NAME : RABARI MAHI ALPESHBHAI

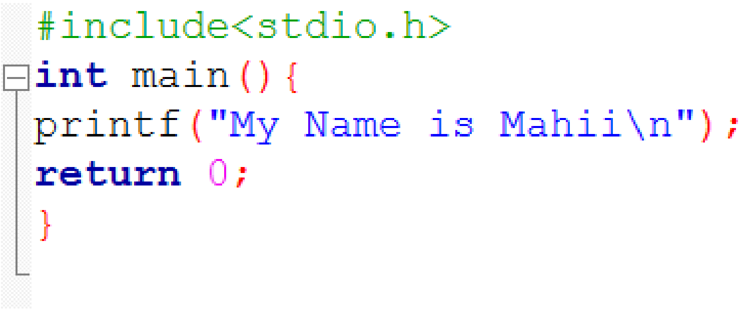
ID NO : 25CE095

CE : DIV 2

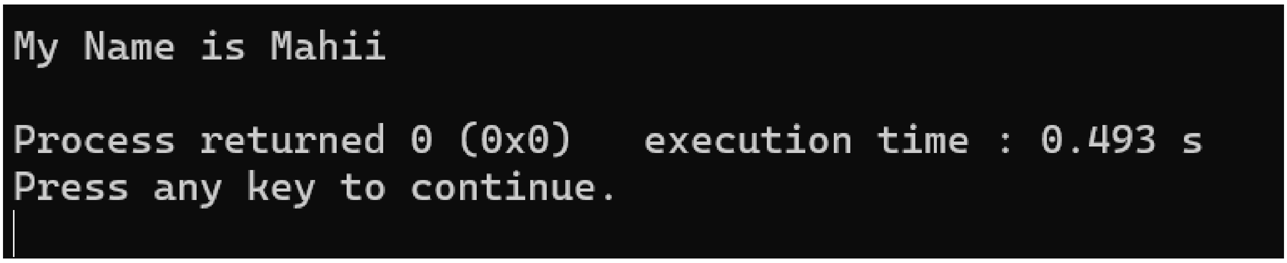
BATCH : A2

PRACTICAL 1 :

➔ **A simple C program that displays your name on the screen .**



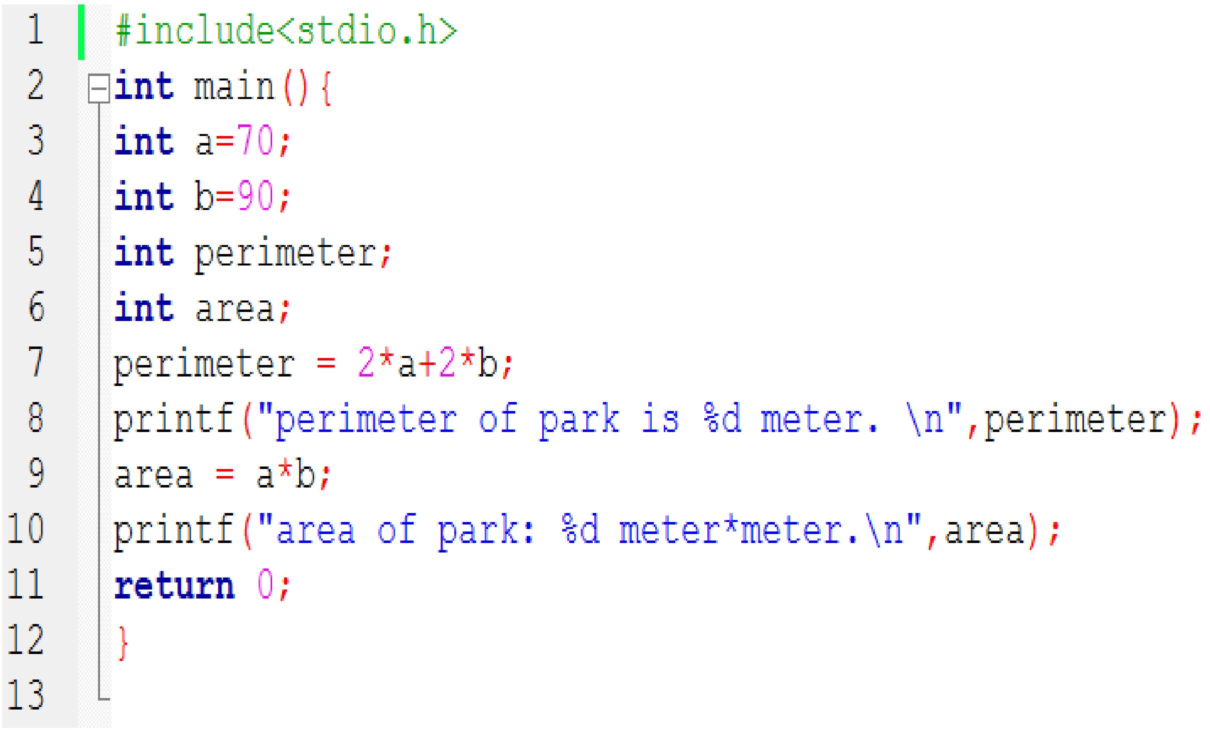
```c
#include<stdio.h>
int main(){
printf("My Name is Mahii\n");
return 0;
}
```



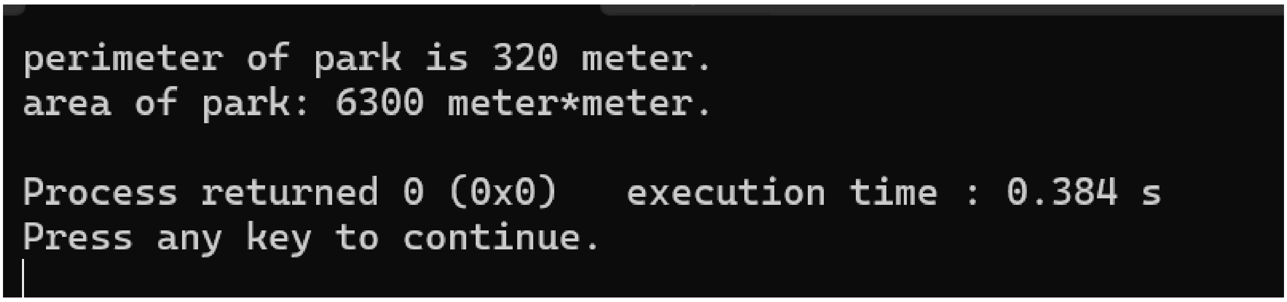
```
My Name is Mahii

Process returned 0 (0x0)   execution time : 0.493 s
Press any key to continue.
```

PRACTICAL 2:

➔ calculate the area and perimeter to finalize blueprint.



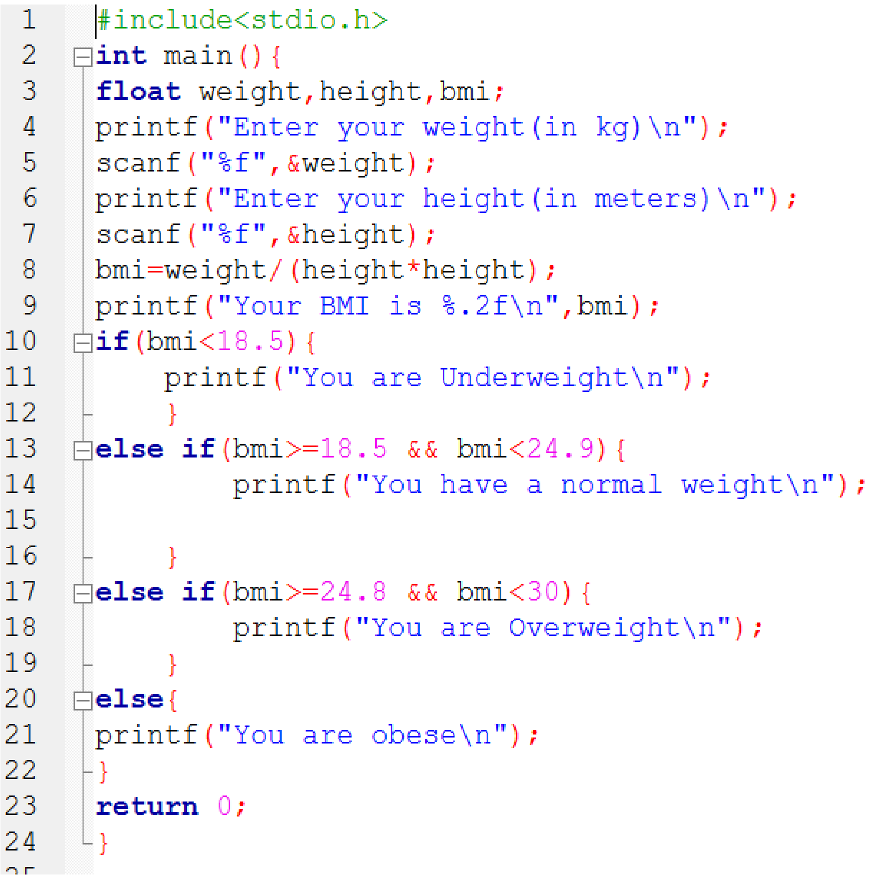
```c
#include<stdio.h>
int main(){
int a=70;
int b=90;
int perimeter;
int area;
perimeter = 2*a+2*b;
printf("perimeter of park is %d meter. \n",perimeter);
area = a*b;
printf("area of park: %d meter*meter.\n",area);
return 0;
}
```



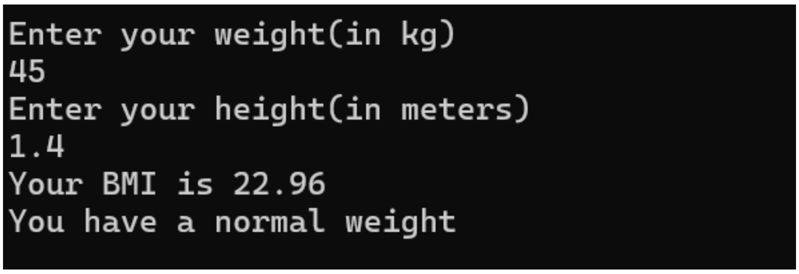
```
perimeter of park is 320 meter.
area of park: 6300 meter*meter.

Process returned 0 (0x0)   execution time : 0.384 s
Press any key to continue.
```

PRACTICAL 3:

➔ **Efficient tool to calculate BMI.**



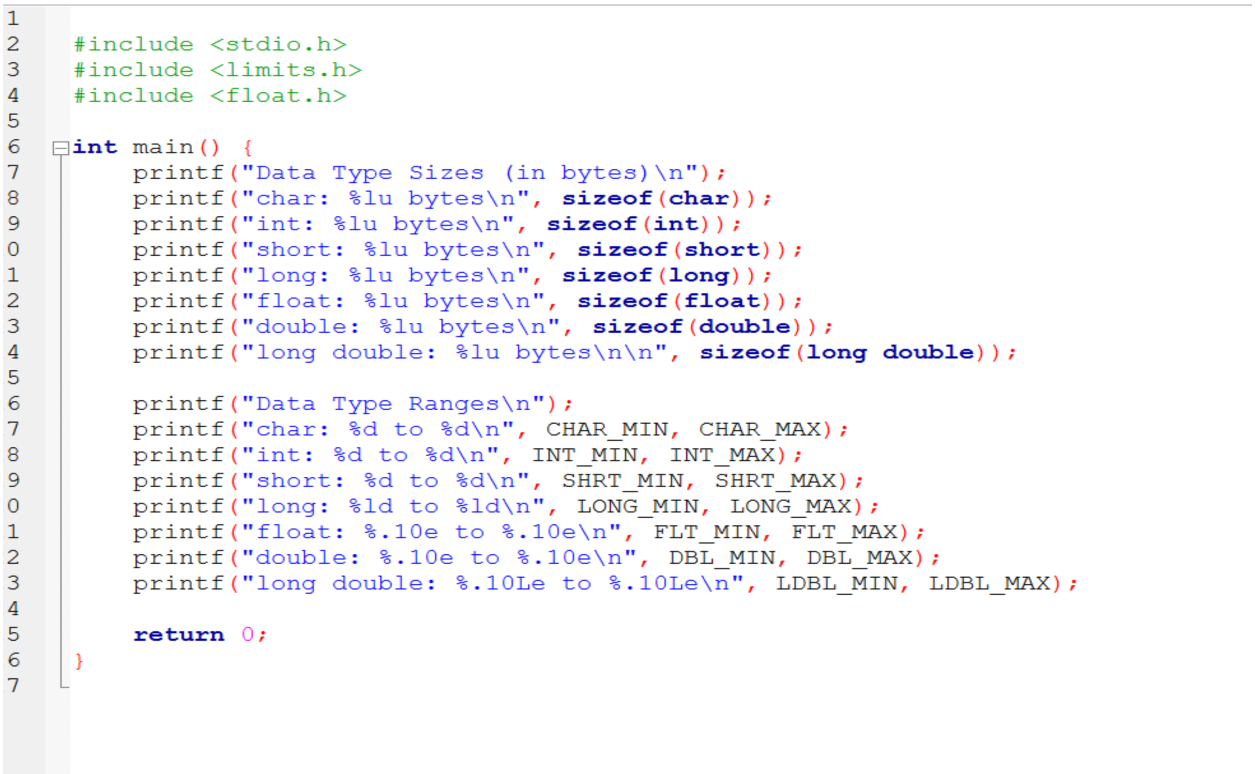
```c
#include<stdio.h>
int main(){
float weight,height,bmi;
printf("Enter your weight(in kg)\n");
scanf("%f",&weight);
printf("Enter your height(in meters)\n");
scanf("%f",&height);
bmi=weight/(height*height);
printf("Your BMI is %.2f\n",bmi);
if(bmi<18.5){
    printf("You are Underweight\n");
    }
else if(bmi>=18.5 && bmi<24.9){
        printf("You have a normal weight\n");

    }
else if(bmi>=24.8 && bmi<30){
        printf("You are Overweight\n");
    }
else{
printf("You are obese\n");
}
return 0;
}
```



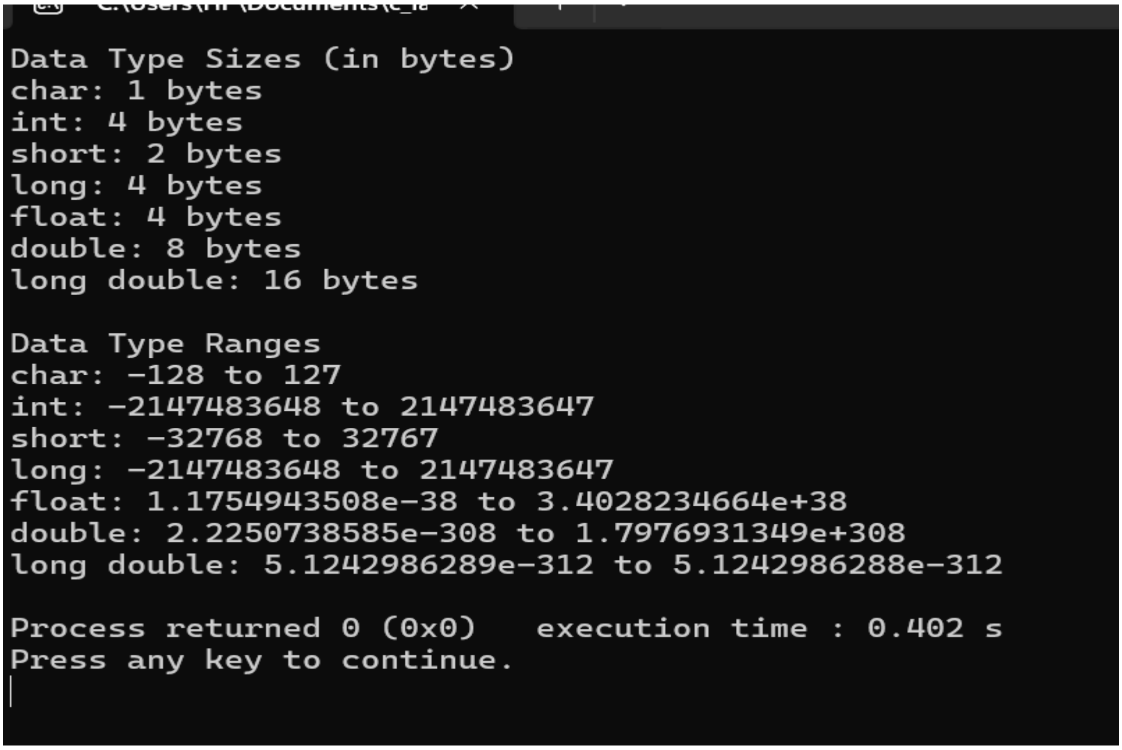
```
Enter your weight(in kg)
45
Enter your height(in meters)
1.4
Your BMI is 22.96
You have a normal weight
```

PRACTICAL 4:

➔ **data types and their ranges!**



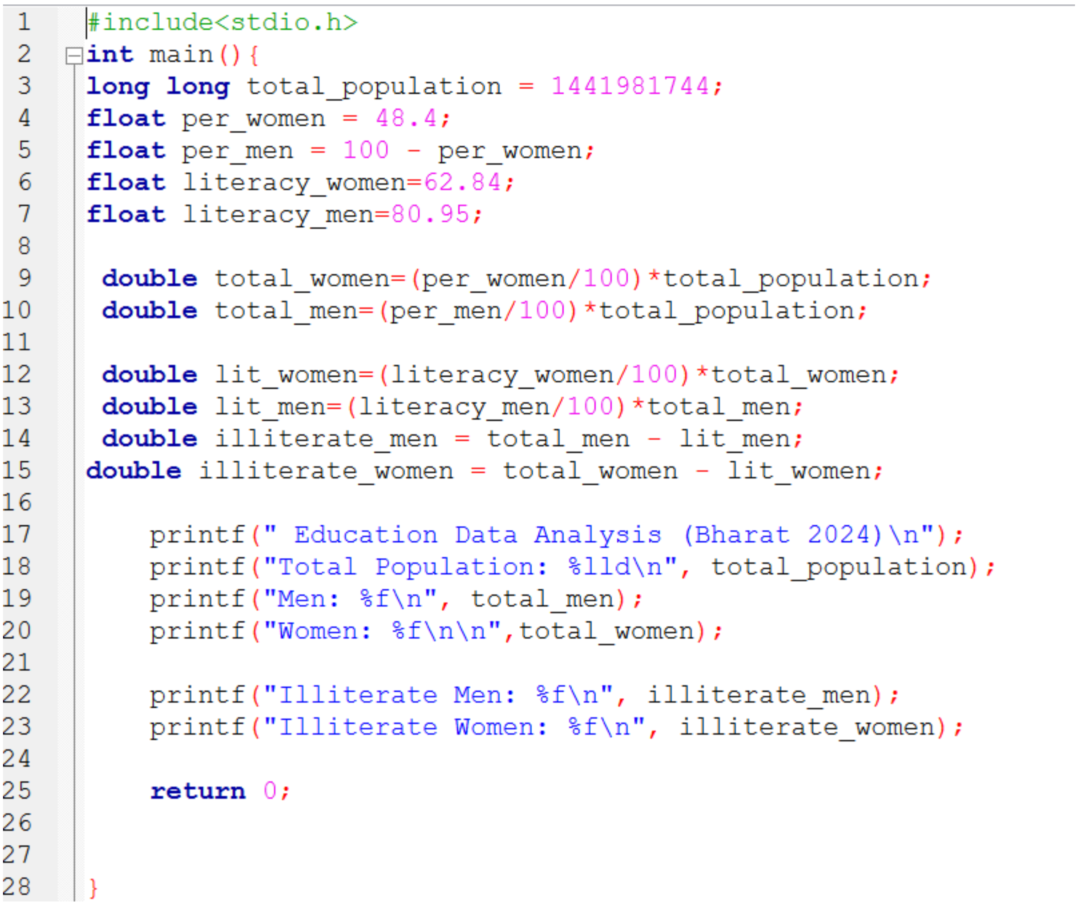
```c
#include <stdio.h>
#include <limits.h>
#include <float.h>

int main() {
    printf("Data Type Sizes (in bytes)\n");
    printf("char: %lu bytes\n", sizeof(char));
    printf("int: %lu bytes\n", sizeof(int));
    printf("short: %lu bytes\n", sizeof(short));
    printf("long: %lu bytes\n", sizeof(long));
    printf("float: %lu bytes\n", sizeof(float));
    printf("double: %lu bytes\n", sizeof(double));
    printf("long double: %lu bytes\n\n", sizeof(long double));

    printf("Data Type Ranges\n");
    printf("char: %d to %d\n", CHAR_MIN, CHAR_MAX);
    printf("int: %d to %d\n", INT_MIN, INT_MAX);
    printf("short: %d to %d\n", SHRT_MIN, SHRT_MAX);
    printf("long: %ld to %ld\n", LONG_MIN, LONG_MAX);
    printf("float: %.10e to %.10e\n", FLT_MIN, FLT_MAX);
    printf("double: %.10e to %.10e\n", DBL_MIN, DBL_MAX);
    printf("long double: %.10Le to %.10Le\n", LDBL_MIN, LDBL_MAX);

    return 0;
}
```



