

CLASS WORKS

1.

```
#include<stdio.h>
int main(){
    char a = 1;
    while (a<=10)
    {
        printf("a=%d\n",a);
        a++;
    }
}
```

OUTPUT

a=1
a=2
a=3
a=4
a=5
a=6
a=7
a=8
a=9
a=10

2 WAP to print the number between using do while

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

    int i = 1;
    do{
        printf("%d\n",i);

        i++;
    }
    while(i<=10);
}
```

OUTPUT

1
2

3
4
5
6
7
8
9
10

3.

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

```
int main() {  
    int i = 1;  
    do {  
        int j = 1;  
        do {  
            printf("%d x %d = %d\n", i, j, i * j);  
            j++;  
        }  
        while (j <= 10);  
        i++;  
    }  
    while (i <= 10);  
  
    return 0;  
}
```

OUTPUT

1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9

$$1 \times 10 = 10$$

$$2 \times 1 = 2$$

$$2 \times 2 = 4$$

$$2 \times 3 = 6$$

$$2 \times 4 = 8$$

$$2 \times 5 = 10$$

$$2 \times 6 = 12$$

$$2 \times 7 = 14$$

$$2 \times 8 = 16$$

$$2 \times 9 = 18$$

$$2 \times 10 = 20$$

$$3 \times 1 = 3$$

$$3 \times 2 = 6$$

$$3 \times 3 = 9$$

$$3 \times 4 = 12$$

$$3 \times 5 = 15$$

$$3 \times 6 = 18$$

$$3 \times 7 = 21$$

$$3 \times 8 = 24$$

$$3 \times 9 = 27$$

$$3 \times 10 = 30$$

$$4 \times 1 = 4$$

$$4 \times 2 = 8$$

$$4 \times 3 = 12$$

$$4 \times 4 = 16$$

$$4 \times 5 = 20$$

$$4 \times 6 = 24$$

$$4 \times 7 = 28$$

$$4 \times 8 = 32$$

$$4 \times 9 = 36$$

$$4 \times 10 = 40$$

$$5 \times 1 = 5$$

$$5 \times 2 = 10$$

$$5 \times 3 = 15$$

$$5 \times 4 = 20$$

$$5 \times 5 = 25$$

$$5 \times 6 = 30$$

$$5 \times 7 = 35$$

$$5 \times 8 = 40$$

$$5 \times 9 = 45$$

$$5 \times 10 = 50$$

$$6 \times 1 = 6$$

$$6 \times 2 = 12$$

$$6 \times 3 = 18$$

$6 \times 4 = 24$
 $6 \times 5 = 30$
 $6 \times 6 = 36$
 $6 \times 7 = 42$
 $6 \times 8 = 48$
 $6 \times 9 = 54$
 $6 \times 10 = 60$
 $7 \times 1 = 7$
 $7 \times 2 = 14$
 $7 \times 3 = 21$
 $7 \times 4 = 28$
 $7 \times 5 = 35$
 $7 \times 6 = 42$
 $7 \times 7 = 49$
 $7 \times 8 = 56$
 $7 \times 9 = 63$
 $7 \times 10 = 70$
 $8 \times 1 = 8$
 $8 \times 2 = 16$
 $8 \times 3 = 24$
 $8 \times 4 = 32$
 $8 \times 5 = 40$
 $8 \times 6 = 48$
 $8 \times 7 = 56$
 $8 \times 8 = 64$
 $8 \times 9 = 72$
 $8 \times 10 = 80$
 $9 \times 1 = 9$
 $9 \times 2 = 18$
 $9 \times 3 = 27$
 $9 \times 4 = 36$
 $9 \times 5 = 45$
 $9 \times 6 = 54$
 $9 \times 7 = 63$
 $9 \times 8 = 72$
 $9 \times 9 = 81$
 $9 \times 10 = 90$
 $10 \times 1 = 10$
 $10 \times 2 = 20$
 $10 \times 3 = 30$
 $10 \times 4 = 40$
 $10 \times 5 = 50$
 $10 \times 6 = 60$
 $10 \times 7 = 70$

10 x 8 = 80
10 x 9 = 90
10 x 10 = 100

4.

// w a p to print the 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 X %d = %d\n",i,j ,i*j);  
            j++;  
        }  
        i++;  
    }  
}
```

5.

