

## Day6\_assignment\_abhirami

### //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 (Write a Program) to print the following 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++;
    }
}

```

## **//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) {

```

```

        printf("* ");
        j++;
    }

    printf("\n");
    num++;
}
}

```

### **WAP to print number from 1-10**

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

```

void main() {
    int num = 1;
    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: ");
    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: ");
    scanf("%d", &limit);

    printf("%d %d ", digit0, digit1);

    for(int i = 2; 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: ");
    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 tries left.**

**Enter a guess: 12 Sorry, 12 is wrong. My number is less than that. You have 4 tries left**

**Enter a guess: 0 Sorry, 0 is wrong. My number is less than that. You have 3 tries left.**

**Enter a guess: 4 Sorry, 4 is wrong. My number is less than that. You have 2 tries left. Enter a guess: 2 Congratulations. You guessed it**

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

```
#include <stdlib.h>
```

```
#include <time.h>
```

```
int generate_random_number();
```

```
int main() {
```

```
    int count = 5, guess, rand_num;
```

```
    srand(time(0));
```

```
    rand_num = generate_random_number();
```

```
    printf("This is a guessing game.\n");
```

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

```
    while (count > 0) {
```

```
        printf("You have %d tries left. Enter a guess: ", count);
```

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

```
        if (guess < 0 || guess > 20) {
```

```
            printf("Please enter a number between 0 and 20.\n");
```

```
            continue;
```

```
        }
```

```
        if (guess == rand_num) {
```

```
            printf("Congratulations! You guessed it!\n");
```

```
            break;
```

```
        } else if (guess < rand_num) {
```

```
            printf("Sorry, %d is too low. My number is greater than that.\n", guess);
```

```
        } else {
```

```
            printf("Sorry, %d is too high. My number is less than that.\n", guess);
```

```
        }
```

```
        count--;
```

```
        if (count == 0) {
```

```

        printf("Sorry, you've used all your tries. The number was %d.\n", rand_num);
    }
}

return 0;
}

int generate_random_number() {
    return rand() % 21;
}

```

**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>
```

```

int main() {
    int num, sum = 0;
    int count = 0;

    printf("Enter up to 20 integers (enter a negative number to skip it):\n");

    for (int i = 0; i < 20; i++) {
        printf("Enter integer %d: ", i + 1);
        scanf("%d", &num);

        if (num < 0) {
            continue;
        }
    }
}

```

```

        if (num % 2 == 0) {
            sum += num;
        }

        count++;
    }

    printf("Total sum of even numbers: %d\n", 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.**

**has context menu**

```
#include<stdio.h>
int main()
{
    int op,money,balance=0;
    char name[]="";
    int number;
    int a=0;
    printf("Welcome to banking System\n");
    printf("\nChoose the options\n 1.Create account \n 2.Deposit money \n 3.Withdraw money \n
4.Check balance \n 5.Exit");
    while(a!=1)
    {
        printf("\nChoose the option: ");
        scanf("%d",&op);
        switch(op)
        {
            case 1:
                printf("\n");
                printf("Enter the account name: ");
                scanf("%s",name);
                printf("Account created\n");
                printf("Account number is 101\n");
                break;
```



```

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

case 3:
    printf("\n");
    printf("Enter the account number: ");
    scanf("%d",&number);
    printf("Enter the amount to withdraw: ");
    scanf("%d",&money);
    balance=balance-money;
    printf("New account balance is %d\n",balance);
    break;

case 4:
    printf("\n");
    printf("Enter the account number: ");
    scanf("%d",&number);
    printf("The account balance is %d\n",balance);
    break;

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

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

**has context menu**

```
int main() {
    float temperatures[30], totalTemp = 0.0, averageTemp, highestTemp, lowestTemp;
    int daysAboveAverage = 0;

    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", &temperatures[i]);
        totalTemp += temperatures[i];
    }

    averageTemp = totalTemp / 30;

    highestTemp = temperatures[0];
    lowestTemp = temperatures[0];
```

```

for (int i = 1; i < 30; i++) {
    if (temperatures[i] > highestTemp) {
        highestTemp = temperatures[i];
    }
    if (temperatures[i] < lowestTemp) {
        lowestTemp = temperatures[i];
    }
}

for (int i = 0; i < 30; i++) {
    if (temperatures[i] > averageTemp) {
        daysAboveAverage++;
    }
}

printf("\nAverage Temperature: %.1f\n", averageTemp);
printf("Highest Temperature: %.1f\n", highestTemp);
printf("Lowest Temperature: %.1f\n", lowestTemp);
printf("Days Above Average: %d\n", daysAboveAverage);

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>
```

```
int main() {
```

```
    char pdt_name[20];
```

```
    int pdt_no, action;
```

```
    float pdt_price;
```

```
    printf("Inventory management system\n");
```

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

```
    while(1) {
```

```
        printf("Enter action: ");
```

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

```
        switch(action) {
```

```
            case 1:
```

```
                printf("Enter product name: ");
```

```
                scanf("%s", pdt_name);
```

```
printf("Enter number of products: ");
scanf("%d", &pdt_no);
printf("Enter product price: ");
scanf("%f", &pdt_price);
printf("Product added successfully!\n");
break;
```

case 2:

```
printf("Enter updated quantity: ");
scanf("%d", &pdt_no);
printf("Product quantity updated\n");
break;
```

case 3:

```
pdt_name[0] = '\0';
pdt_no = 0;
pdt_price = 0.0f;
printf("Product details deleted\n");
break;
```

case 4:

```
if (pdt_name[0] == '\0') {
    printf("No product available in inventory.\n");
} else {
    printf("Product name: %s\n", pdt_name);
    printf("Product quantity: %d\n", pdt_no);
    printf("Product price: %.2f\n", pdt_price);
}
break;
```

case 5:

```
printf("Exiting system\n");
return 0;
```

default:

```
printf("Invalid action. Please try again.\n");
```

```
}
```

```
}
```

```
return 0;
```

```
}
```