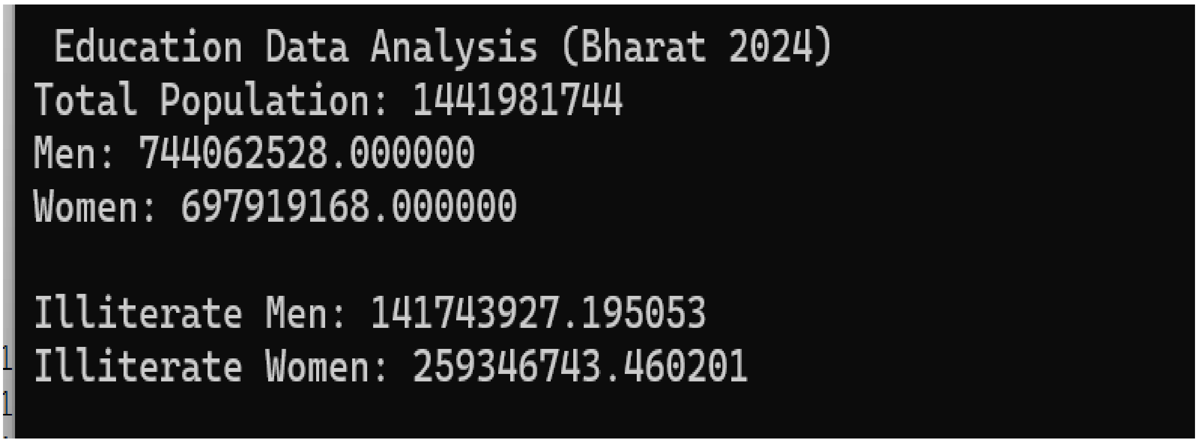
```
Data Type Sizes (in bytes)
char: 1 bytes
int: 4 bytes
short: 2 bytes
long: 4 bytes
float: 4 bytes
double: 8 bytes
long double: 16 bytes

Data Type Ranges
char: -128 to 127
int: -2147483648 to 2147483647
short: -32768 to 32767
long: -2147483648 to 2147483647
float: 1.1754943508e-38 to 3.4028234664e+38
double: 2.2250738585e-308 to 1.7976931349e+308
long double: 5.1242986289e-312 to 5.1242986288e-312

Process returned 0 (0x0)   execution time : 0.402 s
Press any key to continue.
```

PRACTICAL 5:

➔ **compute the count of illiterate men and women using these statistics.**



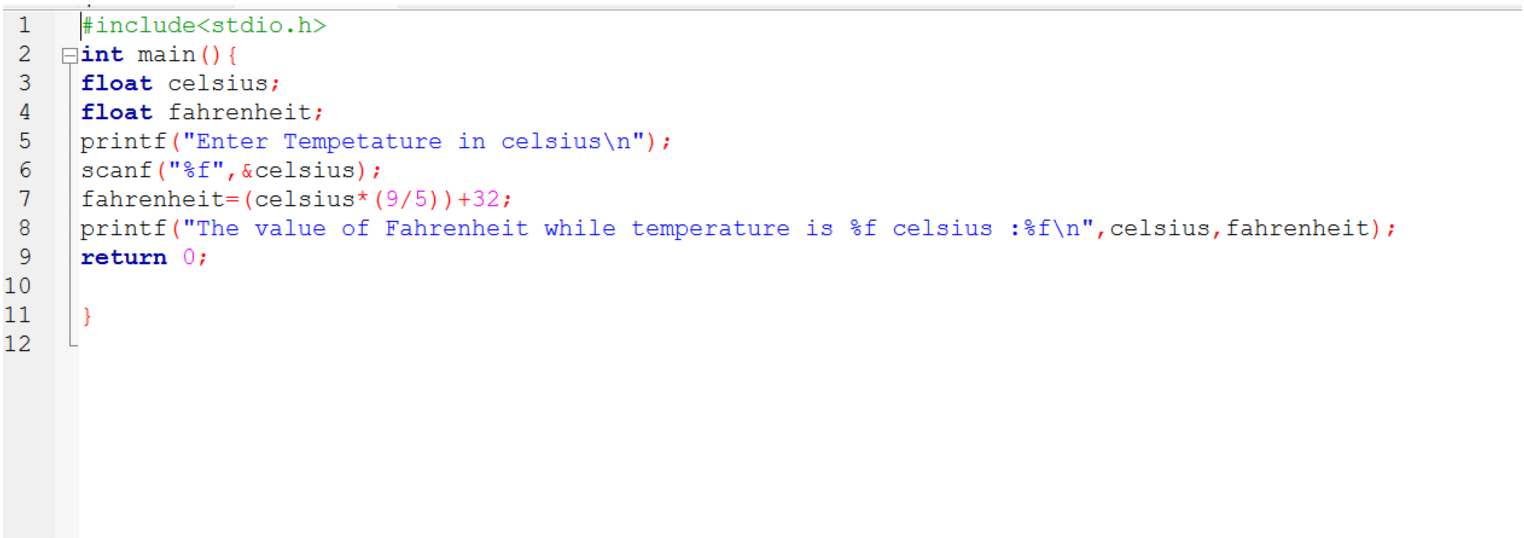
```c
#include<stdio.h>
int main(){
long long total_population = 1441981744;
float per_women = 48.4;
float per_men = 100 - per_women;
float literacy_women=62.84;
float literacy_men=80.95;

 double total_women=(per_women/100)*total_population;
 double total_men=(per_men/100)*total_population;

 double lit_women=(literacy_women/100)*total_women;
 double lit_men=(literacy_men/100)*total_men;
 double illiterate_men = total_men - lit_men;
double illiterate_women = total_women - lit_women;

    printf(" Education Data Analysis (Bharat 2024)\n");
    printf("Total Population: %lld\n", total_population);
    printf("Men: %f\n", total_men);
    printf("Women: %f\n\n",total_women);

    printf("Illiterate Men: %f\n", illiterate_men);
    printf("Illiterate Women: %f\n", illiterate_women);

    return 0;


}
```



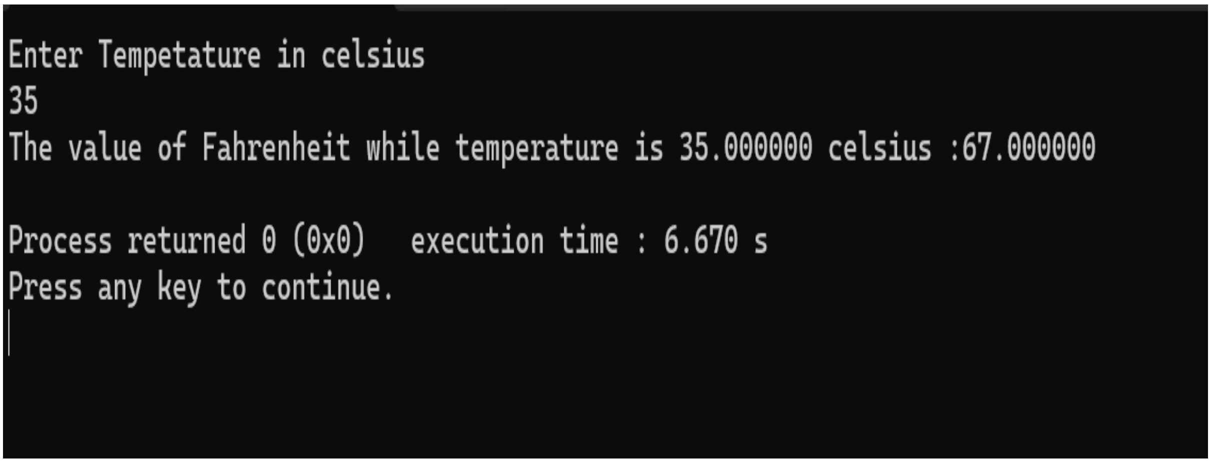
```
 Education Data Analysis (Bharat 2024)
Total Population: 1441981744
Men: 744062528.000000
Women: 697919168.000000

Illiterate Men: 141743927.195053
Illiterate Women: 259346743.460201
```

PRACTICAL 6:

➔ **A program to do conversions to Fahrenheit.**



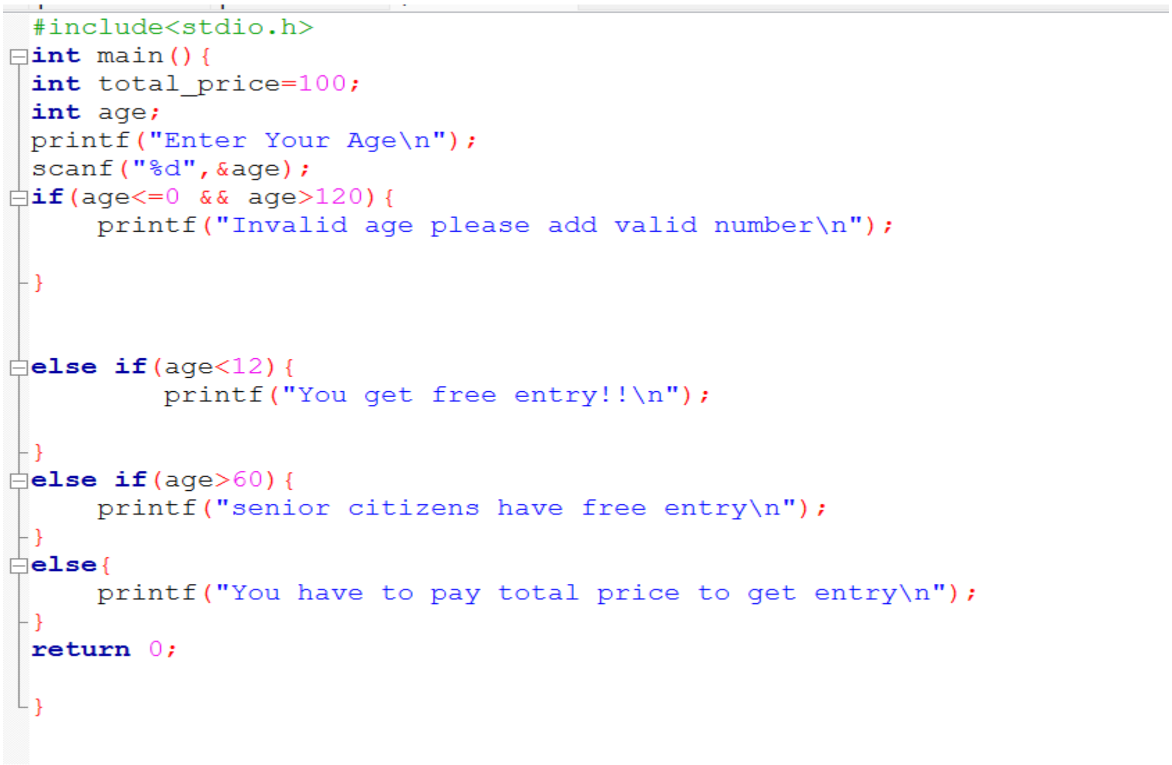
```c
#include<stdio.h>
int main(){
float celsius;
float fahrenheit;
printf("Enter Tempetature in celsius\n");
scanf("%f",&celsius);
fahrenheit=(celsius*(9/5))+32;
printf("The value of Fahrenheit while temperature is %f celsius :%f\n",celsius,fahrenheit);
return 0;

}
```



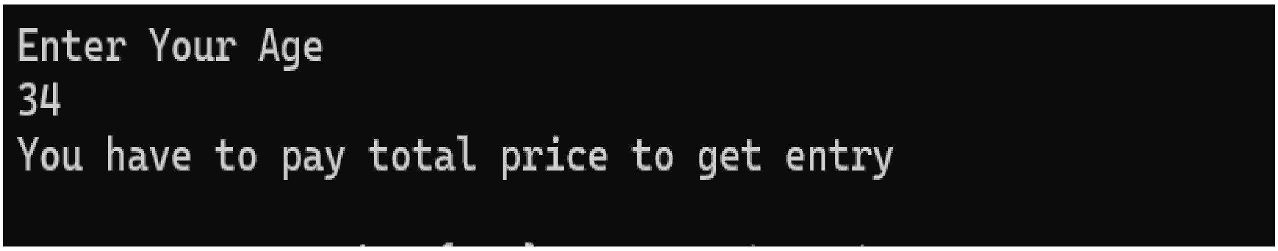
```
Enter Tempetature in celsius
35
The value of Fahrenheit while temperature is 35.000000 celsius :67.000000

Process returned 0 (0x0)   execution time : 6.670 s
Press any key to continue.
```

PRACTICAL 7:

➔ an automated ticketing system to reduce wait times at entry gates. The system needs to decide whether a visitor should be charged for entry or not.



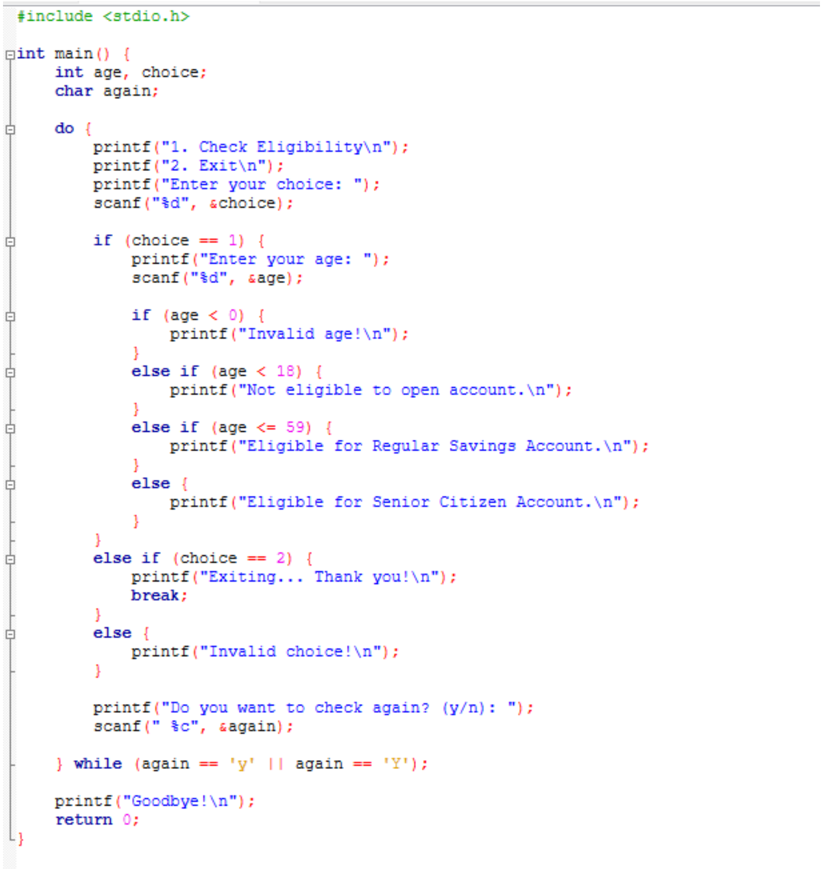
```c
#include<stdio.h>
int main(){
int total_price=100;
int age;
printf("Enter Your Age\n");
scanf("%d",&age);
if(age<=0 && age>120){
    printf("Invalid age please add valid number\n");

}


else if(age<12){
        printf("You get free entry!!\n");

}
else if(age>60){
    printf("senior citizens have free entry\n");
}
else{
    printf("You have to pay total price to get entry\n");
}
return 0;

}
```



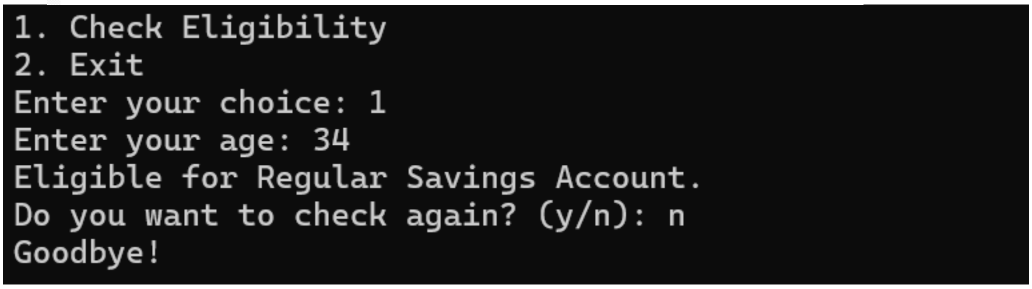
```
Enter Your Age
34
You have to pay total price to get entry
```

PRACTICAL 8:

➔ An eligibility checker at their branches.

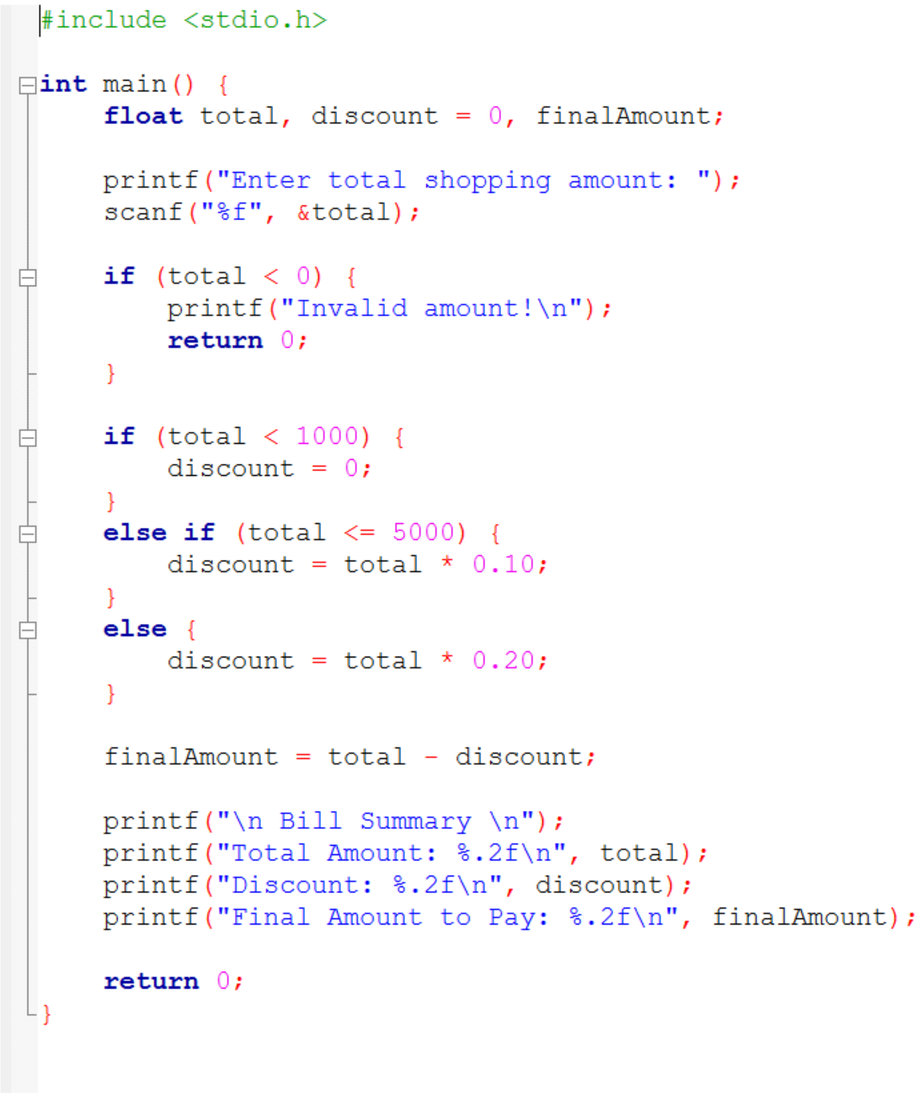


```c
#include <stdio.h>

int main() {
    int age, choice;
    char again;

    do {
        printf("1. Check Eligibility\n");
        printf("2. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        if (choice == 1) {
            printf("Enter your age: ");
            scanf("%d", &age);

            if (age < 0) {
                printf("Invalid age!\n");
            }
            else if (age < 18) {
                printf("Not eligible to open account.\n");
            }
            else if (age <= 59) {
                printf("Eligible for Regular Savings Account.\n");
            }
            else {
                printf("Eligible for Senior Citizen Account.\n");
            }
        }
        else if (choice == 2) {
            printf("Exiting... Thank you!\n");
            break;
        }
        else {
            printf("Invalid choice!\n");
        }

        printf("Do you want to check again? (y/n): ");
        scanf(" %c", &again);

    } while (again == 'y' || again == 'Y');

    printf("Goodbye!\n");
    return 0;
}
```

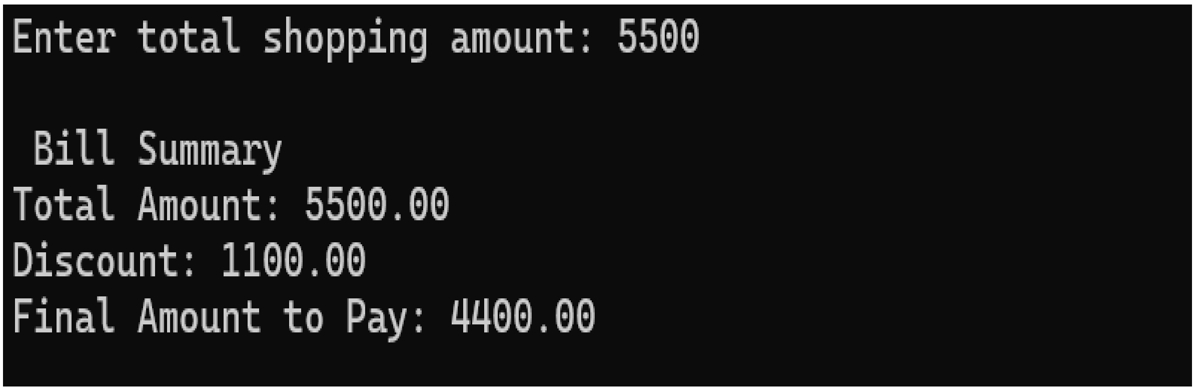


```
1. Check Eligibility
2. Exit
Enter your choice: 1
Enter your age: 34
Eligible for Regular Savings Account.
Do you want to check again? (y/n): n
Goodbye!
```

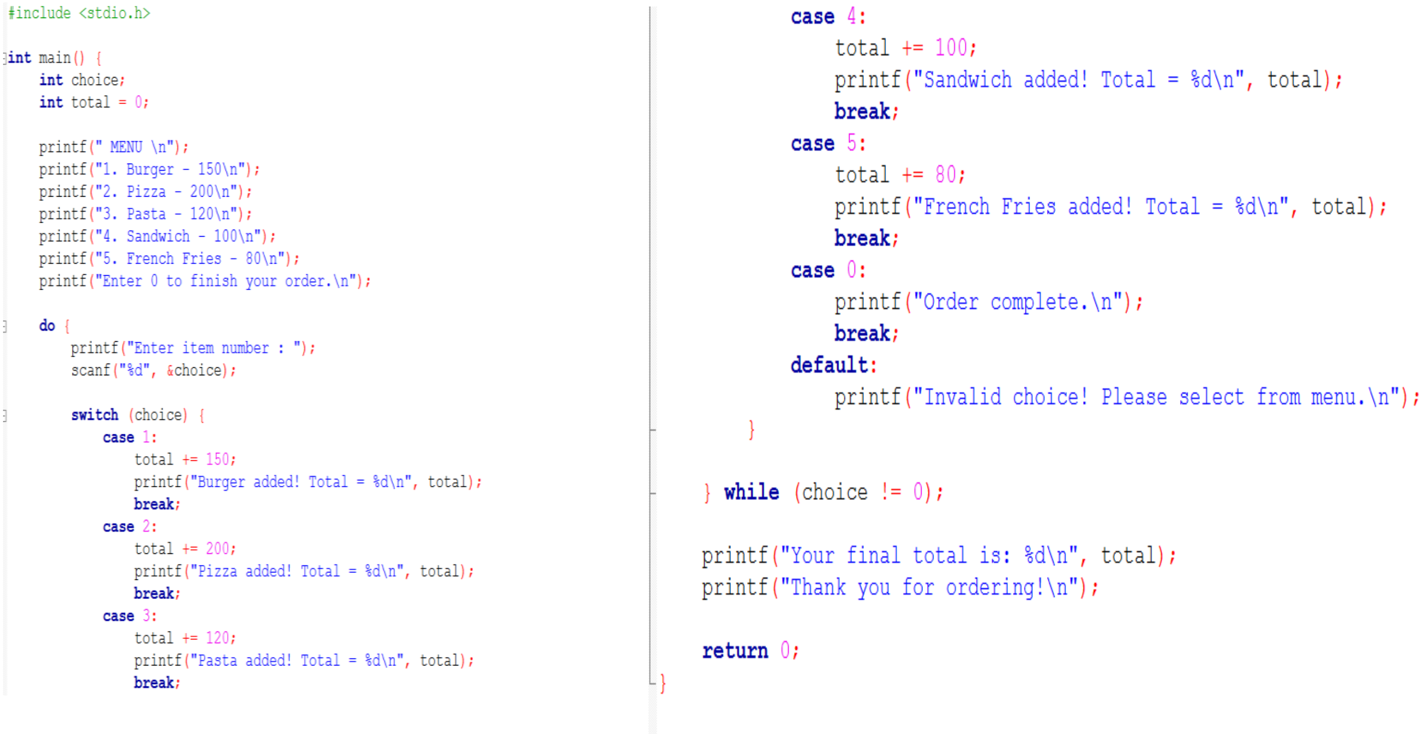
PRACTICAL 9: A budding developer, with designing the logic for the discount system.