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

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

OUTPUT

```
-----  
*  
* *  
* * *  
* * * *
```

* * * * *

6.

```
#include<stdio.h>
int main()
{
    int row = 4,i=1,j;
    while(i<=row){
        int space = row -i ;

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

OUTPUT

```
-----
      *
     **
    ***
```

7.

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

8.

```
#include<stdio.h>
int main(){
    int n,sum =0;
    printf("enter the natural number");
    scanf("%d",&n);
    for(int i =1;i<=n;i++){
        sum = sum + i;
    }
    printf("the sum of natural numbers are%d",sum);
    return 0;
}
```

9.

```
#include<stdio.h>
int main(){
    int n,r,reverse=0;
    printf("enter the number to be reversed");
    scanf("%d",&n);
    for(;n!=0;){
        r = n%10;
        reverse = reverse * 10 +r;
        n = n/10;
    }
    printf("the reverse of a number is %d",reverse);
}
```

OUTPUT

1234
4321

10.

```
#include<stdio.h>
int main(){
    int first ,second, next,n;
    first =0;
    second =1;
    printf("enter the number");
    scanf("%d",&n);
    printf("%d\n",first);
    for(int i = 0;i< n; i++){
```

```

        if(second>=n){
            break;
        }
        else{
            printf("%d\n",second);
            next = first + second;
            first = second;
            second= next;
        }

    }
}

```

OUTPUT

```

-----
0
1
1
2
3
5
8

```

11.

```

#include<stdio.h>
int main(){
    int i;
    for(i=0;i<=10;i++){
        if(i==5){
            break;
        }
        printf("%d\n",i);
    }
}

```


12.

```
#include<stdio.h>
int main(){
    int i;
    for(i=0;i<=10;i++){
        if(i==5){
            continue;
        }
        printf("%d\n",i);
    }
}
```

13.

```
#include <stdio.h>

int main() {
    int n, sum = 0;

    printf("enter up to 20 integers(enter -1 to stop)");
    for(int i =0;i<=20;i++){
        scanf("%d",&n);
        if (n==-1)
        {
            break;
        }
        else if (n<0)
        {
            continue;
        }
        else if (n%2==0)
        {
            sum = sum+n;
        }
    }
}
```

```

    }

}
printf("the sum of even numbers are =%d",sum);
return 0;
}

```

OUTPUT

```

-----

1
2
3
4
5
6
7
-1
2
the sum of even numbers are 14
-----
14.

```

ASSIGNMENTS

```

1.
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main() {
    int n, count = 4;
    srand(time(NULL));
    int randomnumber = rand() % 10;

    while(1){
        printf("enter the number");
        scanf("%d",&n);
        if(n<20 && n>0){
            printf("you have %d tries left\n",count);
            if(n == randomnumber){
                printf("you win the game\n");
            }
        }
    }
}

```

```

        break;
    }
    else if(randomnumber<n){
        printf("the value is high\n");
    }
    else if(randomnumber>n){
        printf("the value is low\n");
    }
    count = count -1;

}
else{
    printf("the invalid input");
}
if(count == 0){
    printf("sorry you lost");
    printf("the random number is %d",randomnumber);
    break;
}
}
}
}

```

OUTPUT

```

-----
enter the number1
you have 4 tries left
the value is low
enter the number2
you have 3 tries left
the value is low
enter the number3
you have 2 tries left
you win the game

```

2.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).
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>
#include<string.h>
int main(){
    char name[100];
    char accno[100];
    float amountbalance =0.0;
    int op;
    char accnew[100];
    float balance;
    float debit;

    while(1)
    {
        printf("THE BANK");
        printf("1. create account\n");
        printf("1. deposit account\n");
        printf("1. withdraw account\n");
        printf("1. exit account\n");
        printf("enter the option\n");
        scanf("%d",&op);

        switch (op)
        {
            case 1:
                printf("enter the name");
                scanf("%s",name);
                printf("enter the account number");
                scanf("%d",&accno);
                printf("you have success fully created the account in this bank");
                break;
            case 2:

                printf("enter the account number");
                scanf("%d",&accnew);

                if(strcmp(accnew,accno)==0){
                    printf("enter the amount");
                    scanf("%f",&balance);
                    amountbalance = amountbalance + balance;
                    printf("the current balance is %f",amountbalance);
                }
            }
        }
    }

