
        if(lowest_temp > data_array[i])
        {
            lowest_temp = data_array[i];
        }
    }
    avrg = sum/30;
    printf("Average Temperature of Month:%.1f\n",avrg);
    printf("Highest Temperature Recorded:%.1f\n",highest_temp);
```

```

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

int days_abv_avg=0;
for(int i=0;i<30;i++)
{
    if(data_array[i] > avg)
    {
        days_abv_avg++;
    }
}
printf("Number of Days Above Average Temperature:%d",days_abv_avg);

return 0;
}

```

### Problem Statement 3: 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() {
    char prod_name[30];
    int quantities;
    float price;
    int op;
```

```
    while(1) {
```

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

```
        switch(op) {
            case 1: {
                printf("Enter product name: ");
                getchar(); // Consume the newline character left by previous input
                scanf("%s", prod_name);
                printf("Enter product quantity ");
                scanf("%d", &quantities);
                printf("Enter product price: ");
                scanf("%f", &price);
                printf("Details entered successfully!\n");
                break;
            }

```

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

```

```

        break;
    }

    case 3: {
        prod_name[0]='\0';
        quantities = 0;
        price = 0;
        printf("Product deleted successfully\n");
        break;
    }

    case 4: {
        if(prod_name[0] != '\0')
        {
            printf("Product Name: %s | Quantity: %d | Price: $%.2f \n",prod_name,quantities,price);
        }
        else
        {
            printf("Inventory is empty!\n");
        }

        break;
    }

    case 5: {
        printf("Exiting the system.\n");
        return 0;
    }

    default:
        printf("Invalid option! Please choose a valid option.\n");
        break;
    }
}

return 0;
}

```