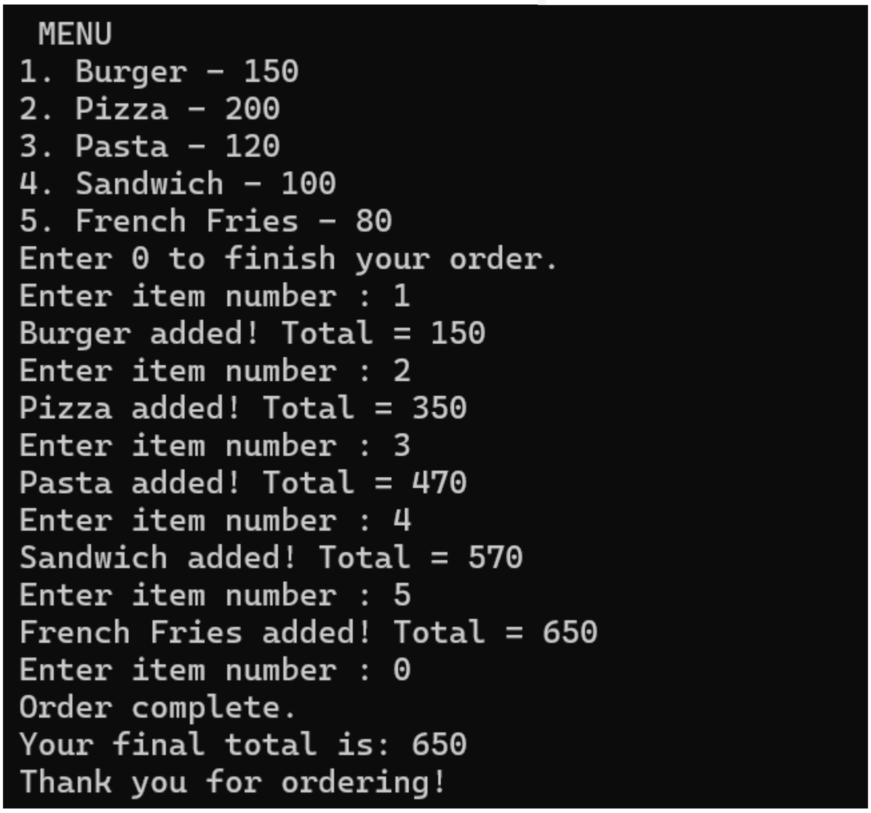
```c
#include <stdio.h>

int main() {
    float total, discount = 0, finalAmount;

    printf("Enter total shopping amount: ");
    scanf("%f", &total);

    if (total < 0) {
        printf("Invalid amount!\n");
        return 0;
    }

    if (total < 1000) {
        discount = 0;
    }
    else if (total <= 5000) {
        discount = total * 0.10;
    }
    else {
        discount = total * 0.20;
    }

    finalAmount = total - discount;

    printf("\n Bill Summary \n");
    printf("Total Amount: %.2f\n", total);
    printf("Discount: %.2f\n", discount);
    printf("Final Amount to Pay: %.2f\n", finalAmount);

    return 0;
}
```



```
Enter total shopping amount: 5500

 Bill Summary
Total Amount: 5500.00
Discount: 1100.00
Final Amount to Pay: 4400.00
```



```c
#include <stdio.h>

int main() {
    int choice;
    int total = 0;

    printf(" MENU \n");
    printf("1. Burger - 150\n");
    printf("2. Pizza - 200\n");
    printf("3. Pasta - 120\n");
    printf("4. Sandwich - 100\n");
    printf("5. French Fries - 80\n");
    printf("Enter 0 to finish your order.\n");

    do {
        printf("Enter item number : ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                total += 150;
                printf("Burger added! Total = %d\n", total);
                break;
            case 2:
                total += 200;
                printf("Pizza added! Total = %d\n", total);
                break;
            case 3:
                total += 120;
                printf("Pasta added! Total = %d\n", total);
                break;
            case 4:
                total += 100;
                printf("Sandwich added! Total = %d\n", total);
                break;
            case 5:
                total += 80;
                printf("French Fries added! Total = %d\n", total);
                break;
            case 0:
                printf("Order complete.\n");
                break;
            default:
                printf("Invalid choice! Please select from menu.\n");
        }

    } while (choice != 0);

    printf("Your final total is: %d\n", total);
    printf("Thank you for ordering!\n");

    return 0;
}
```

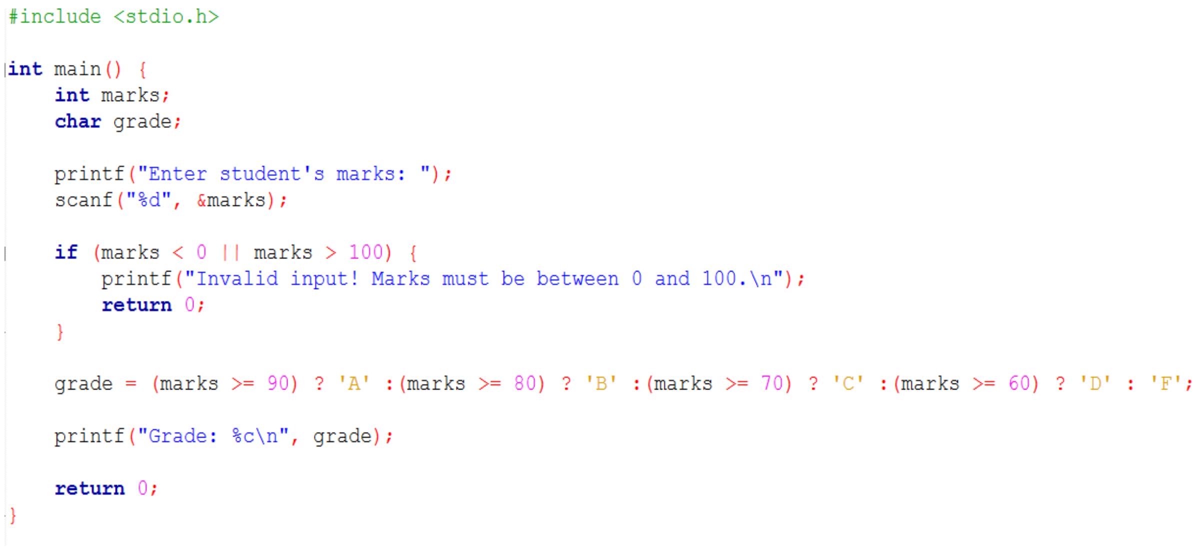


```
 MENU
1. Burger - 150
2. Pizza - 200
3. Pasta - 120
4. Sandwich - 100
5. French Fries - 80
Enter 0 to finish your order.
Enter item number : 1
Burger added! Total = 150
Enter item number : 2
Pizza added! Total = 350
Enter item number : 3
Pasta added! Total = 470
Enter item number : 4
Sandwich added! Total = 570
Enter item number : 5
French Fries added! Total = 650
Enter item number : 0
Order complete.
Your final total is: 650
Thank you for ordering!
```

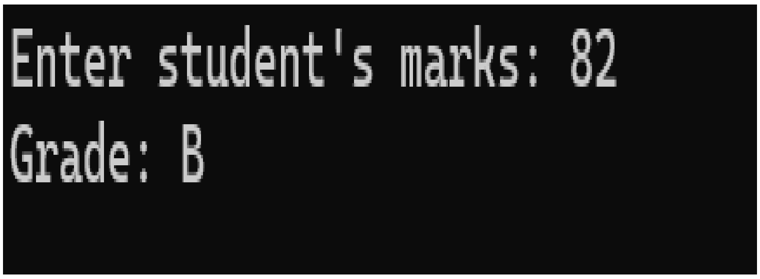
PRACTICAL 10 :

➔ MENU USING SWITCH CASE

PRACTICAL 11 : Take student’s marks as input and display the corresponding grade as output according to the following criteria:

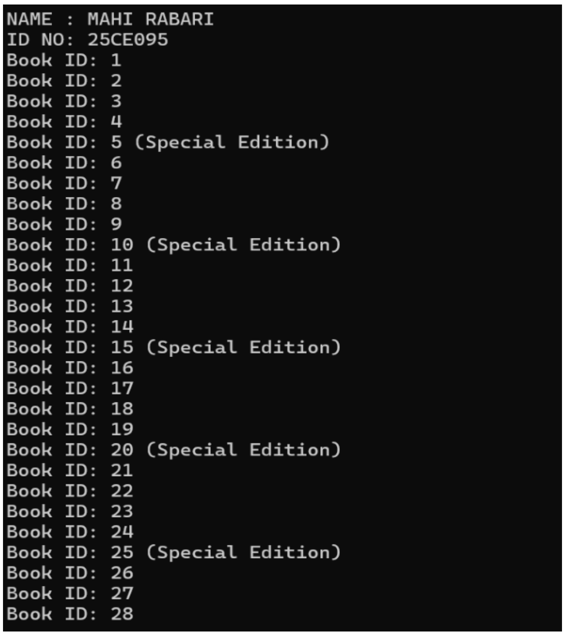


```c
#include <stdio.h>

int main() {
    int marks;
    char grade;

    printf("Enter student's marks: ");
    scanf("%d", &marks);

    if (marks < 0 || marks > 100) {
        printf("Invalid input! Marks must be between 0 and 100.\n");
        return 0;
    }

    grade = (marks >= 90) ? 'A' :(marks >= 80) ? 'B' :(marks >= 70) ? 'C' :(marks >= 60) ? 'D' : 'F';

    printf("Grade: %c\n", grade);

    return 0;
}
```

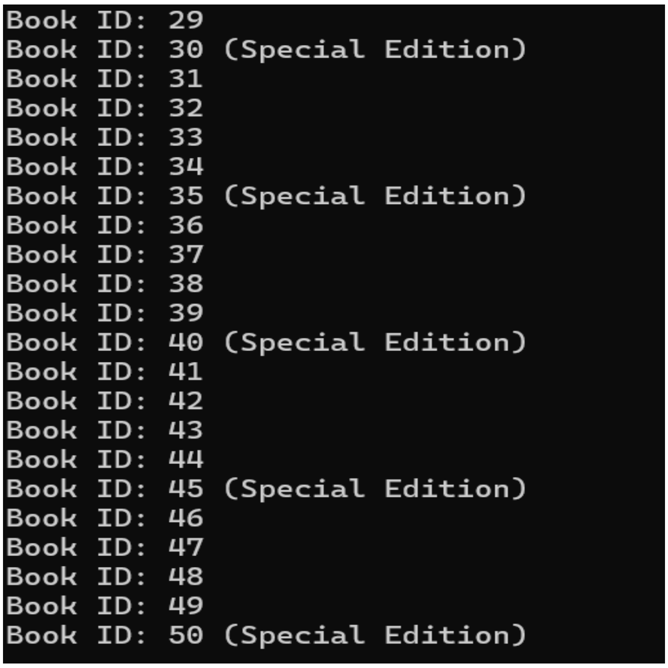


```
Enter student's marks: 82
Grade: B
```

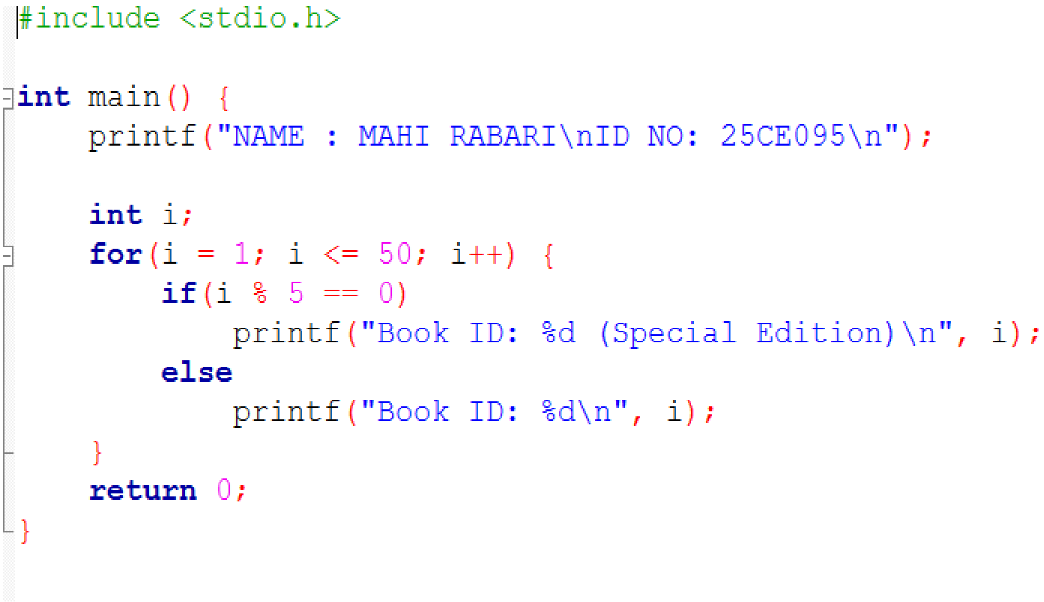
PRACTICAL 12: Displays each book ID on a new line and highlights special editions with a clear note next to their ID.



```
NAME : MAHI RABARI
ID NO: 25CE095
Book ID: 1
Book ID: 2
Book ID: 3
Book ID: 4
Book ID: 5 (Special Edition)
Book ID: 6
Book ID: 7
Book ID: 8
Book ID: 9
Book ID: 10 (Special Edition)
Book ID: 11
Book ID: 12
Book ID: 13
Book ID: 14
Book ID: 15 (Special Edition)
Book ID: 16
Book ID: 17
Book ID: 18
Book ID: 19
Book ID: 20 (Special Edition)
Book ID: 21
Book ID: 22
Book ID: 23
Book ID: 24
Book ID: 25 (Special Edition)
Book ID: 26
Book ID: 27
Book ID: 28
```

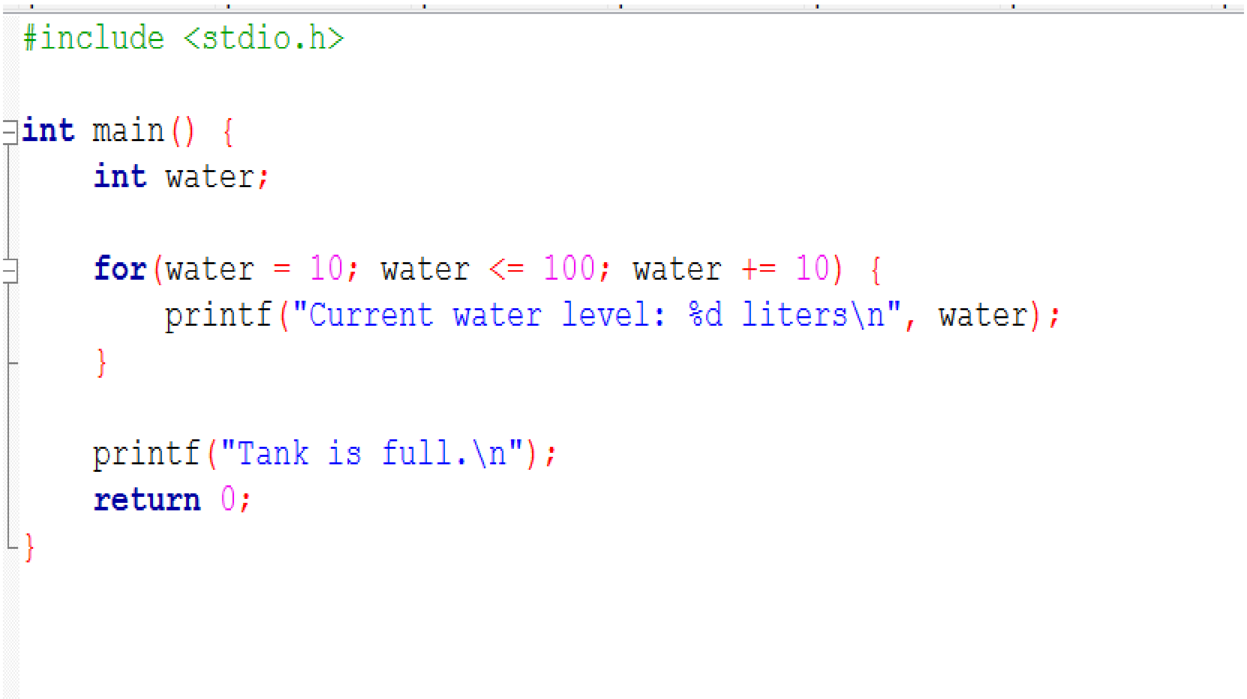


```
Book ID: 29
Book ID: 30 (Special Edition)
Book ID: 31
Book ID: 32
Book ID: 33
Book ID: 34
Book ID: 35 (Special Edition)
Book ID: 36
Book ID: 37
Book ID: 38
Book ID: 39
Book ID: 40 (Special Edition)
Book ID: 41
Book ID: 42
Book ID: 43
Book ID: 44
Book ID: 45 (Special Edition)
Book ID: 46
Book ID: 47
Book ID: 48
Book ID: 49
Book ID: 50 (Special Edition)
```

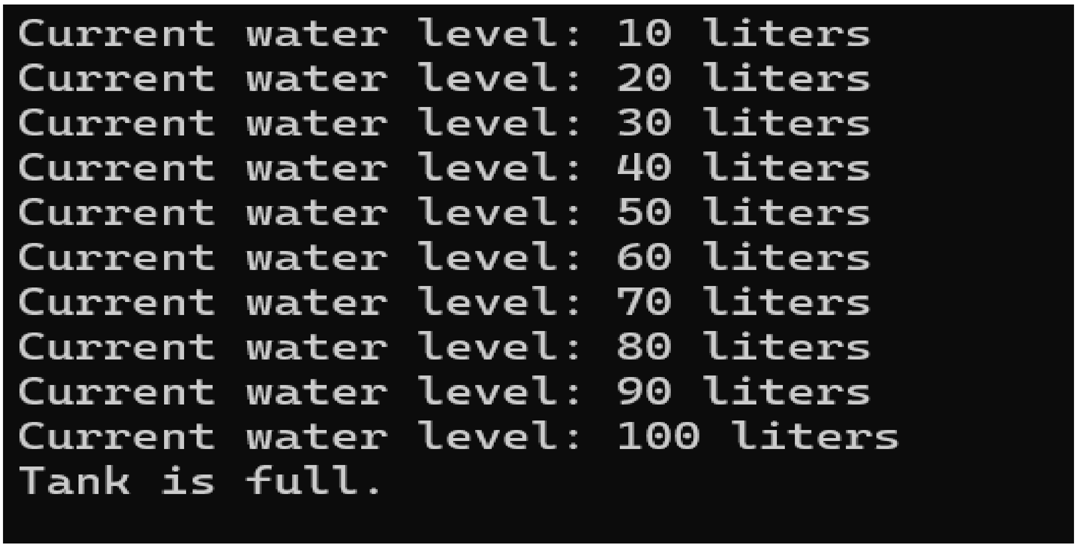


```c
#include <stdio.h>

int main() {
    printf("NAME : MAHI RABARI\nID NO: 25CE095\n");

    int i;
    for(i = 1; i <= 50; i++) {
        if(i % 5 == 0)
            printf("Book ID: %d (Special Edition)\n", i);
        else
            printf("Book ID: %d\n", i);
    }
    return 0;
}
```

PRACTICAL 13: **This system starts refilling an empty tank (initially at 0 liters) until it reaches its maximum capacity of 100 liters.**