```

```

    }
    else{
        printf("the account is an incorrect in the bank");
    }
    break;
case 3 :
    printf("enter the account number");
    scanf("%d",&accnew);

    if(strcmp(accnew,accno)==0){
        printf("enter the amount which is to be debited");
        scanf("%f",&debit);
        amountbalance = amountbalance - debit;

        printf("the current balance is %2f",amountbalance);

    }
    else{
        printf("the account number is in current");
    }

break;

case 4 :
    printf("enter the account number");
    scanf("%d",&accnew);

    if(strcmp(accnew,accno)==0){
        printf("the current balance is %2f",amountbalance);

    }
    else{
        printf("the account number is in current");

    }

    break;
case 5:
    printf("exit the statement");
    return 0;
    break;

default:
    break;
}

```

```
}  
}
```

OUTPUT

```
#include<stdio.h>  
#include<string.h>  
int main(){  
    char name[100];  
    char accno[100];  
    float amountbalance =0.0;  
    int op;  
    char accnew[100];  
    float balance;  
    float debit;  
  
    while(1)  
    {  
        printf("THE BANK\n");  
        printf("1. create account\n");  
        printf("2. deposit account\n");  
        printf("3. withdraw account\n");  
        printf("3. balance account\n");  
        printf("5. exit account\n");  
        printf("enter the option\n");  
        scanf("%d",&op);  
  
        switch (op)  
        {  
            case 1:  
                printf("enter the name ");  
                scanf("%s",name);  
                printf("enter the account number");  
                scanf("%d",&accno);  
                printf("you have success fully created the account in this bank");  
                break;  
            case 2:  
  
                printf("enter the account number");  
                scanf("%d",&accnew);  
  
                if(strcmp(accnew,accno)==0){
```

```

printf("enter the amount");
scanf("%f",&balance);
amountbalance = amountbalance + balance;
printf("the current balance is %f",amountbalance);

}
else{
    printf("the account is an incorrect in the bank");
}
break;
case 3 :
    printf("enter the account number");
    scanf("%d",&accnew);

    if(strcmp(accnew,accno)==0){
        printf("enter the amount which is to be debited");
        scanf("%f",&debit);
        amountbalance = amountbalance - debit;

        printf("the current balance is %2f",amountbalance);

    }
    else{
        printf("the account number is in current");
    }

break;

case 4 :
    printf("enter the account number");
    scanf("%d",&accnew);

    if(strcmp(accnew,accno)==0){
        printf("the current balance is %2f",amountbalance);

    }
    else{
        printf("the account number is in current");

    }

    break;
case 5:
    printf("exit the statement");
    return 0;

```



```

        break;

    default:
        break;
    }

}
}

```

OUTPUT

THE BANK

```

1. create account
2. deposit account
3. withdraw account
3. balance account
5. exit account
enter the option

```

1

enter the name anusree

enter the account number1234

you have success fully created the account in this bankTHE BANK

```

1. create account
2. deposit account
3. withdraw account
3. balance account
5. exit account
enter the option

```

2

enter the account number1234

enter the amount2000

the current balance is 2000.000000THE BANK

```

1. create account
2. deposit account
3. withdraw account
3. balance account
5. exit account
enter the option

```

3

enter the account number1234

enter the amount which is to be debited200

the current balance is 1800.000000THE BANK

```

1. create account

```

2. deposit account
3. withdraw account
3. balance account
5. exit account
enter the option
4
enter the account number1234
the current balance is 1800.000000THE BANK
1. create account
2. deposit account
3. withdraw account
3. balance account
5. exit account
enter the option
5
exit the statement