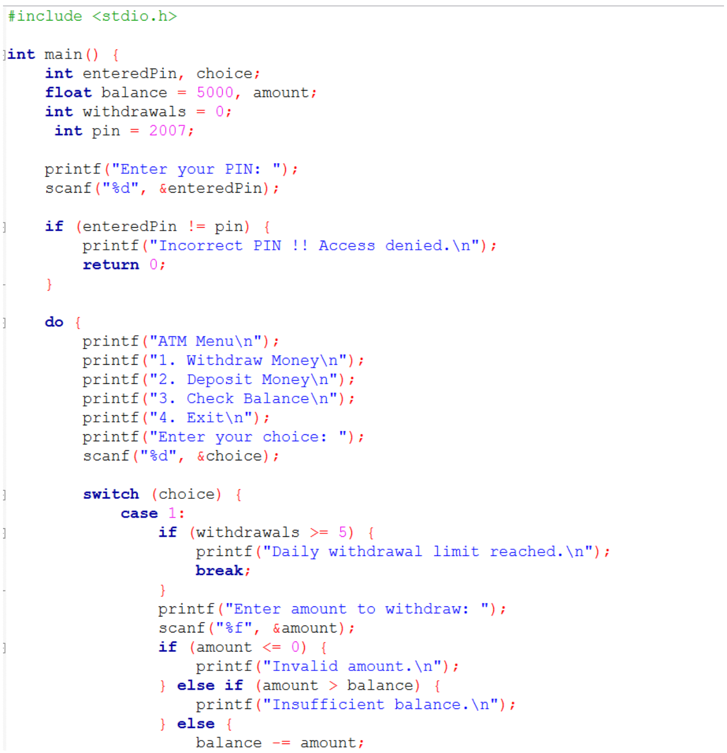


```c
#include <stdio.h>

int main() {
    int water;

    for(water = 10; water <= 100; water += 10) {
        printf("Current water level: %d liters\n", water);
    }

    printf("Tank is full.\n");
    return 0;
}
```

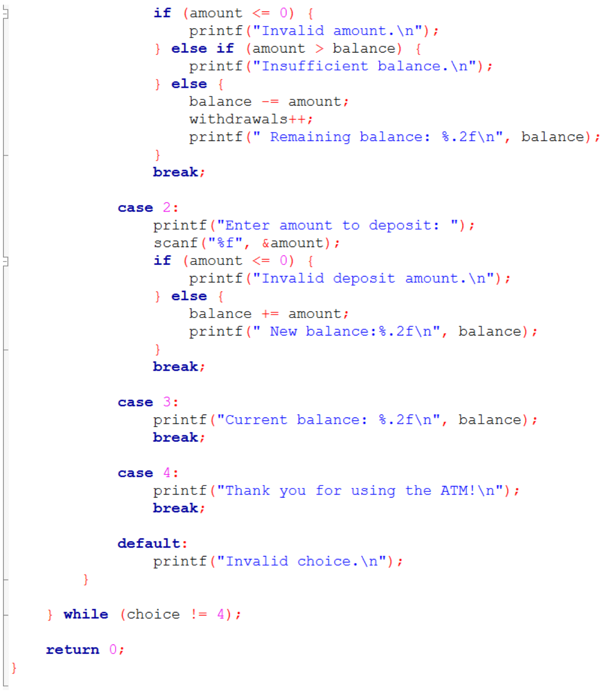


```
Current water level: 10 liters
Current water level: 20 liters
Current water level: 30 liters
Current water level: 40 liters
Current water level: 50 liters
Current water level: 60 liters
Current water level: 70 liters
Current water level: 80 liters
Current water level: 90 liters
Current water level: 100 liters
Tank is full.
```

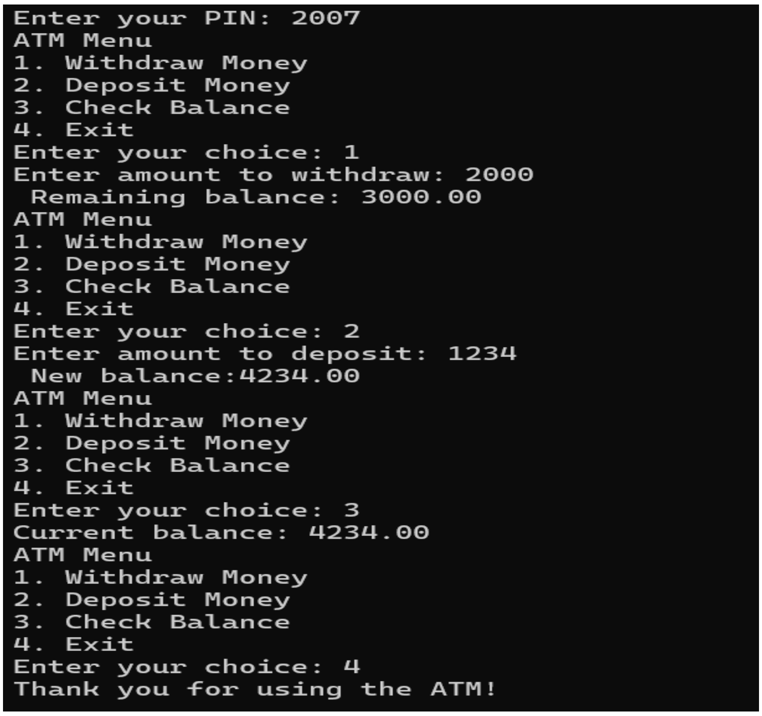
PRACTICAL 14: ATM SYSTEM



```c
#include <stdio.h>

int main() {
    int enteredPin, choice;
    float balance = 5000, amount;
    int withdrawals = 0;
     int pin = 2007;

    printf("Enter your PIN: ");
    scanf("%d", &enteredPin);

    if (enteredPin != pin) {
        printf("Incorrect PIN !! Access denied.\n");
        return 0;
    }

    do {
        printf("ATM Menu\n");
        printf("1. Withdraw Money\n");
        printf("2. Deposit Money\n");
        printf("3. Check Balance\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                if (withdrawals >= 5) {
                    printf("Daily withdrawal limit reached.\n");
                    break;
                }
                printf("Enter amount to withdraw: ");
                scanf("%f", &amount);
                if (amount <= 0) {
                    printf("Invalid amount.\n");
                } else if (amount > balance) {
                    printf("Insufficient balance.\n");
                } else {
                    balance -= amount;
```



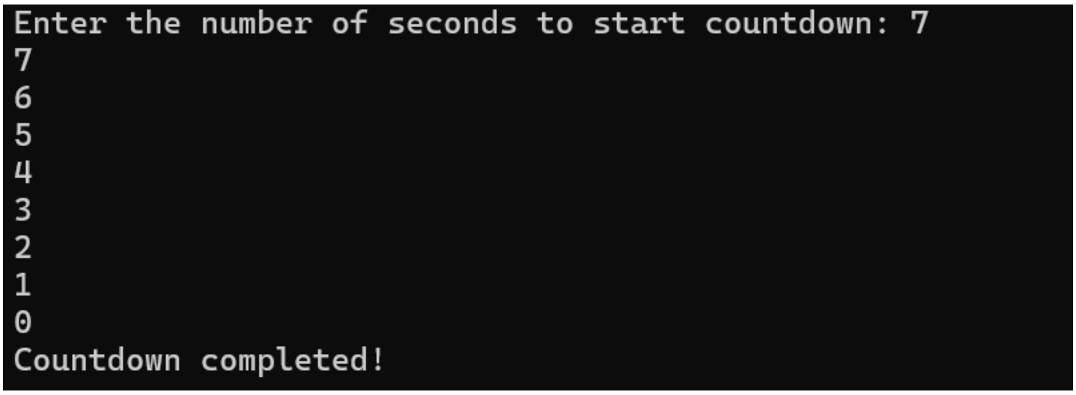
```c
                if (amount <= 0) {
                    printf("Invalid amount.\n");
                } else if (amount > balance) {
                    printf("Insufficient balance.\n");
                } else {
                    balance -= amount;
                    withdrawals++;
                    printf(" Remaining balance: %.2f\n", balance);
                }
                break;

            case 2:
                printf("Enter amount to deposit: ");
                scanf("%f", &amount);
                if (amount <= 0) {
                    printf("Invalid deposit amount.\n");
                } else {
                    balance += amount;
                    printf(" New balance:%.2f\n", balance);
                }
                break;

            case 3:
                printf("Current balance: %.2f\n", balance);
                break;

            case 4:
                printf("Thank you for using the ATM!\n");
                break;

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

    } while (choice != 4);

    return 0;
}
```



```
Enter your PIN: 2007
ATM Menu
1. Withdraw Money
2. Deposit Money
3. Check Balance
4. Exit
Enter your choice: 1
Enter amount to withdraw: 2000
 Remaining balance: 3000.00
ATM Menu
1. Withdraw Money
2. Deposit Money
3. Check Balance
4. Exit
Enter your choice: 2
Enter amount to deposit: 1234
 New balance:4234.00
ATM Menu
1. Withdraw Money
2. Deposit Money
3. Check Balance
4. Exit
Enter your choice: 3
Current balance: 4234.00
ATM Menu
1. Withdraw Money
2. Deposit Money
3. Check Balance
4. Exit
Enter your choice: 4
Thank you for using the ATM!
```

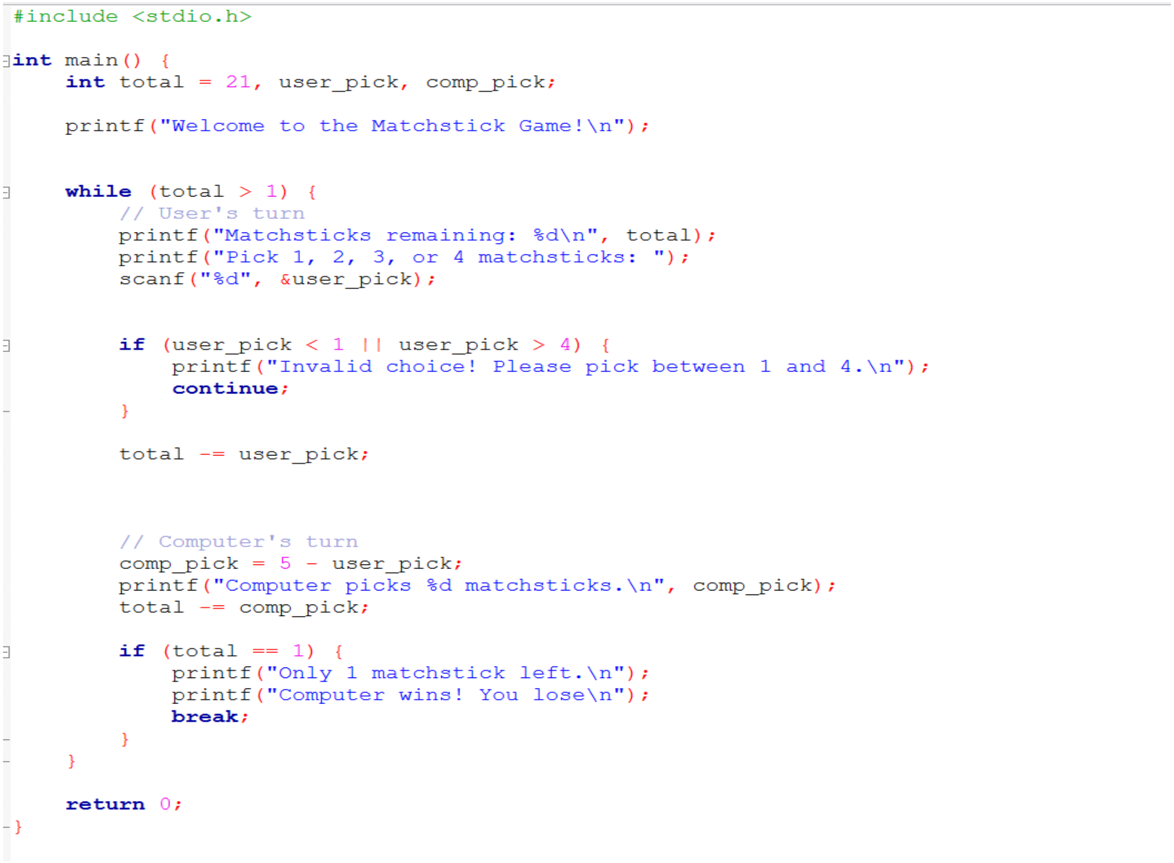
PRACTICAL 15 : Allows user to set a starting number of seconds and then count down to zero, displaying each second as it decrements.

```c
#include <stdio.h>

int main() {
    int seconds;

    printf("Enter the number of seconds to start countdown: ");
    scanf("%d", &seconds);

    if (seconds < 0) {
        printf("Please enter a non-negative number.\n");
        return 0;
    }

    while (seconds >= 0) {
        printf("%d\n", seconds);
        seconds--;
    }

    printf("Countdown completed!\n");
    return 0;
}
```

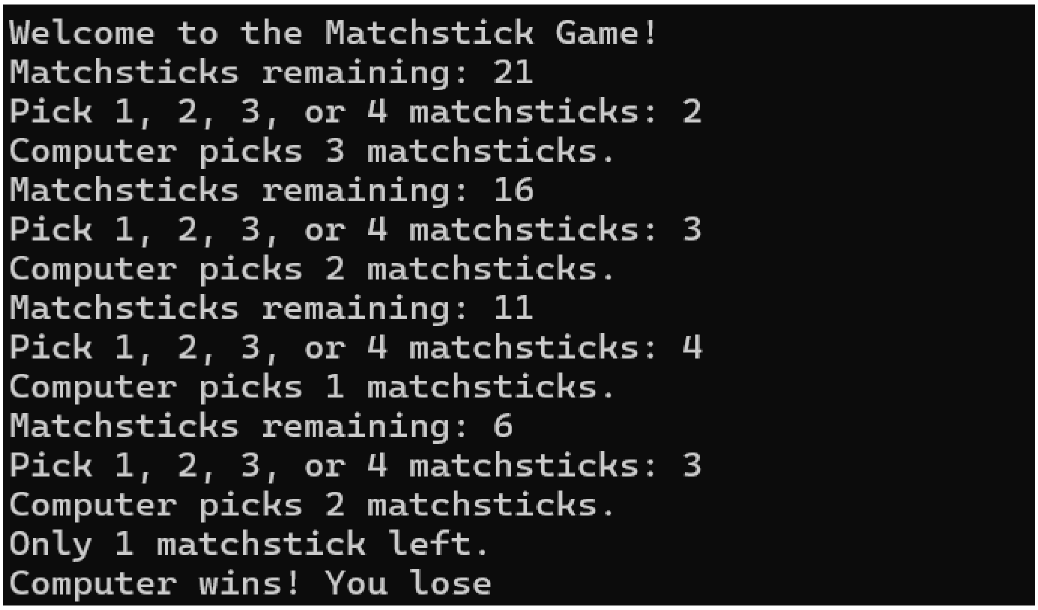


```
Enter the number of seconds to start countdown: 7
7
6
5
4
3
2
1
0
Countdown completed!
```

PRACTICAL 16 : A matchstick game between the user and the computer.

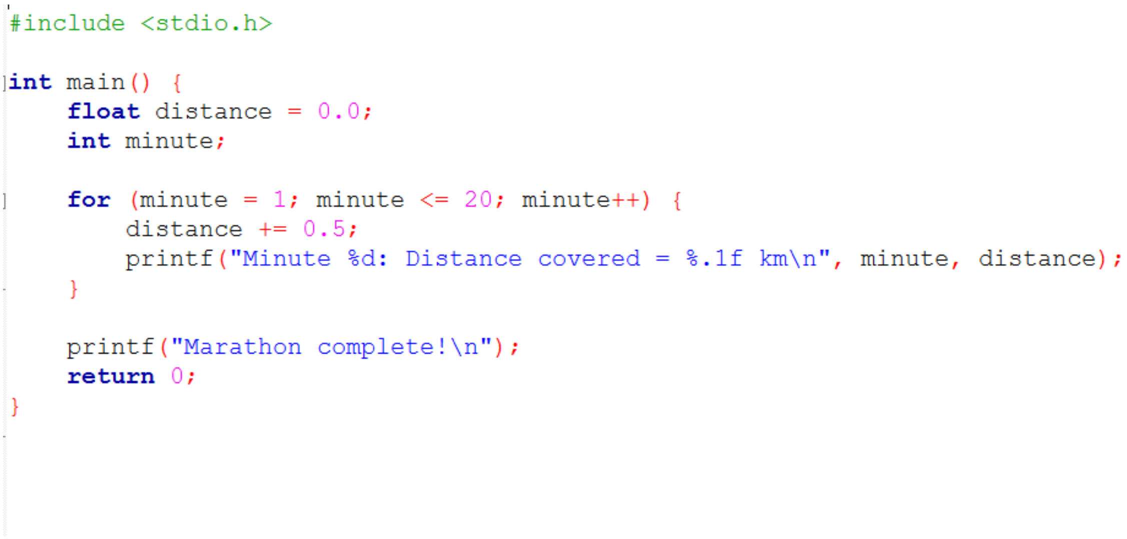


```c
#include <stdio.h>

int main() {
    int total = 21, user_pick, comp_pick;

    printf("Welcome to the Matchstick Game!\n");


    while (total > 1) {
        // User's turn
        printf("Matchsticks remaining: %d\n", total);
        printf("Pick 1, 2, 3, or 4 matchsticks: ");
        scanf("%d", &user_pick);


        if (user_pick < 1 || user_pick > 4) {
            printf("Invalid choice! Please pick between 1 and 4.\n");
            continue;
        }

        total -= user_pick;



        // Computer's turn
        comp_pick = 5 - user_pick;
        printf("Computer picks %d matchsticks.\n", comp_pick);
        total -= comp_pick;

        if (total == 1) {
            printf("Only 1 matchstick left.\n");
            printf("Computer wins! You lose\n");
            break;
        }
    }

    return 0;
}
```

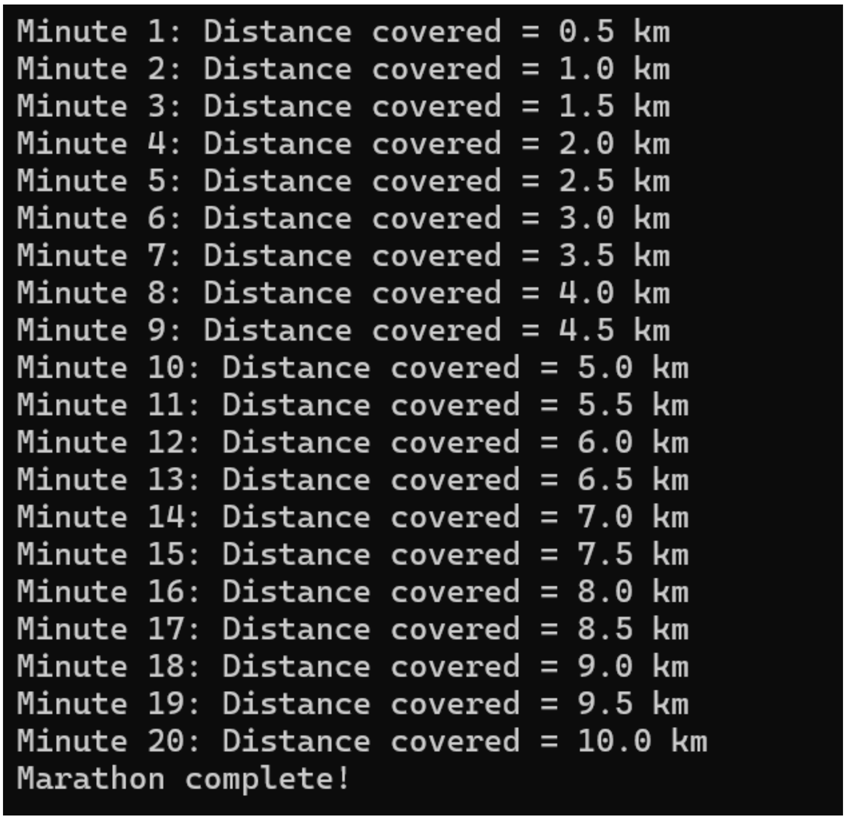


```
Welcome to the Matchstick Game!
Matchsticks remaining: 21
Pick 1, 2, 3, or 4 matchsticks: 2
Computer picks 3 matchsticks.
Matchsticks remaining: 16
Pick 1, 2, 3, or 4 matchsticks: 3
Computer picks 2 matchsticks.
Matchsticks remaining: 11
Pick 1, 2, 3, or 4 matchsticks: 4
Computer picks 1 matchsticks.
Matchsticks remaining: 6
Pick 1, 2, 3, or 4 matchsticks: 3
Computer picks 2 matchsticks.
Only 1 matchstick left.
Computer wins! You lose
```

PRACTICAL 17: A program to simulate the distance tracking.

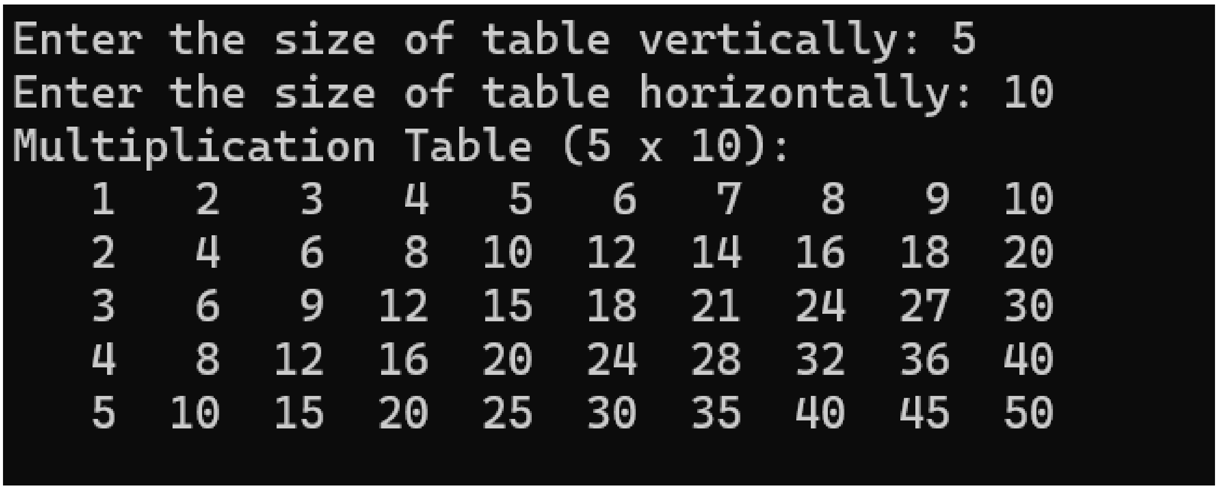


```c
#include <stdio.h>

int main() {
    float distance = 0.0;
    int minute;

    for (minute = 1; minute <= 20; minute++) {
        distance += 0.5;
        printf("Minute %d: Distance covered = %.1f km\n", minute, distance);
    }

    printf("Marathon complete!\n");
    return 0;
}
```

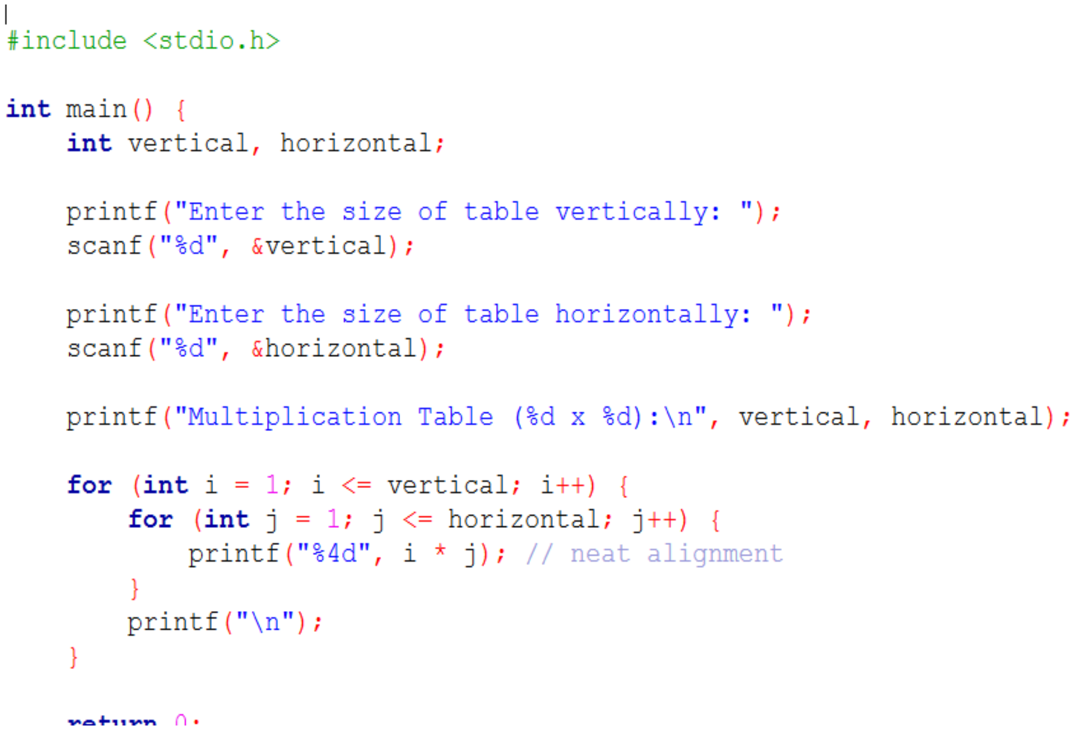


```
Minute 1: Distance covered = 0.5 km
Minute 2: Distance covered = 1.0 km
Minute 3: Distance covered = 1.5 km
Minute 4: Distance covered = 2.0 km
Minute 5: Distance covered = 2.5 km
Minute 6: Distance covered = 3.0 km
Minute 7: Distance covered = 3.5 km
Minute 8: Distance covered = 4.0 km
Minute 9: Distance covered = 4.5 km
Minute 10: Distance covered = 5.0 km
Minute 11: Distance covered = 5.5 km
Minute 12: Distance covered = 6.0 km
Minute 13: Distance covered = 6.5 km
Minute 14: Distance covered = 7.0 km
Minute 15: Distance covered = 7.5 km
Minute 16: Distance covered = 8.0 km
Minute 17: Distance covered = 8.5 km
Minute 18: Distance covered = 9.0 km
Minute 19: Distance covered = 9.5 km
Minute 20: Distance covered = 10.0 km
Marathon complete!
```

PRACTICAL 18: **Display a multiplication table based on user input**

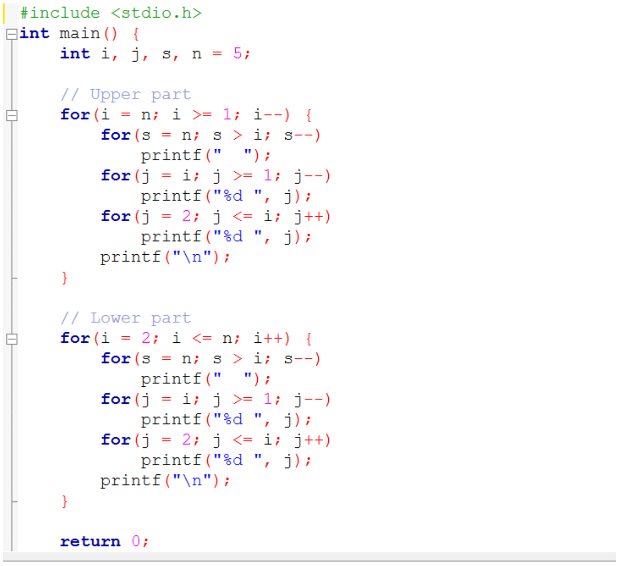


```
Enter the size of table vertically: 5
Enter the size of table horizontally: 10
Multiplication Table (5 x 10):
   1   2   3   4   5   6   7   8   9  10
   2   4   6   8  10  12  14  16  18  20
   3   6   9  12  15  18  21  24  27  30
   4   8  12  16  20  24  28  32  36  40
   5  10  15  20  25  30  35  40  45  50
```

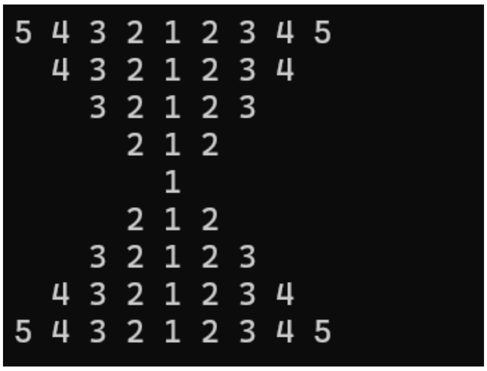


```c
#include <stdio.h>

int main() {
    int vertical, horizontal;

    printf("Enter the size of table vertically: ");
    scanf("%d", &vertical);

    printf("Enter the size of table horizontally: ");
    scanf("%d", &horizontal);

    printf("Multiplication Table (%d x %d):\n", vertical, horizontal);

    for (int i = 1; i <= vertical; i++) {
        for (int j = 1; j <= horizontal; j++) {
            printf("%4d", i * j); // neat alignment
        }
        printf("\n");
    }

    return 0;
```

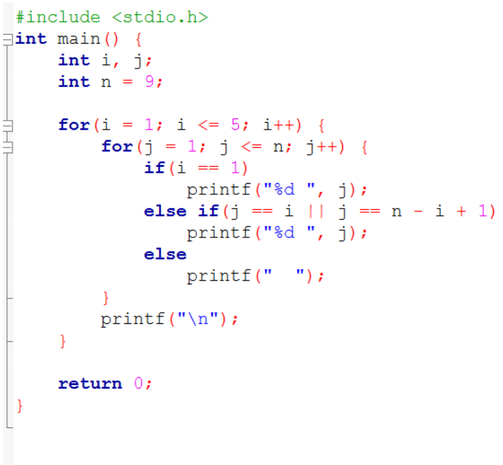
PRACTICAL 19:PATTERNS



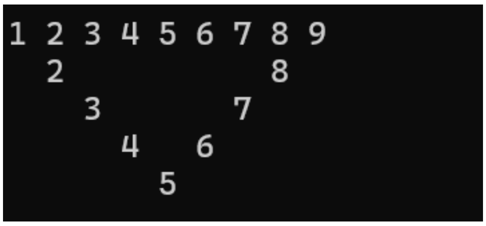
```
#include <stdio.h>
int main() {
    int i, j, s, n = 5;

    // Upper part
    for(i = n; i >= 1; i--) {
        for(s = n; s > i; s--)
            printf("  ");
        for(j = i; j >= 1; j--)
            printf("%d ", j);
        for(j = 2; j <= i; j++)
            printf("%d ", j);
        printf("\n");
    }

    // Lower part
    for(i = 2; i <= n; i++) {
        for(s = n; s > i; s--)
            printf("  ");
        for(j = i; j >= 1; j--)
            printf("%d ", j);
        for(j = 2; j <= i; j++)
            printf("%d ", j);
        printf("\n");
    }

    return 0;
```



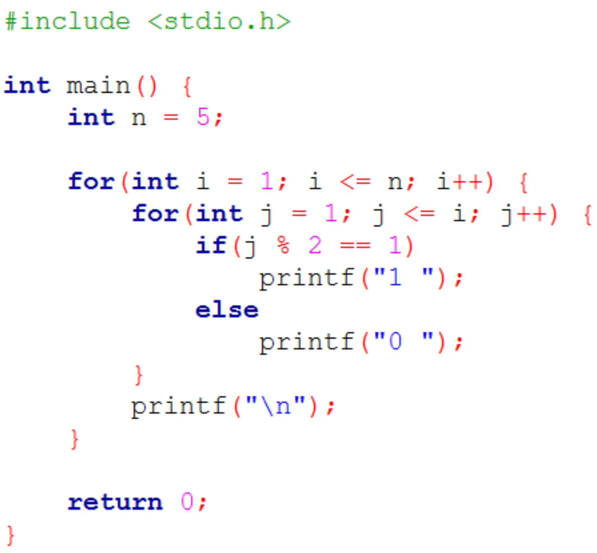
```
5 4 3 2 1 2 3 4 5
  4 3 2 1 2 3 4
    3 2 1 2 3
      2 1 2
        1
      2 1 2
    3 2 1 2 3
  4 3 2 1 2 3 4
5 4 3 2 1 2 3 4 5
```



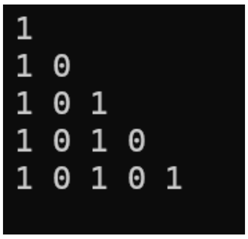
```
#include <stdio.h>
int main() {
    int i, j;
    int n = 9;

    for(i = 1; i <= 5; i++) {
        for(j = 1; j <= n; j++) {
            if(i == 1)
                printf("%d ", j);
            else if(j == i || j == n - i + 1)
                printf("%d ", j);
            else
                printf("  ");
        }
        printf("\n");
    }

    return 0;
}
```



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

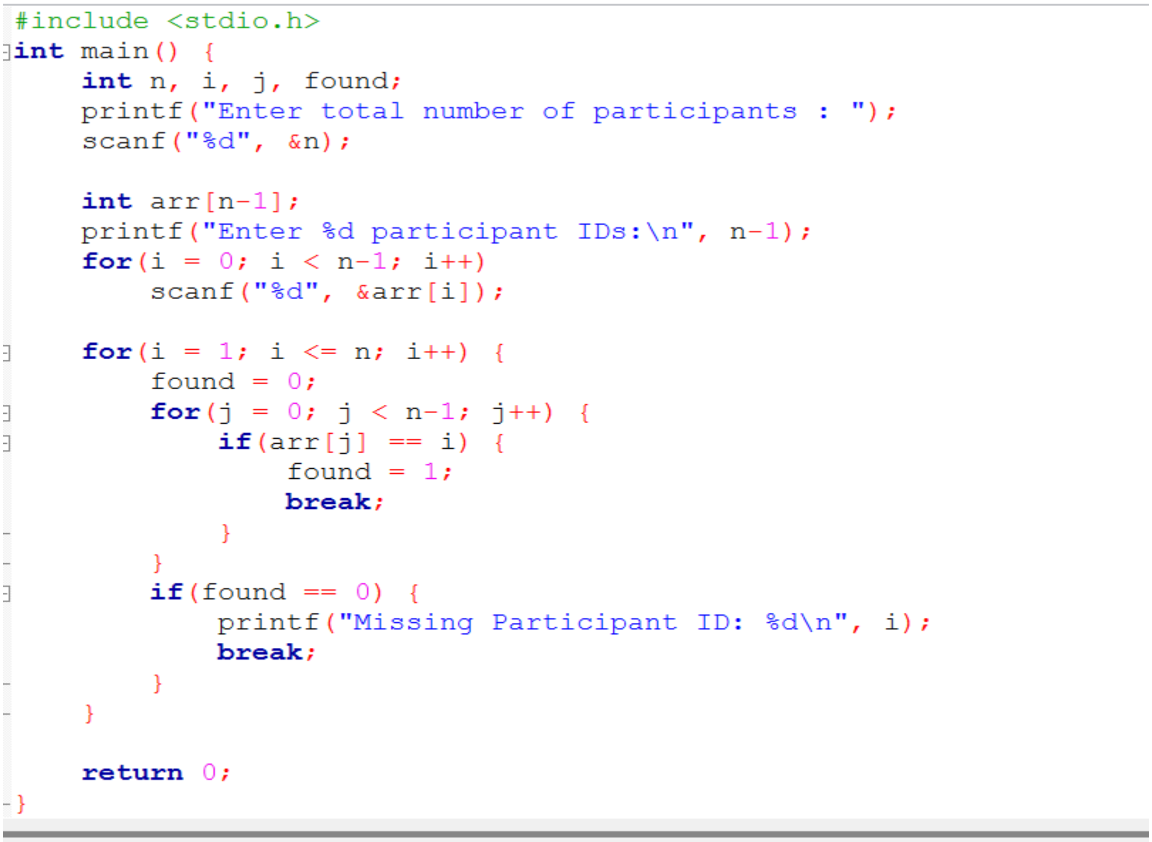


```
#include <stdio.h>

int main() {
    int n = 5;

    for(int i = 1; i <= n; i++) {
        for(int j = 1; j <= i; j++) {
            if(j % 2 == 1)
                printf("1 ");
            else
                printf("0 ");
        }
        printf("\n");
    }

    return 0;
}
```

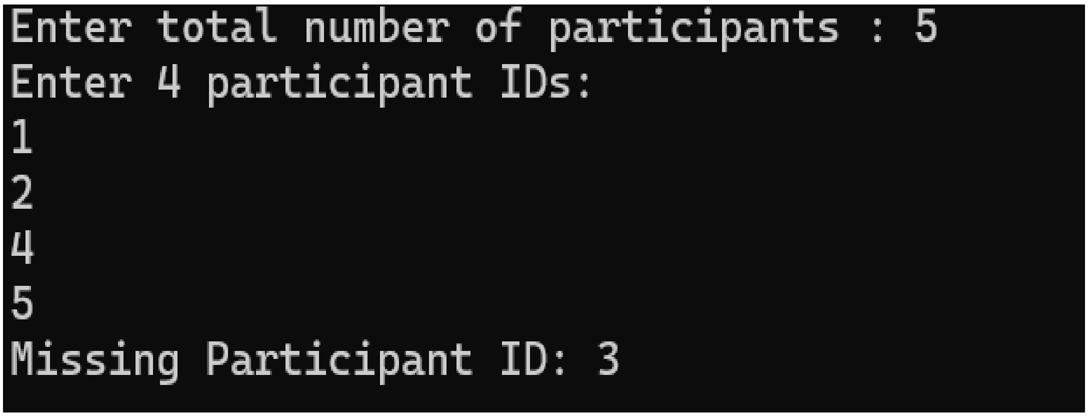


```
1
1 0
1 0 1
1 0 1 0
1 0 1 0 1
```

PRACTICAL 20:FIND MISSINF ID

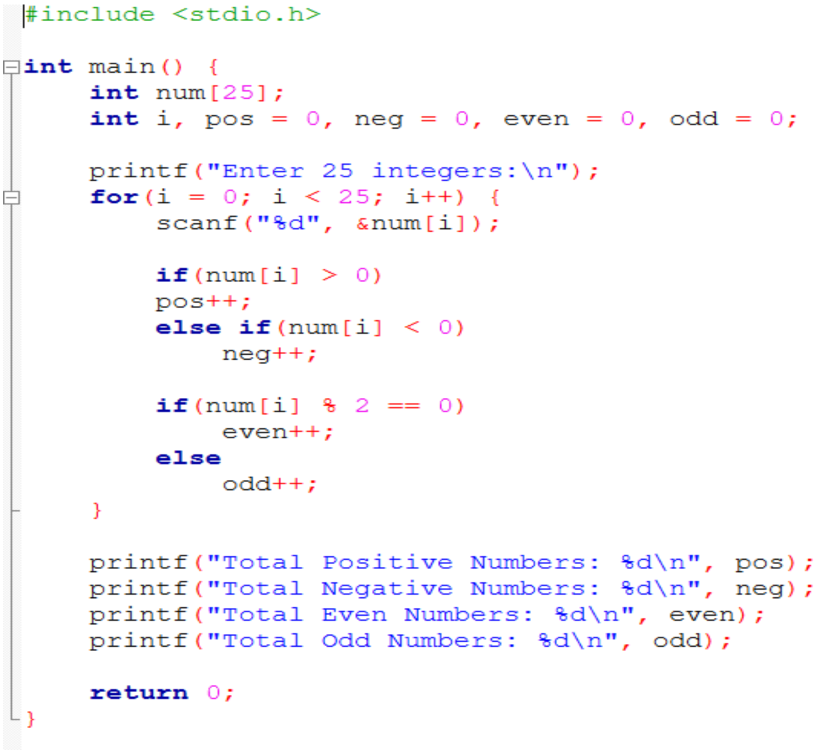


```c
#include <stdio.h>
int main() {
    int n, i, j, found;
    printf("Enter total number of participants : ");
    scanf("%d", &n);

    int arr[n-1];
    printf("Enter %d participant IDs:\n", n-1);
    for(i = 0; i < n-1; i++)
        scanf("%d", &arr[i]);

    for(i = 1; i <= n; i++) {
        found = 0;
        for(j = 0; j < n-1; j++) {
            if(arr[j] == i) {
                found = 1;
                break;
            }
        }
        if(found == 0) {
            printf("Missing Participant ID: %d\n", i);
            break;
        }
    }

    return 0;
}
```

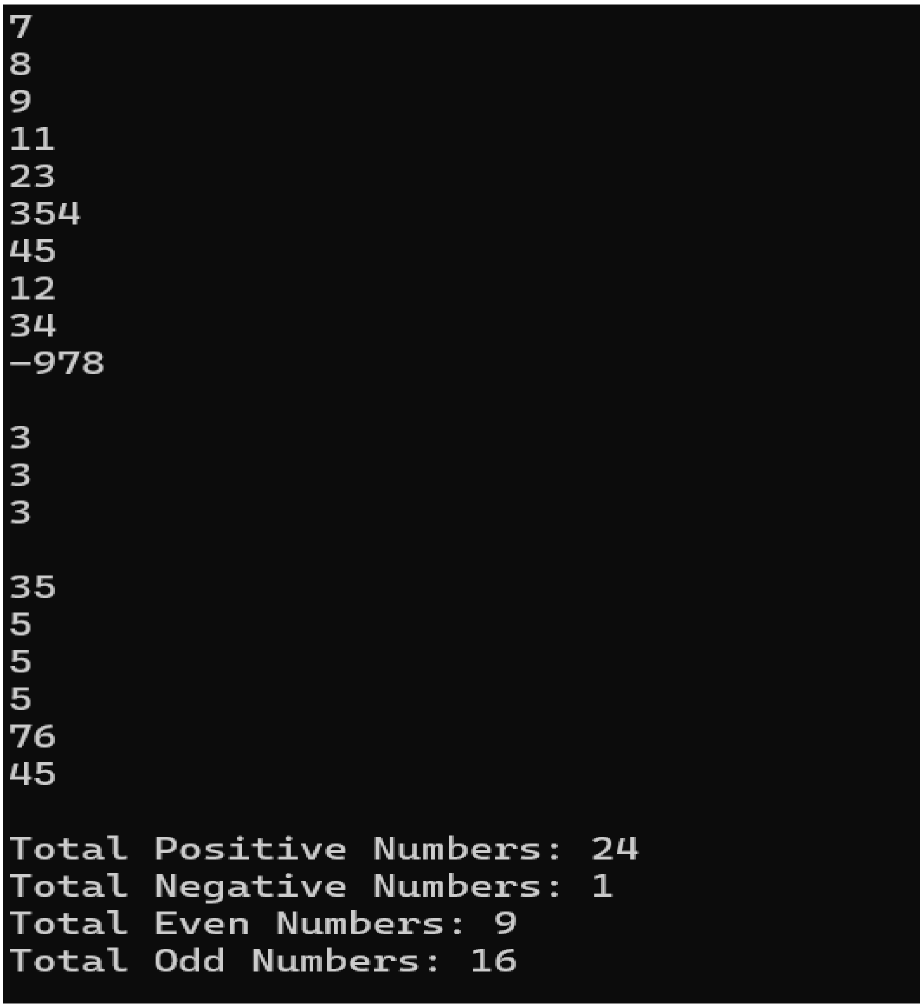


```
Enter total number of participants : 5
Enter 4 participant IDs:
1
2
4
5
Missing Participant ID: 3
```

PRACTICAL 21: C based console application that assists operators in understanding the nature of the numbers they enter.