3

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 DAYS_IN_MONTH 30
```

```
int main() {
```

```
    float temperatures[DAYS_IN_MONTH];
```

```
    float sum = 0.0, averageTemp, highestTemp, lowestTemp;
```

```
    int aboveAverageDays = 0;
```

```
    printf("Enter temperatures for each day of the month (%d days):\n", DAYS_IN_MONTH);
```

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

```
        printf("Day %d temperature: ", i + 1);
```

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

```
        sum += temperatures[i];
```

```
    }
```

```
    averageTemp = sum / DAYS_IN_MONTH;
```

```

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

for (int i = 0; i < DAYS_IN_MONTH; i++) {
    if (temperatures[i] > highestTemp) {
        highestTemp = temperatures[i];
    }
    if (temperatures[i] < lowestTemp) {
        lowestTemp = temperatures[i];
    }
    if (temperatures[i] > averageTemp) {
        aboveAverageDays++;
    }
}

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

return 0;
}

```

4.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>
#include <string.h>
```

```

int main() {
    int op;
    char name[50];
    int quality;
    float price;
    char new[50];

    while(1) {
        printf("Inventory Management System\n");
        printf("1. Add product\n");
        printf("2. Update product\n");
        printf("3. Delete product\n");
        printf("4. View product\n");
        printf("5. Exit\n");
        printf("Enter the choice: ");
        scanf("%d", &op);

        switch (op) {
            case 1:
                printf("Enter the product name: ");
                scanf("%s", name);
                printf("Enter the product quantity: ");
                scanf("%d", &quality);
                printf("Enter the price of the product: ");
                scanf("%f", &price);
                printf("The product has been entered.\n");
                break;

            case 2:
                printf("Enter the product name to update: ");
                scanf("%s", new);
                if (strcmp(name, new) == 0) {
                    printf("Product found. Enter the new quantity: ");
                    scanf("%d", &quality);
                    printf("Enter the new price of the product: ");
                    scanf("%f", &price);
                    printf("The product has been updated.\n");
                } else {
                    printf("Product not found.\n");
                }
                break;

            case 3:
                printf("Enter the product name to delete: ");

```

```

scanf("%s", new);
if (strcmp(name, new) == 0) {
    name[0]='\0';
    quality=0;
    price = 0;
    printf("the product has been deleted");
}else{
    printf("Product not found.\n");
}
break;

case 4:
    printf("Product name: %s\n", name);
    printf("Product quantity: %d\n", quality);
    printf("Product price: %.2f\n", price);
    break;

case 5:
    printf("Exiting the program.\n");
    return 0;

default:
    printf("Invalid choice. Please try again.\n");
}
}

return 0;
}

```

CLASS WORKS

1.

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

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

```

OUTPUT

```
a=1
a=2
a=3
a=4
a=5
a=6
a=7
a=8
a=9
a=10
```

2 WAP to print the number between using do while

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

    int i = 1;
    do{
        printf("%d\n",i);

        i++;
    }
    while(i<=10);
}
```

OUTPUT

```
1
2
3
4
5
6
7
8
9
10
```

3.

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

```
int main() {
```



```

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

return 0;
}

```

OUTPUT

```

1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9
1 x 10 = 10
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9

```

$3 \times 4 = 12$
 $3 \times 5 = 15$
 $3 \times 6 = 18$
 $3 \times 7 = 21$
 $3 \times 8 = 24$
 $3 \times 9 = 27$
 $3 \times 10 = 30$
 $4 \times 1 = 4$
 $4 \times 2 = 8$
 $4 \times 3 = 12$
 $4 \times 4 = 16$
 $4 \times 5 = 20$
 $4 \times 6 = 24$
 $4 \times 7 = 28$
 $4 \times 8 = 32$
 $4 \times 9 = 36$
 $4 \times 10 = 40$
 $5 \times 1 = 5$
 $5 \times 2 = 10$
 $5 \times 3 = 15$
 $5 \times 4 = 20$
 $5 \times 5 = 25$
 $5 \times 6 = 30$
 $5 \times 7 = 35$
 $5 \times 8 = 40$
 $5 \times 9 = 45$
 $5 \times 10 = 50$
 $6 \times 1 = 6$
 $6 \times 2 = 12$
 $6 \times 3 = 18$
 $6 \times 4 = 24$
 $6 \times 5 = 30$
 $6 \times 6 = 36$
 $6 \times 7 = 42$
 $6 \times 8 = 48$
 $6 \times 9 = 54$
 $6 \times 10 = 60$
 $7 \times 1 = 7$
 $7 \times 2 = 14$
 $7 \times 3 = 21$
 $7 \times 4 = 28$
 $7 \times 5 = 35$
 $7 \times 6 = 42$
 $7 \times 7 = 49$