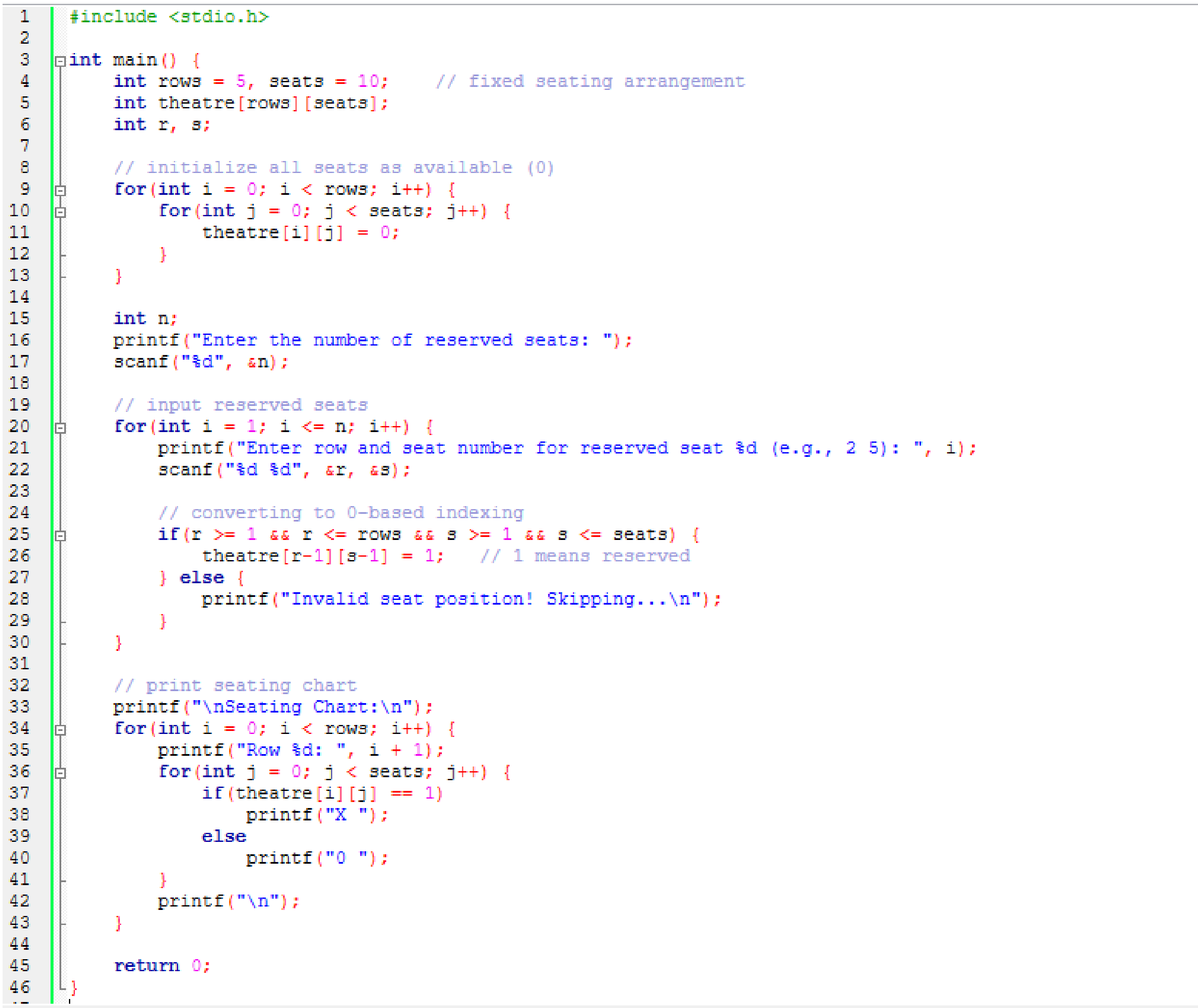


```c
#include <stdio.h>

int main() {
    int num[25];
    int i, pos = 0, neg = 0, even = 0, odd = 0;

    printf("Enter 25 integers:\n");
    for(i = 0; i < 25; i++) {
        scanf("%d", &num[i]);

        if(num[i] > 0)
        pos++;
        else if(num[i] < 0)
            neg++;

        if(num[i] % 2 == 0)
            even++;
        else
            odd++;
    }

    printf("Total Positive Numbers: %d\n", pos);
    printf("Total Negative Numbers: %d\n", neg);
    printf("Total Even Numbers: %d\n", even);
    printf("Total Odd Numbers: %d\n", odd);

    return 0;
}
```

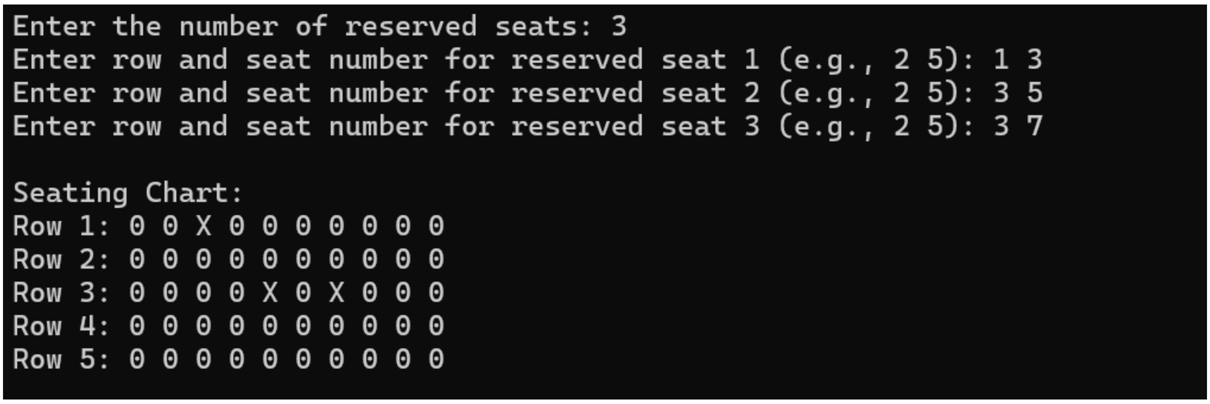


```
7
8
9
11
23
354
45
12
34
-978

3
3
3

35
5
5
5
76
45

Total Positive Numbers: 24
Total Negative Numbers: 1
Total Even Numbers: 9
Total Odd Numbers: 16
```

PRACTICAL 22:THEATRE SEAT BOOKING

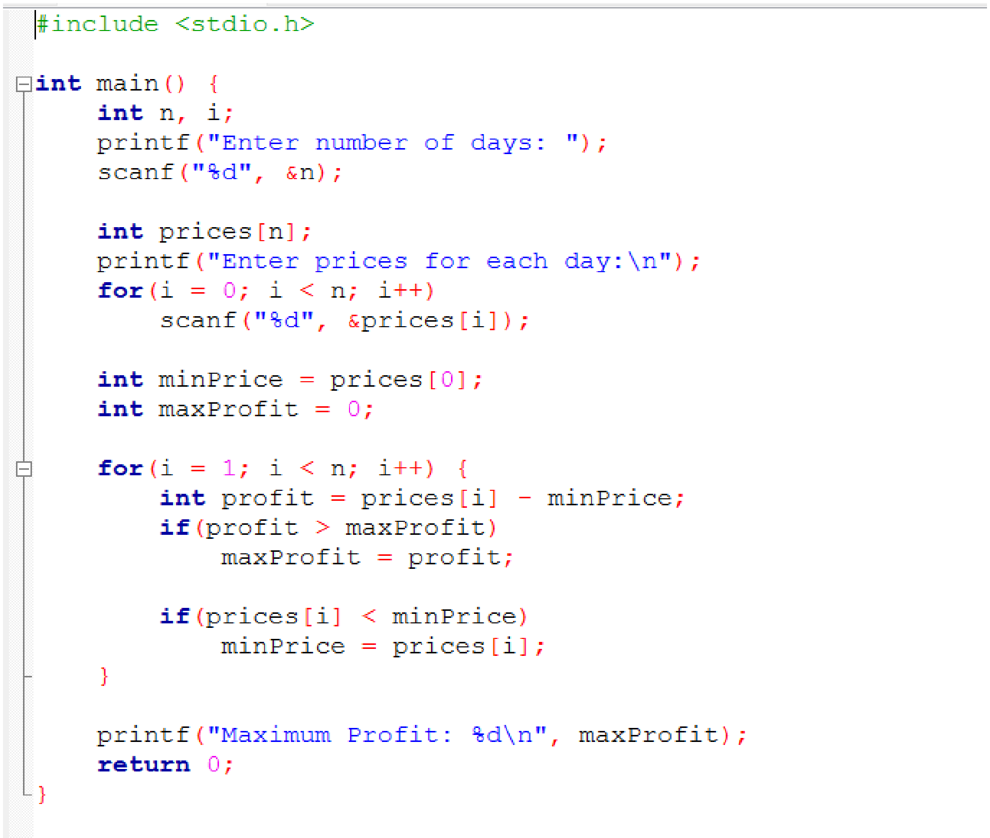


```c
#include <stdio.h>

int main() {
    int rows = 5, seats = 10;     // fixed seating arrangement
    int theatre[rows][seats];
    int r, s;

    // initialize all seats as available (0)
    for(int i = 0; i < rows; i++) {
        for(int j = 0; j < seats; j++) {
            theatre[i][j] = 0;
        }
    }

    int n;
    printf("Enter the number of reserved seats: ");
    scanf("%d", &n);

    // input reserved seats
    for(int i = 1; i <= n; i++) {
        printf("Enter row and seat number for reserved seat %d (e.g., 2 5): ", i);
        scanf("%d %d", &r, &s);

        // converting to 0-based indexing
        if(r >= 1 && r <= rows && s >= 1 && s <= seats) {
            theatre[r-1][s-1] = 1;    // 1 means reserved
        } else {
            printf("Invalid seat position! Skipping...\n");
        }
    }

    // print seating chart
    printf("\nSeating Chart:\n");
    for(int i = 0; i < rows; i++) {
        printf("Row %d: ", i + 1);
        for(int j = 0; j < seats; j++) {
            if(theatre[i][j] == 1)
                printf("X ");
            else
                printf("0 ");
        }
        printf("\n");
    }

    return 0;
}
```

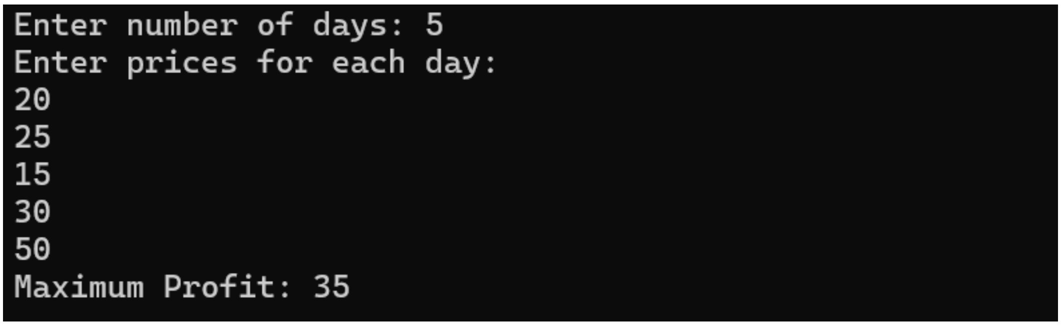


```
Enter the number of reserved seats: 3
Enter row and seat number for reserved seat 1 (e.g., 2 5): 1 3
Enter row and seat number for reserved seat 2 (e.g., 2 5): 3 5
Enter row and seat number for reserved seat 3 (e.g., 2 5): 3 7

Seating Chart:
Row 1: 0 0 X 0 0 0 0 0 0 0
Row 2: 0 0 0 0 0 0 0 0 0 0
Row 3: 0 0 0 0 X 0 X 0 0 0
Row 4: 0 0 0 0 0 0 0 0 0 0
Row 5: 0 0 0 0 0 0 0 0 0 0
```

PRACTICAL 23: **C to determine the maximum profit you can achieve from this transaction. If no profit is possible (i.e., the fruit price only decreases or stays the same), return 0.**

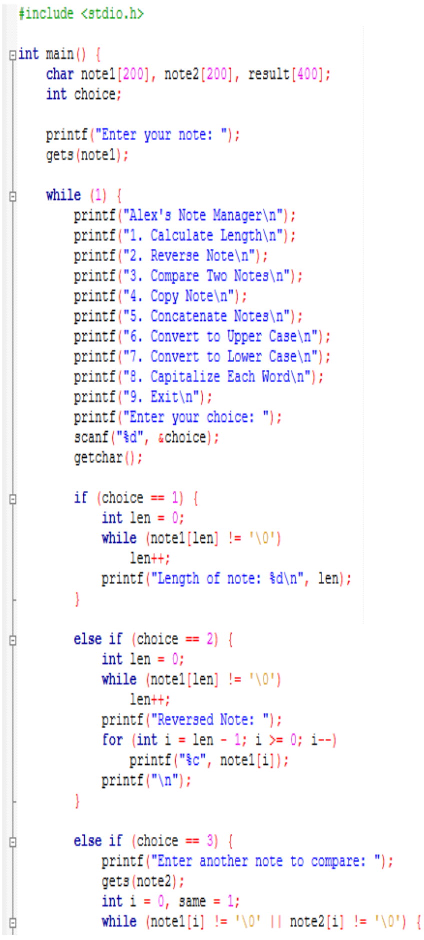


```c
#include <stdio.h>

int main() {
    int n, i;
    printf("Enter number of days: ");
    scanf("%d", &n);

    int prices[n];
    printf("Enter prices for each day:\n");
    for(i = 0; i < n; i++)
        scanf("%d", &prices[i]);

    int minPrice = prices[0];
    int maxProfit = 0;

    for(i = 1; i < n; i++) {
        int profit = prices[i] - minPrice;
        if(profit > maxProfit)
            maxProfit = profit;

        if(prices[i] < minPrice)
            minPrice = prices[i];
    }

    printf("Maximum Profit: %d\n", maxProfit);
    return 0;
}
```

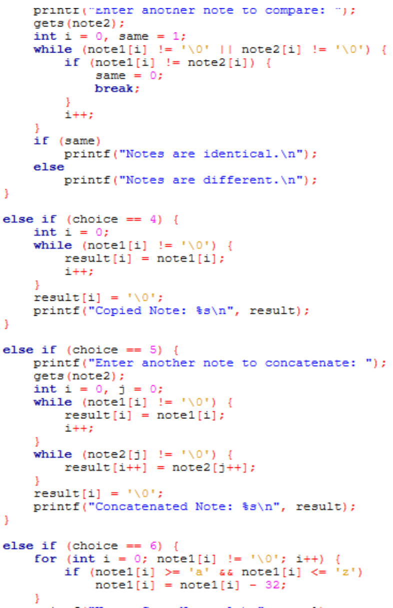


```
Enter number of days: 5
Enter prices for each day:
20
25
15
30
50
Maximum Profit: 35
```

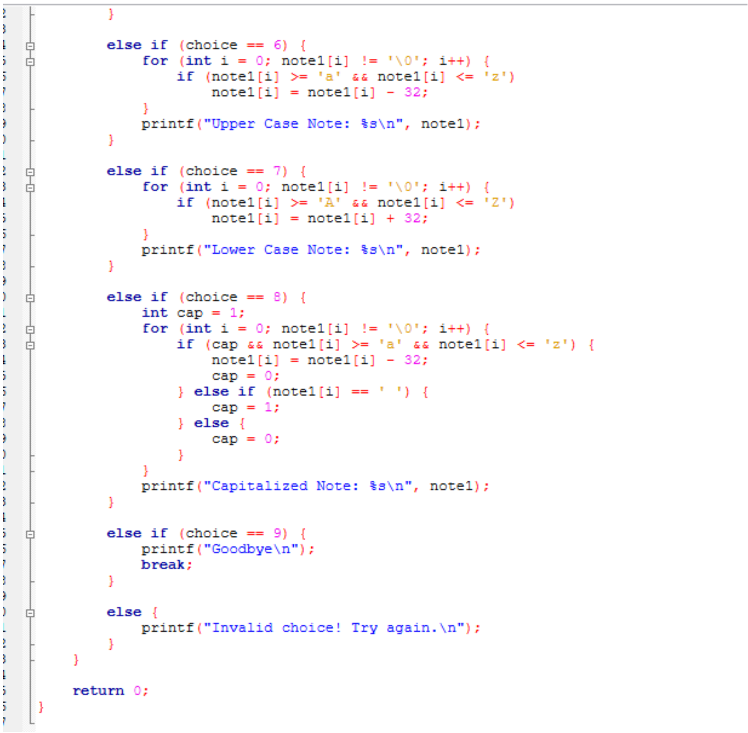
PRACTICAL 24: Manage notes and perform the following operations manually (without using built-in string manipulation functions from):



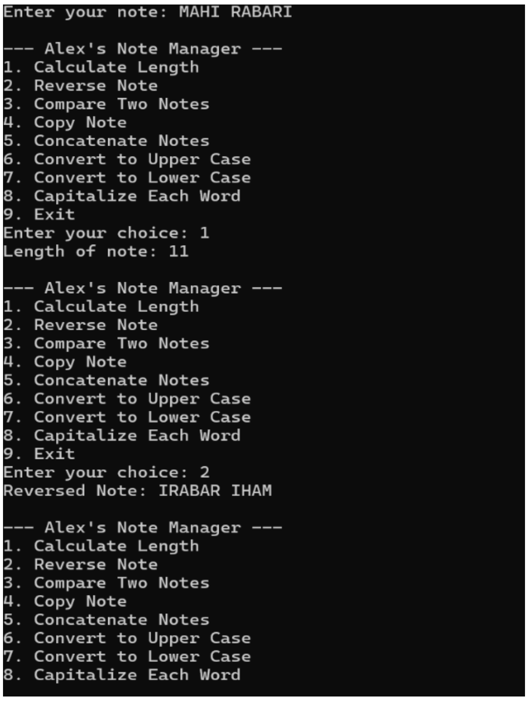
```c
#include <stdio.h>

int main() {
    char note1[200], note2[200], result[400];
    int choice;

    printf("Enter your note: ");
    gets(note1);

    while (1) {
        printf("Alex's Note Manager\n");
        printf("1. Calculate Length\n");
        printf("2. Reverse Note\n");
        printf("3. Compare Two Notes\n");
        printf("4. Copy Note\n");
        printf("5. Concatenate Notes\n");
        printf("6. Convert to Upper Case\n");
        printf("7. Convert to Lower Case\n");
        printf("8. Capitalize Each Word\n");
        printf("9. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        getchar();

        if (choice == 1) {
            int len = 0;
            while (note1[len] != '\0')
                len++;
            printf("Length of note: %d\n", len);
        }

        else if (choice == 2) {
            int len = 0;
            while (note1[len] != '\0')
                len++;
            printf("Reversed Note: ");
            for (int i = len - 1; i >= 0; i--)
                printf("%c", note1[i]);
            printf("\n");
        }

        else if (choice == 3) {
            printf("Enter another note to compare: ");
            gets(note2);
            int i = 0, same = 1;
            while (note1[i] != '\0' || note2[i] != '\0') {
```



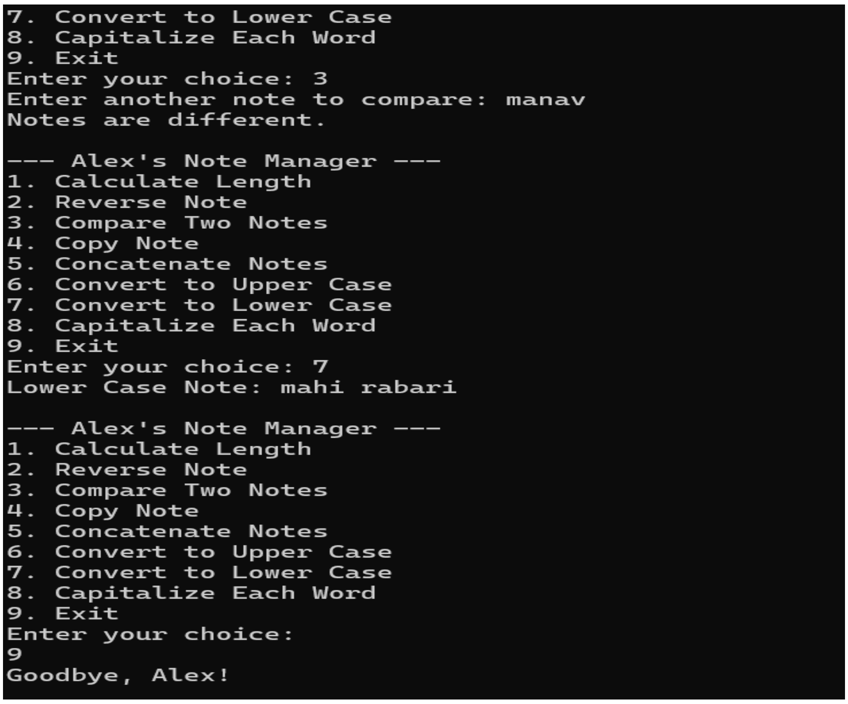
```c
            printf("Enter another note to compare: ");
            gets(note2);
            int i = 0, same = 1;
            while (note1[i] != '\0' || note2[i] != '\0') {
                if (note1[i] != note2[i]) {
                    same = 0;
                    break;
                }
                i++;
            }
            if (same)
                printf("Notes are identical.\n");
            else
                printf("Notes are different.\n");
        }

        else if (choice == 4) {
            int i = 0;
            while (note1[i] != '\0') {
                result[i] = note1[i];
                i++;
            }
            result[i] = '\0';
            printf("Copied Note: %s\n", result);
        }

        else if (choice == 5) {
            printf("Enter another note to concatenate: ");
            gets(note2);
            int i = 0, j = 0;
            while (note1[i] != '\0') {
                result[i] = note1[i];
                i++;
            }
            while (note2[j] != '\0') {
                result[i++] = note2[j++];
            }
            result[i] = '\0';
            printf("Concatenated Note: %s\n", result);
        }

        else if (choice == 6) {
            for (int i = 0; note1[i] != '\0'; i++) {
                if (note1[i] >= 'a' && note1[i] <= 'z')
                    note1[i] = note1[i] - 32;
            }
```



```c
        }

        else if (choice == 6) {
            for (int i = 0; note1[i] != '\0'; i++) {
                if (note1[i] >= 'a' && note1[i] <= 'z')
                    note1[i] = note1[i] - 32;
            }
            printf("Upper Case Note: %s\n", note1);
        }

        else if (choice == 7) {
            for (int i = 0; note1[i] != '\0'; i++) {
                if (note1[i] >= 'A' && note1[i] <= 'Z')
                    note1[i] = note1[i] + 32;
            }
            printf("Lower Case Note: %s\n", note1);
        }

        else if (choice == 8) {
            int cap = 1;
            for (int i = 0; note1[i] != '\0'; i++) {
                if (cap && note1[i] >= 'a' && note1[i] <= 'z') {
                    note1[i] = note1[i] - 32;
                    cap = 0;
                } else if (note1[i] == ' ') {
                    cap = 1;
                } else {
                    cap = 0;
                }
            }
            printf("Capitalized Note: %s\n", note1);
        }

        else if (choice == 9) {
            printf("Goodbye\n");
            break;
        }

        else {
            printf("Invalid choice! Try again.\n");
        }
    }

    return 0;
}
```

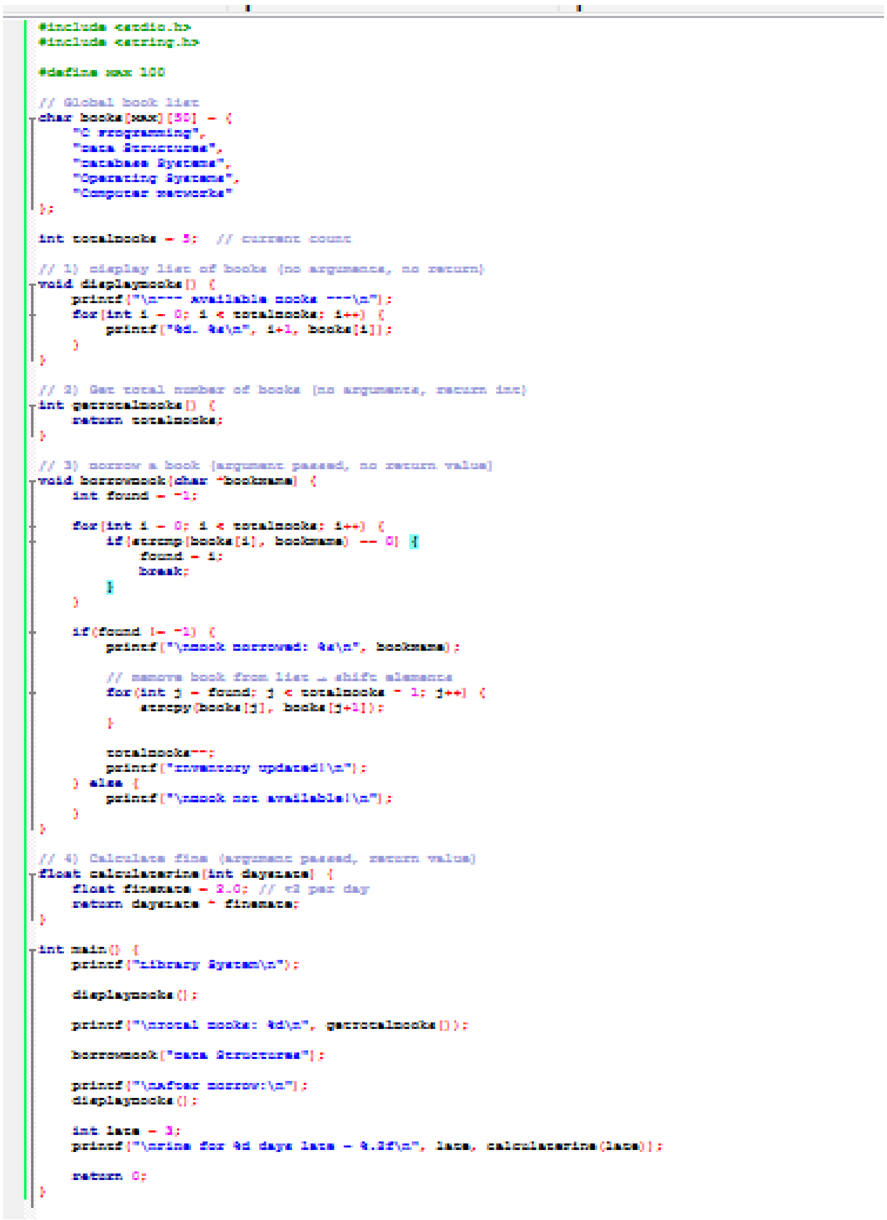


```
Enter your note: MAHI RABARI

--- Alex's Note Manager ---
1. Calculate Length
2. Reverse Note
3. Compare Two Notes
4. Copy Note
5. Concatenate Notes
6. Convert to Upper Case
7. Convert to Lower Case
8. Capitalize Each Word
9. Exit
Enter your choice: 1
Length of note: 11

--- Alex's Note Manager ---
1. Calculate Length
2. Reverse Note
3. Compare Two Notes
4. Copy Note
5. Concatenate Notes
6. Convert to Upper Case
7. Convert to Lower Case
8. Capitalize Each Word
9. Exit
Enter your choice: 2
Reversed Note: IRABAR IHAM

--- Alex's Note Manager ---
1. Calculate Length
2. Reverse Note
3. Compare Two Notes
4. Copy Note
5. Concatenate Notes
6. Convert to Upper Case
7. Convert to Lower Case
8. Capitalize Each Word
```

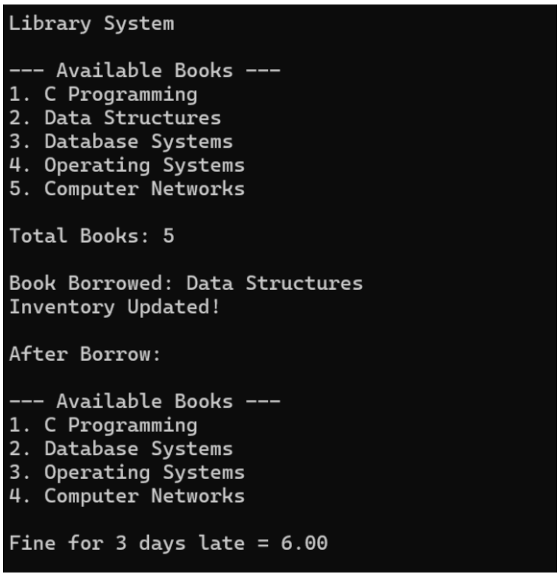


```
7. Convert to Lower Case
8. Capitalize Each Word
9. Exit
Enter your choice: 3
Enter another note to compare: manav
Notes are different.

--- Alex's Note Manager ---
1. Calculate Length
2. Reverse Note
3. Compare Two Notes
4. Copy Note
5. Concatenate Notes
6. Convert to Upper Case
7. Convert to Lower Case
8. Capitalize Each Word
9. Exit
Enter your choice: 7
Lower Case Note: mahi rabari

--- Alex's Note Manager ---
1. Calculate Length
2. Reverse Note
3. Compare Two Notes
4. Copy Note
5. Concatenate Notes
6. Convert to Upper Case
7. Convert to Lower Case
8. Capitalize Each Word
9. Exit
Enter your choice:
9
Goodbye, Alex!
```

PRACTICAL 25: library management

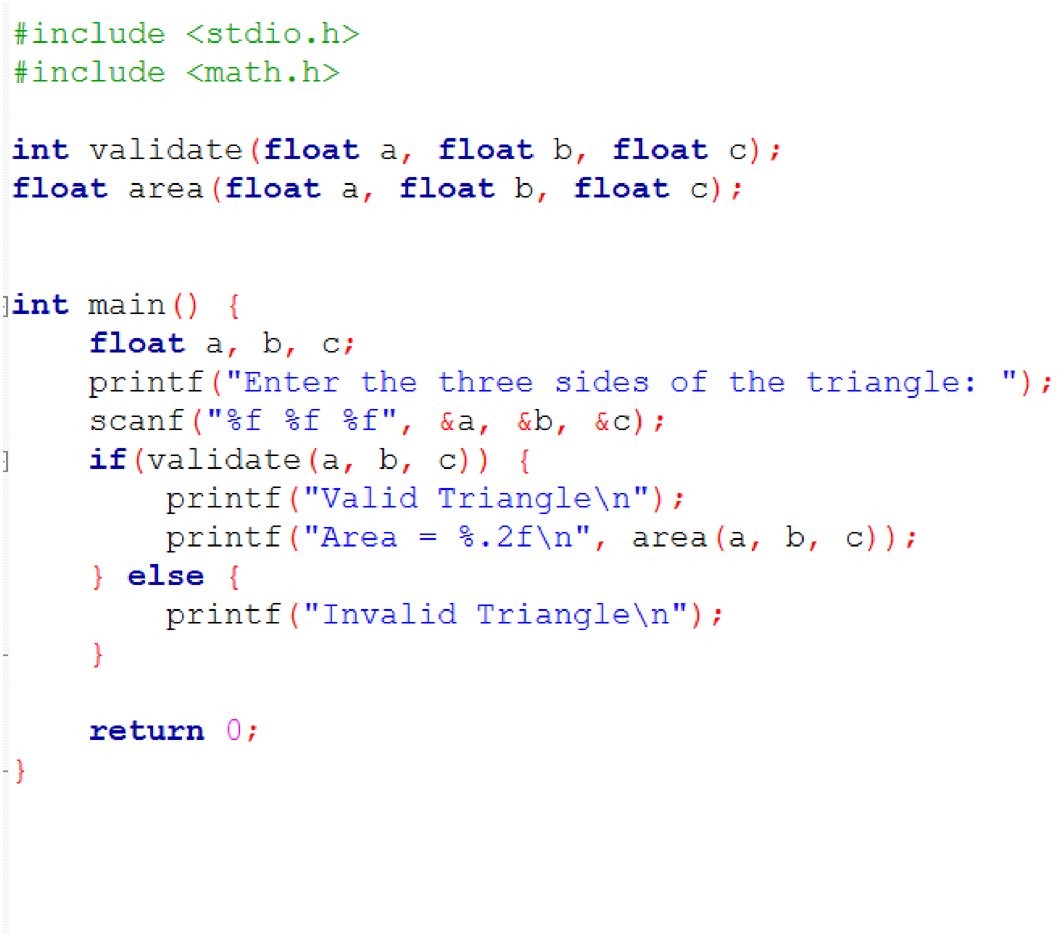


```c
#include <stdio.h>
#include <string.h>

#define MAX 100

// Global book list
char books[MAX][50] = {
    "C Programming",
    "Data Structures",
    "Database Systems",
    "Operating Systems",
    "Computer Networks"
};

int totalBooks = 5;  // current count

// 1) Display list of books (no arguments, no return)
void displayBooks() {
    printf("\n--- Available Books ---\n");
    for(int i = 0; i < totalBooks; i++) {
        printf("%d. %s\n", i+1, books[i]);
    }
}

// 2) Get total number of books (no arguments, return int)
int getTotalBooks() {
    return totalBooks;
}

// 3) Borrow a book (argument passed, no return value)
void borrowBook(char *bookName) {
    int found = -1;

    for(int i = 0; i < totalBooks; i++) {
        if(strcmp(books[i], bookName) == 0) {
            found = i;
            break;
        }
    }

    if(found != -1) {
        printf("\nBook Borrowed: %s\n", bookName);

        // Remove book from list → shift elements
        for(int j = found; j < totalBooks - 1; j++) {
            strcpy(books[j], books[j+1]);
        }

        totalBooks--;
        printf("Inventory Updated!\n");
    } else {
        printf("\nBook not available!\n");
    }
}

// 4) Calculate fine (argument passed, return value)
float calculateFine(int daysLate) {
    float fineRate = 2.0; // ₹2 per day
    return daysLate * fineRate;
}

int main() {
    printf("Library System\n");

    displayBooks();

    printf("\nTotal Books: %d\n", getTotalBooks());

    borrowBook("Data Structures");

    printf("\nAfter Borrow:\n");
    displayBooks();

    int late = 3;
    printf("\nFine for %d days late = %.2f\n", late, calculateFine(late));

    return 0;
}
```

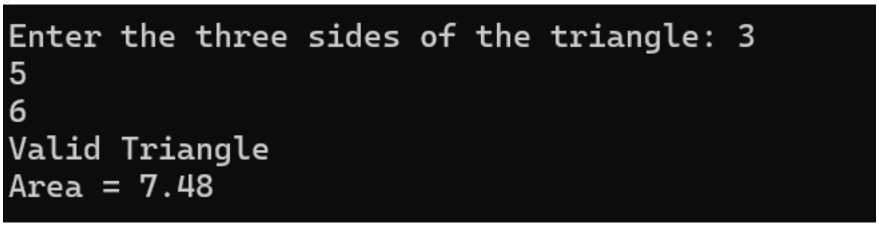


```
Library System

--- Available Books ---
1. C Programming
2. Data Structures
3. Database Systems
4. Operating Systems
5. Computer Networks

Total Books: 5

Book Borrowed: Data Structures
Inventory Updated!

After Borrow:

--- Available Books ---
1. C Programming
2. Database Systems
3. Operating Systems
4. Computer Networks

Fine for 3 days late = 6.00
```

PRACTICAL 26: Creating a program to assist the architech.

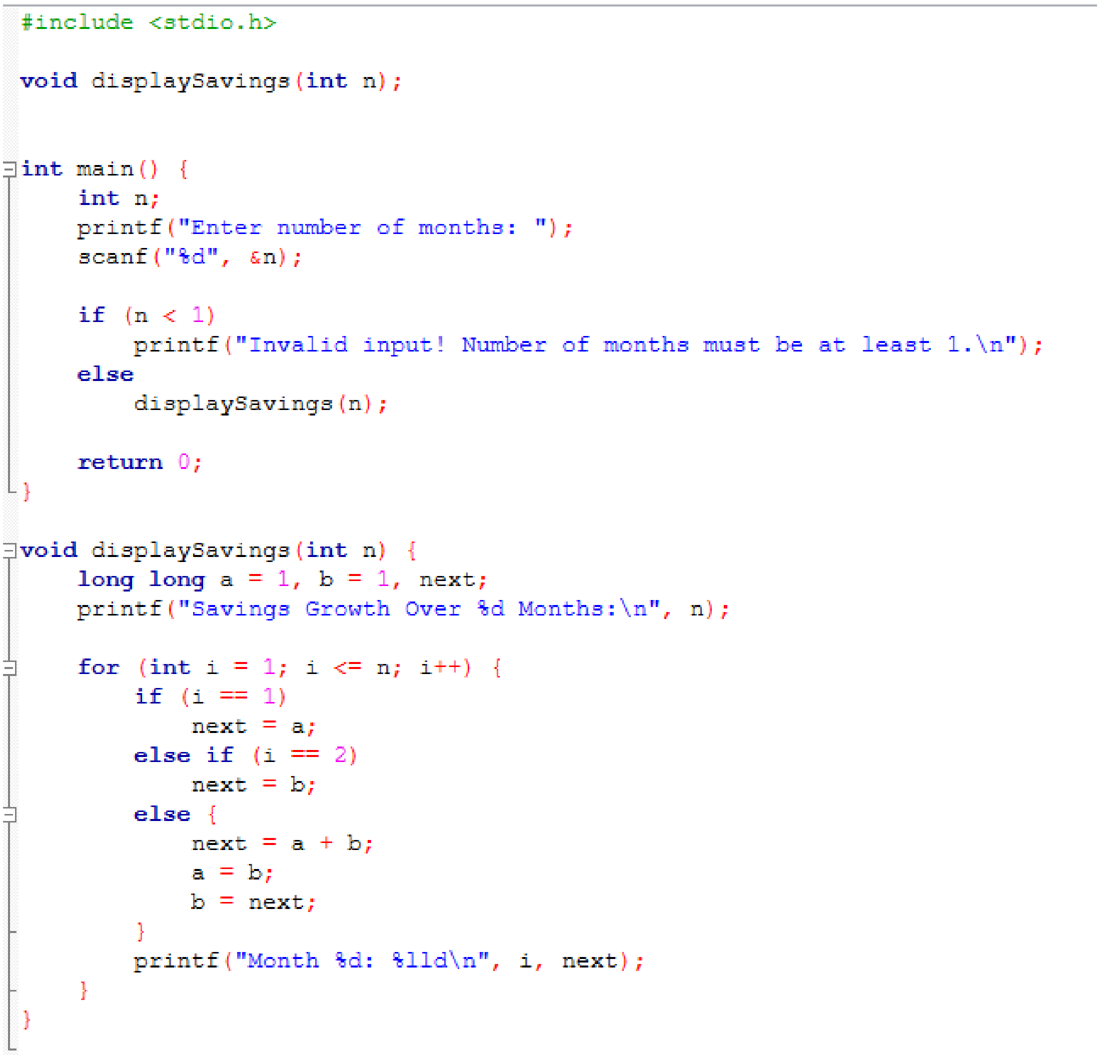


```c
#include <stdio.h>
#include <math.h>

int validate(float a, float b, float c);
float area(float a, float b, float c);


int main() {
    float a, b, c;
    printf("Enter the three sides of the triangle: ");
    scanf("%f %f %f", &a, &b, &c);
    if(validate(a, b, c)) {
        printf("Valid Triangle\n");
        printf("Area = %.2f\n", area(a, b, c));
    } else {
        printf("Invalid Triangle\n");
    }

    return 0;
}
```

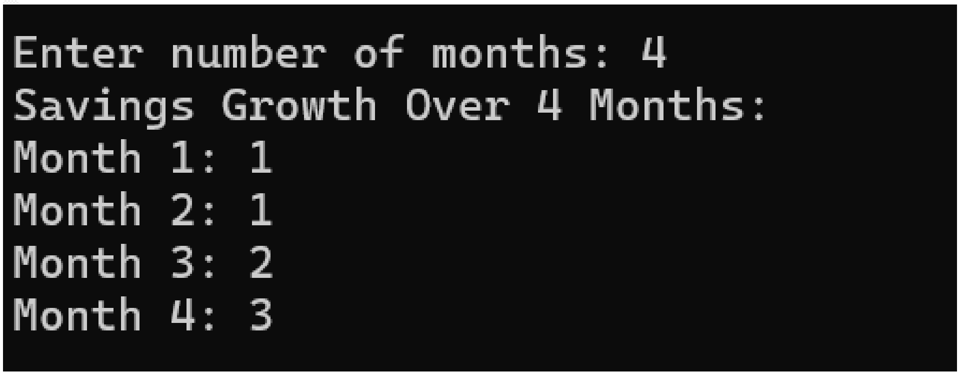


```
Enter the three sides of the triangle: 3
5
6
Valid Triangle
Area = 7.48
```

PRACTICAL 27: **The amount saved in a given month is often influenced by the sum of the savings from the two previous months. This growth pattern is similar to the Fibonacci series.**

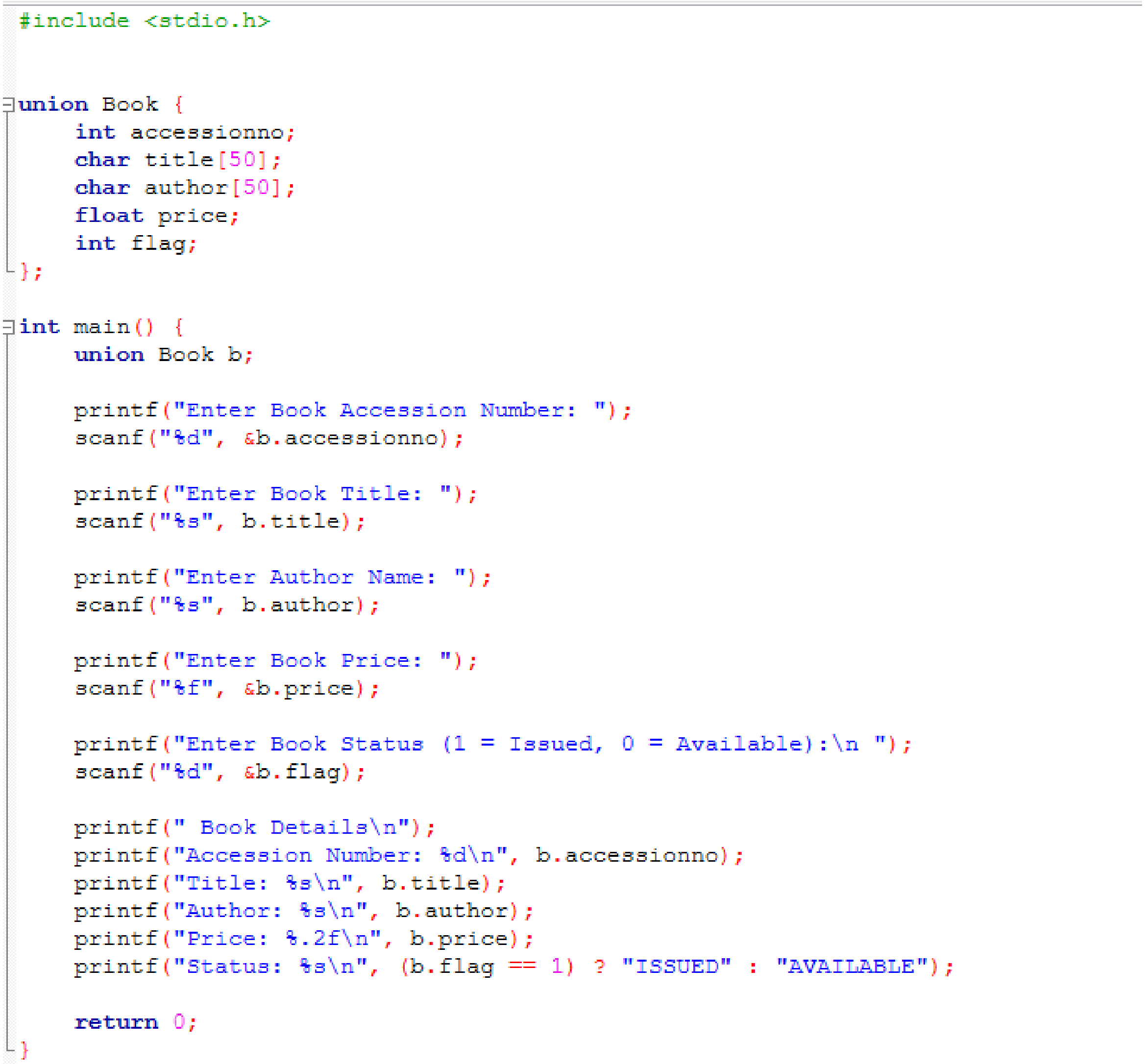


```c
#include <stdio.h>

void displaySavings(int n);


int main() {
    int n;
    printf("Enter number of months: ");
    scanf("%d", &n);

    if (n < 1)
        printf("Invalid input! Number of months must be at least 1.\n");
    else
        displaySavings(n);

    return 0;
}

void displaySavings(int n) {
    long long a = 1, b = 1, next;
    printf("Savings Growth Over %d Months:\n", n);

    for (int i = 1; i <= n; i++) {
        if (i == 1)
            next = a;
        else if (i == 2)
            next = b;
        else {
            next = a + b;
            a = b;
            b = next;
        }
        printf("Month %d: %lld\n", i, next);
    }
}
```