7 x 8 = 56
7 x 9 = 63
7 x 10 = 70
8 x 1 = 8
8 x 2 = 16
8 x 3 = 24
8 x 4 = 32
8 x 5 = 40
8 x 6 = 48
8 x 7 = 56
8 x 8 = 64
8 x 9 = 72
8 x 10 = 80
9 x 1 = 9
9 x 2 = 18
9 x 3 = 27
9 x 4 = 36
9 x 5 = 45
9 x 6 = 54
9 x 7 = 63
9 x 8 = 72
9 x 9 = 81
9 x 10 = 90
10 x 1 = 10
10 x 2 = 20
10 x 3 = 30
10 x 4 = 40
10 x 5 = 50
10 x 6 = 60
10 x 7 = 70
10 x 8 = 80
10 x 9 = 90
10 x 10 = 100

4.

// w a p to print the 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 X %d = %d\n",i,j ,i*j);
```

```

        j++;
    }
    i++;
}
}

```

5.

```

#include<stdio.h>
int main(){
    int row = 5,i=1,j;
    while(i<=row){
        j=1;
        while(j<=i){
            printf(" ");
            j++;
        }
        printf("\n");
        i++;
    }
}

```

OUTPUT

```

-----
*
* *
* * *
* * * *
* * * * *

```

6.

```

#include<stdio.h>
int main()
{
    int row = 4,i=1,j;
    while(i<=row){
        int space = row -i ;

        while(space>0){
            printf(" ");
            space--;
        }
    }
}

```

```

    }
    j=1;
    while(j<=i){
        printf(" *");
        j++;
    }
    printf("\n");
    i++;
}

}

```

OUTPUT

```

-----
      *
     **
    ***

```

7.

```

#include<stdio.h>
int main(){
    int i;
    for(i = 1; i<=10;i++)
    {
        printf("%d\n",i);
    }
}

```

8.

```

#include<stdio.h>
int main(){
    int n,sum =0;
    printf("enter the natural number");
    scanf("%d",&n);
    for(int i =1;i<=n;i++){
        sum = sum + i;
    }
    printf("the sum of natural numbers are%d",sum);
    return 0;
}

```

9.

```

#include<stdio.h>
int main(){
    int n,r,reverse=0;
    printf("enter the number to be reversed");
    scanf("%d",&n);
    for(;n!=0;){
        r = n%10;
        reverse = reverse * 10 +r;
        n = n/10;
    }
    printf("the reverse of a number is %d",reverse);
}

```

OUTPUT

1234

4321

10.

```

#include<stdio.h>
int main(){
    int first ,second, next,n;
    first =0;
    second =1;
    printf("enter the number");
    scanf("%d",&n);
    printf("%d\n",first);
    for(int i = 0;i< n; i++){

        if(second>=n){
            break;
        }
        else{
            printf("%d\n",second);
            next = first + second;
            first = second;
            second= next;
        }

    }
}

```

OUTPUT

0
1
1
2
3
5
8

11.

```
#include<stdio.h>
int main(){
    int i;
    for(i=0;i<=10;i++){
        if(i==5){
            break;
        }
        printf("%d\n",i);
    }
}
```

12.

```
#include<stdio.h>
int main(){
    int i;
    for(i=0;i<=10;i++){
        if(i==5){
            continue;
        }
    }
}
```

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

13.

```
#include <stdio.h>  
  
int main() {  
    int n, sum = 0;  
  
    printf("enter up to 20 integers(enter -1 to stop)");  
    for(int i =0;i<=20;i++){  
        scanf("%d",&n);  
        if (n==-1)  
        {  
            break;  
        }  
        else if (n<0)  
        {  
            continue;  
        }  
        else if (n%2==0)  
        {  
            sum = sum+n;  
        }  
    }  
    printf("the sum of even numbers are =%d",sum);  
    return 0;  
}
```

OUTPUT

1
2
3
4
5

6
7
-1
2
the sum of even numbers are 14

14.

ASSIGNMENTS

1.

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

```
int main() {
    int n, count = 4;
    srand(time(NULL));
    int randomnumber = rand() % 10;

    while(1){
        printf("enter the number");
        scanf("%d",&n);
        if(n<20 && n>0){
            printf("you have %d tries left\n",count);
            if(n == randomnumber){
                printf("you win the game\n");
                break;
            }
            else if(randomnumber<n){
                printf("the value is high\n");
            }
            else if(randomnumber>n){
                printf("the value is low\n");
            }
            count = count -1;
        }
        else{
            printf("the invalid input");
        }
    }
}
```

```

        if(count == 0){
            printf("sorry you lost");
            printf("the random number is %d",randomnumber);
            break;
        }
    }

}

```

OUTPUT

```

-----
enter the number1
you have 4 tries left
the value is low
enter the number2
you have 3 tries left
the value is low
enter the number3
you have 2 tries left
you win the game

```

2.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).

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>
#include<string.h>
int main(){
    char name[100];
    char accno[100];
    float amountbalance =0.0;
    int op;
    char accnew[100];
    float balance;
    float debit;
```

```

while(1)
{
printf("THE BANK");
printf("1. create account\n");
printf("1. deposit account\n");
printf("1. withdraw account\n");
printf("1. exit account\n");
printf("enter the option\n");
scanf("%d",&op);

switch (op)
{
case 1:
    printf("enter the name");
    scanf("%s",name);
    printf("enter the account number");
    scanf("%d",&accno);
    printf("you have success fully created the account in this bank");
    break;
case 2:

    printf("enter the account number");
    scanf("%d",&accnew);

    if(strcmp(accnew,accno)==0){
        printf("enter the amount");
        scanf("%f",&balance);
        amountbalance = amountbalance + balance;
        printf("the current balance is %f",amountbalance);

    }
    else{
        printf("the account is an incorrect in the bank");
    }
    break;
case 3 :
    printf("enter the account number");
    scanf("%d",&accnew);

    if(strcmp(accnew,accno)==0){
        printf("enter the amount which is to be debited");
        scanf("%f",&debit);
        amountbalance = amountbalance - debit;
    }
    else{
        printf("the account is an incorrect in the bank");
    }
    break;
}
}

```

```

        printf("the current balance is %2f",amountbalance);

    }
    else{
        printf("the account number is in current");
    }

break;

case 4 :
    printf("enter the account number");
    scanf("%d",&accnew);

    if(strcmp(accnew,accno)==0){
        printf("the current balance is %2f",amountbalance);

    }
    else{
        printf("the account number is in current");

    }
    break;
case 5:
    printf("exit the statement");
    return 0;
    break;

default:
    break;
}

}
}

```

OUTPUT

```

#include<stdio.h>
#include<string.h>
int main(){
    char name[100];
    char accno[100];
    float amountbalance =0.0;

```

```

int op;
char accnew[100];
float balance;
float debit;

while(1)
{
printf("THE BANK\n");
printf("1. create account\n");
printf("2. deposit account\n");
printf("3. withdraw account\n");
printf("3. balance account\n");
printf("5. exit account\n");
printf("enter the option\n");
scanf("%d",&op);

switch (op)
{
case 1:
    printf("enter the name ");
    scanf("%s",name);
    printf("enter the account number");
    scanf("%d",&accno);
    printf("you have success fully created the account in this bank");
    break;
case 2:

    printf("enter the account number");
    scanf("%d",&accnew);

    if(strcmp(accnew,accno)==0){
        printf("enter the amount");
        scanf("%f",&balance);
        amountbalance = amountbalance + balance;
        printf("the current balance is %f",amountbalance);

    }
    else{
        printf("the account is an incorrect in the bank");
    }
    break;
case 3 :
    printf("enter the account number");
    scanf("%d",&accnew);

```

```

        if(strcmp(accnew,accno)==0){
            printf("enter the amount which is to be debited");
            scanf("%f",&debit);
            amountbalance = amountbalance - debit;

            printf("the current balance is %2f",amountbalance);

        }
        else{
            printf("the account number is in current");
        }

break;

case 4 :
    printf("enter the account number");
    scanf("%d",&accnew);

    if(strcmp(accnew,accno)==0){
        printf("the current balance is %2f",amountbalance);

    }
    else{
        printf("the account number is in current");

    }
    break;
case 5:
    printf("exit the statement");
    return 0;
    break;

default:
    break;
}

}
}

```

OUTPUT

THE BANK

1. create account
2. deposit account
3. withdraw account
3. balance account
5. exit account

enter the option

1

enter the name anusree

enter the account number1234

you have success fully created the account in this bankTHE BANK

1. create account
2. deposit account
3. withdraw account
3. balance account
5. exit account

enter the option

2

enter the account number1234

enter the amount2000

the current balance is 2000.000000THE BANK

1. create account
2. deposit account
3. withdraw account
3. balance account
5. exit account

enter the option

3

enter the account number1234

enter the amount which is to be debited200

the current balance is 1800.000000THE BANK

1. create account
2. deposit account
3. withdraw account
3. balance account
5. exit account

enter the option

4

enter the account number1234

the current balance is 1800.000000THE BANK

1. create account
2. deposit account
3. withdraw account
3. balance account
5. exit account

enter the option

5

exit the statement

3

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 DAYS_IN_MONTH 30

int main() {
    float temperatures[DAYS_IN_MONTH];
    float sum = 0.0, averageTemp, highestTemp, lowestTemp;
    int aboveAverageDays = 0;

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

    averageTemp = sum / DAYS_IN_MONTH;

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

    for (int i = 0; i < DAYS_IN_MONTH; i++) {
        if (temperatures[i] > highestTemp) {
            highestTemp = temperatures[i];
        }
        if (temperatures[i] < lowestTemp) {
            lowestTemp = temperatures[i];
        }
        if (temperatures[i] > averageTemp) {
```

```

        aboveAverageDays++;
    }
}

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

return 0;
}

```

4.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>
#include <string.h>
```

```
int main() {
    int op;
    char name[50];
    int quality;
    float price;
    char new[50];

    while(1) {
        printf("Inventory Management System\n");
        printf("1. Add product\n");
        printf("2. Update product\n");
        printf("3. Delete product\n");
        printf("4. View product\n");
```

```
printf("5. Exit\n");
printf("Enter the choice: ");
scanf("%d", &op);
```

```
switch (op) {
    case 1:
        printf("Enter the product name: ");
        scanf("%s", name);
        printf("Enter the product quantity: ");
        scanf("%d", &quality);
        printf("Enter the price of the product: ");
        scanf("%f", &price);
        printf("The product has been entered.\n");
        break;

    case 2:
        printf("Enter the product name to update: ");
        scanf("%s", new);
        if (strcmp(name, new) == 0) {
            printf("Product found. Enter the new quantity: ");
            scanf("%d", &quality);
            printf("Enter the new price of the product: ");
            scanf("%f", &price);
            printf("The product has been updated.\n");
        } else {
            printf("Product not found.\n");
        }
        break;

    case 3:
        printf("Enter the product name to delete: ");
        scanf("%s", new);
        if (strcmp(name, new) == 0) {
            name[0]='\0';
            quality=0;
            price = 0;
            printf("the product hase been deleted");
        }else{
            printf("Product not found.\n");
        }
        break;

    case 4:
        printf("Product name: %s\n", name);
```

```
printf("Product quantity: %d\n", quality);  
printf("Product price: %.2f\n", price);  
break;
```

```
case 5:
```

```
printf("Exiting the program.\n");  
return 0;
```

```
default:
```

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

```
}
```

```
}
```

```
return 0;
```

```
}
```