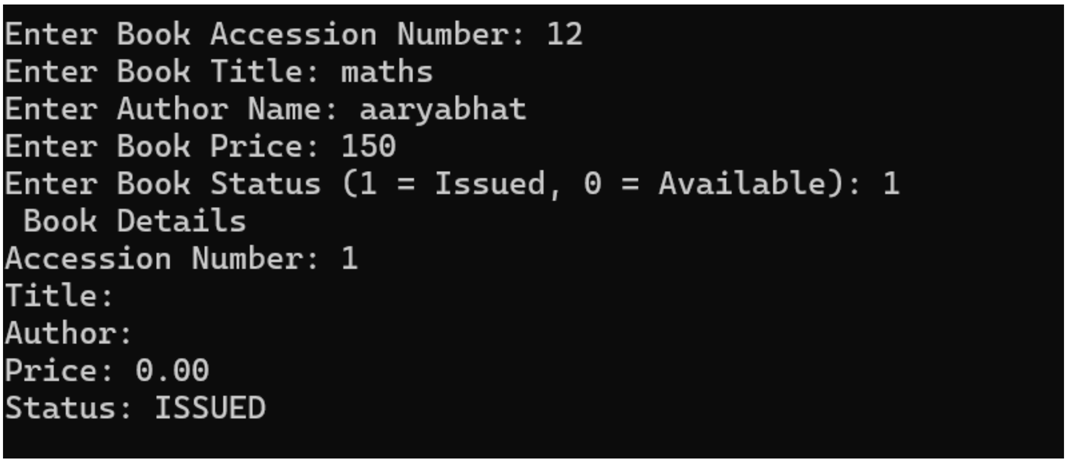


```
Enter number of months: 4
Savings Growth Over 4 Months:
Month 1: 1
Month 2: 1
Month 3: 2
Month 4: 3
```

PRACTICAL 28 : Library management system

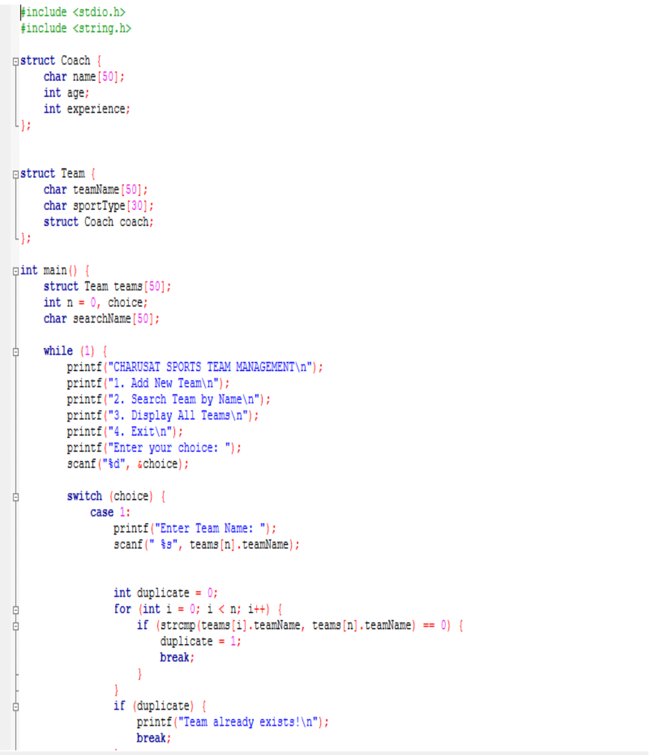


```c
#include <stdio.h>


union Book {
    int accessionno;
    char title[50];
    char author[50];
    float price;
    int flag;
};

int main() {
    union Book b;

    printf("Enter Book Accession Number: ");
    scanf("%d", &b.accessionno);

    printf("Enter Book Title: ");
    scanf("%s", b.title);

    printf("Enter Author Name: ");
    scanf("%s", b.author);

    printf("Enter Book Price: ");
    scanf("%f", &b.price);

    printf("Enter Book Status (1 = Issued, 0 = Available):\n ");
    scanf("%d", &b.flag);

    printf(" Book Details\n");
    printf("Accession Number: %d\n", b.accessionno);
    printf("Title: %s\n", b.title);
    printf("Author: %s\n", b.author);
    printf("Price: %.2f\n", b.price);
    printf("Status: %s\n", (b.flag == 1) ? "ISSUED" : "AVAILABLE");

    return 0;
}
```

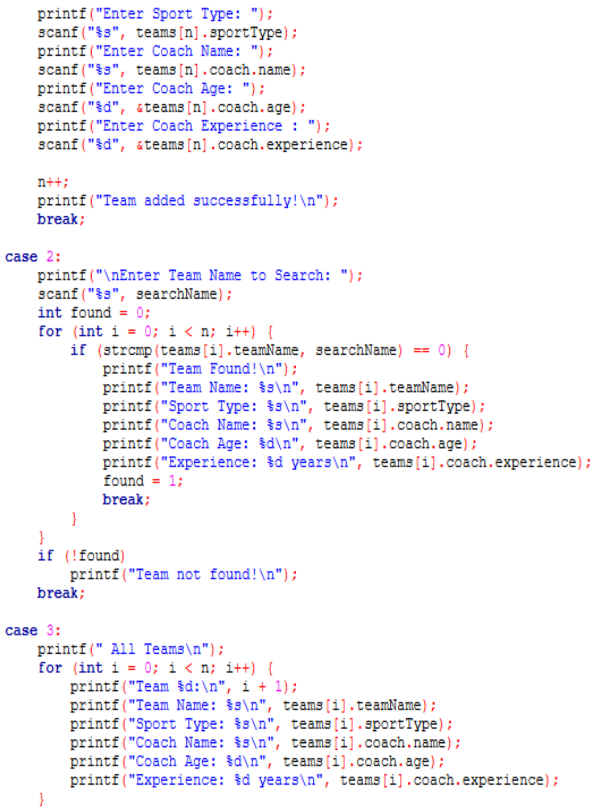


```
Enter Book Accession Number: 12
Enter Book Title: maths
Enter Author Name: aaryabhat
Enter Book Price: 150
Enter Book Status (1 = Issued, 0 = Available): 1
 Book Details
Accession Number: 1
Title:
Author:
Price: 0.00
Status: ISSUED
```

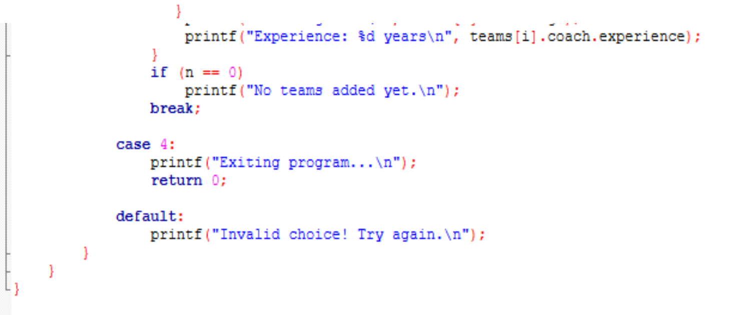
PRACTICAL 29 : C programming, where the team structure includes a sub-structure for coach information.



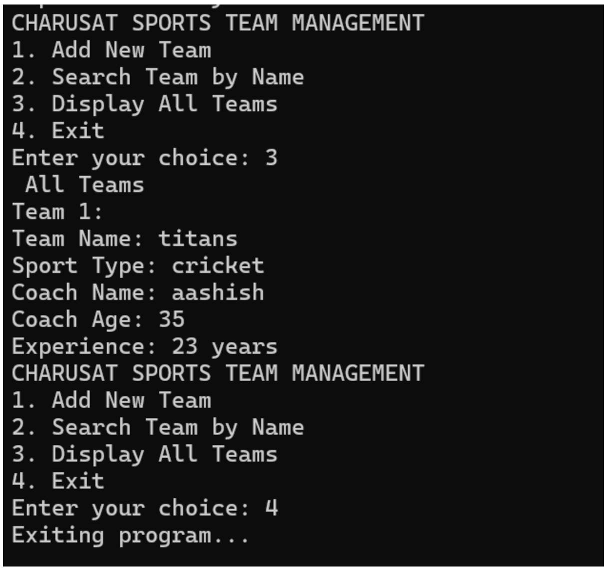
```c
#include <stdio.h>
#include <string.h>

struct Coach {
    char name[50];
    int age;
    int experience;
};


struct Team {
    char teamName[50];
    char sportType[30];
    struct Coach coach;
};

int main() {
    struct Team teams[50];
    int n = 0, choice;
    char searchName[50];

    while (1) {
        printf("CHARUSAT SPORTS TEAM MANAGEMENT\n");
        printf("1. Add New Team\n");
        printf("2. Search Team by Name\n");
        printf("3. Display All Teams\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter Team Name: ");
                scanf(" %s", teams[n].teamName);


                int duplicate = 0;
                for (int i = 0; i < n; i++) {
                    if (strcmp(teams[i].teamName, teams[n].teamName) == 0) {
                        duplicate = 1;
                        break;
                    }
                }
                if (duplicate) {
                    printf("Team already exists!\n");
                    break;
```



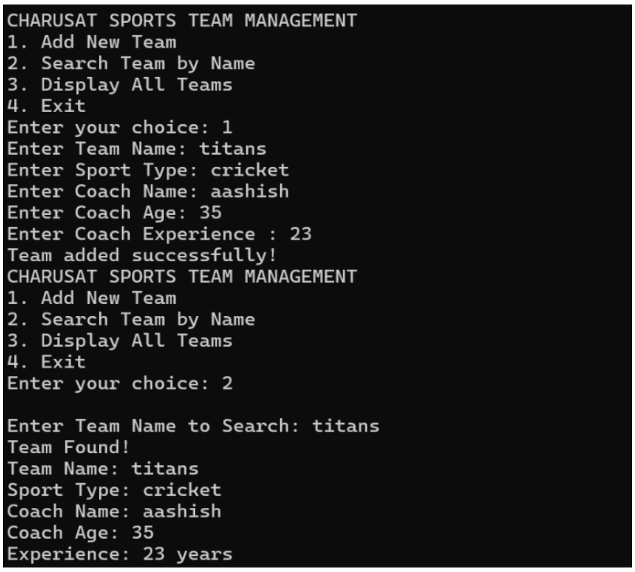
```c
                printf("Enter Sport Type: ");
                scanf("%s", teams[n].sportType);
                printf("Enter Coach Name: ");
                scanf("%s", teams[n].coach.name);
                printf("Enter Coach Age: ");
                scanf("%d", &teams[n].coach.age);
                printf("Enter Coach Experience : ");
                scanf("%d", &teams[n].coach.experience);

                n++;
                printf("Team added successfully!\n");
                break;

            case 2:
                printf("\nEnter Team Name to Search: ");
                scanf("%s", searchName);
                int found = 0;
                for (int i = 0; i < n; i++) {
                    if (strcmp(teams[i].teamName, searchName) == 0) {
                        printf("Team Found!\n");
                        printf("Team Name: %s\n", teams[i].teamName);
                        printf("Sport Type: %s\n", teams[i].sportType);
                        printf("Coach Name: %s\n", teams[i].coach.name);
                        printf("Coach Age: %d\n", teams[i].coach.age);
                        printf("Experience: %d years\n", teams[i].coach.experience);
                        found = 1;
                        break;
                    }
                }
                if (!found)
                    printf("Team not found!\n");
                break;

            case 3:
                printf(" All Teams\n");
                for (int i = 0; i < n; i++) {
                    printf("Team %d:\n", i + 1);
                    printf("Team Name: %s\n", teams[i].teamName);
                    printf("Sport Type: %s\n", teams[i].sportType);
                    printf("Coach Name: %s\n", teams[i].coach.name);
                    printf("Coach Age: %d\n", teams[i].coach.age);
                    printf("Experience: %d years\n", teams[i].coach.experience);
                }
```



```c
                    printf("Experience: %d years\n", teams[i].coach.experience);
                }
                if (n == 0)
                    printf("No teams added yet.\n");
                break;

            case 4:
                printf("Exiting program...\n");
                return 0;

            default:
                printf("Invalid choice! Try again.\n");
        }
    }
}
```

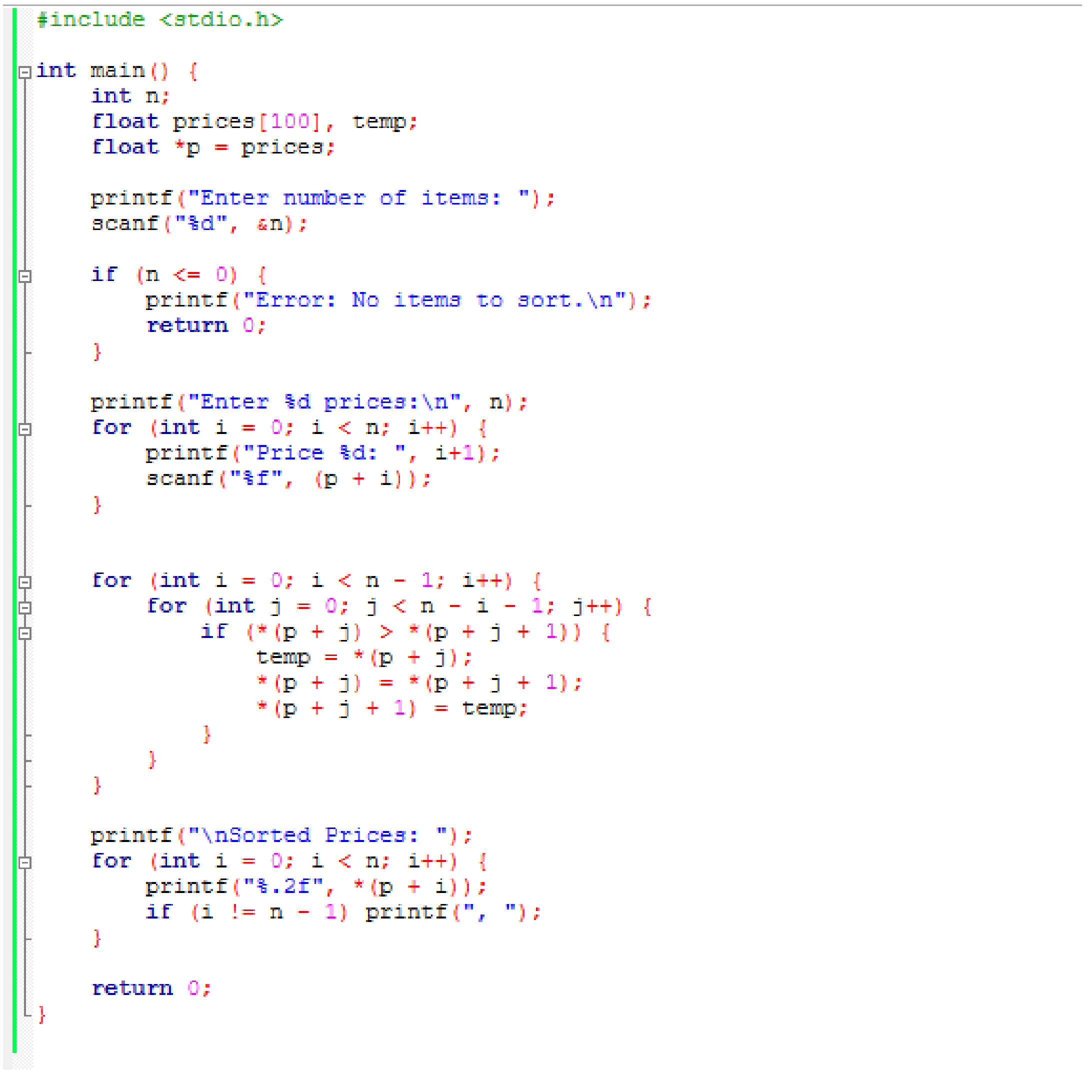


```
CHARUSAT SPORTS TEAM MANAGEMENT
1. Add New Team
2. Search Team by Name
3. Display All Teams
4. Exit
Enter your choice: 3
 All Teams
Team 1:
Team Name: titans
Sport Type: cricket
Coach Name: aashish
Coach Age: 35
Experience: 23 years
CHARUSAT SPORTS TEAM MANAGEMENT
1. Add New Team
2. Search Team by Name
3. Display All Teams
4. Exit
Enter your choice: 4
Exiting program...
```

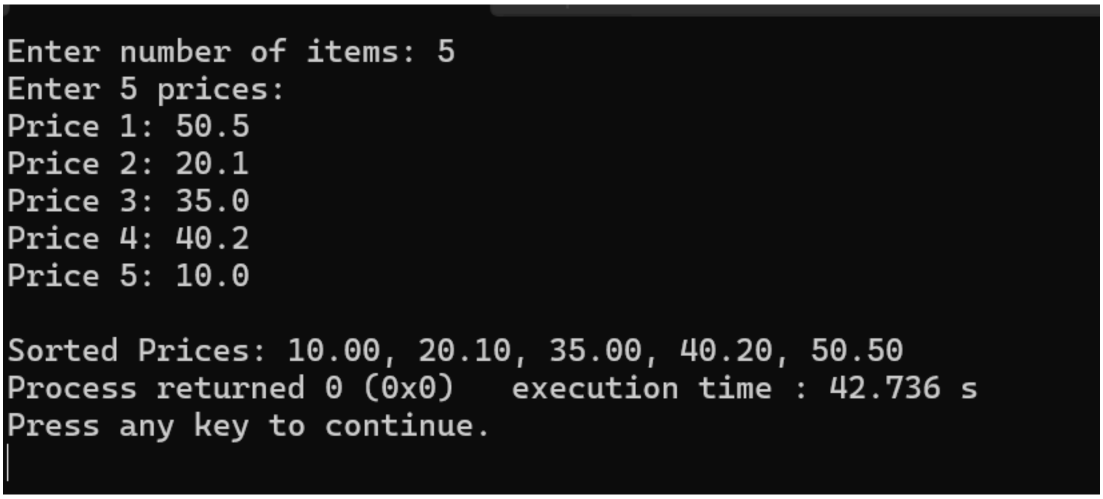


```
CHARUSAT SPORTS TEAM MANAGEMENT
1. Add New Team
2. Search Team by Name
3. Display All Teams
4. Exit
Enter your choice: 1
Enter Team Name: titans
Enter Sport Type: cricket
Enter Coach Name: aashish
Enter Coach Age: 35
Enter Coach Experience : 23
Team added successfully!
CHARUSAT SPORTS TEAM MANAGEMENT
1. Add New Team
2. Search Team by Name
3. Display All Teams
4. Exit
Enter your choice: 2

Enter Team Name to Search: titans
Team Found!
Team Name: titans
Sport Type: cricket
Coach Name: aashish
Coach Age: 35
Experience: 23 years
```

PRACTICAL 30 : C program to assist with this task using pointers.

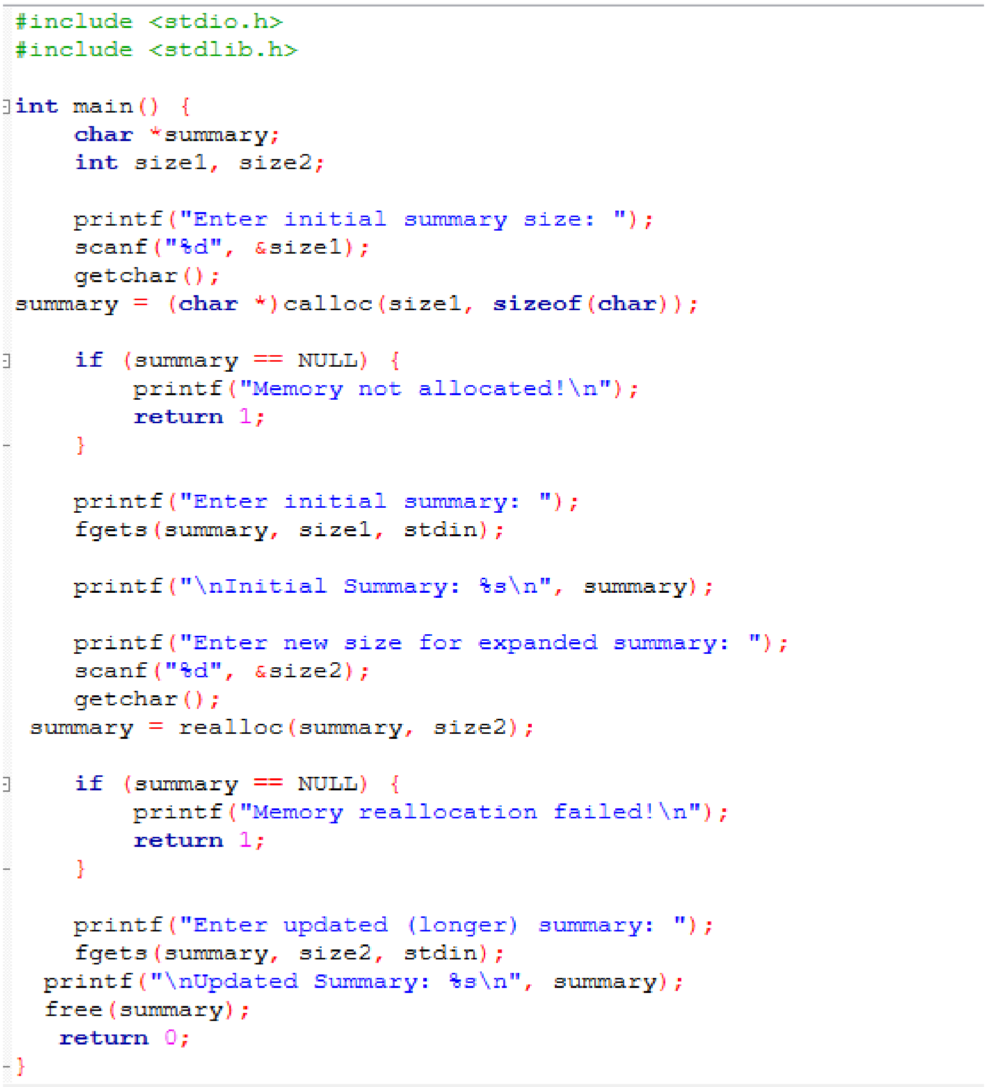


```c
#include <stdio.h>

int main() {
    int n;
    float prices[100], temp;
    float *p = prices;

    printf("Enter number of items: ");
    scanf("%d", &n);

    if (n <= 0) {
        printf("Error: No items to sort.\n");
        return 0;
    }

    printf("Enter %d prices:\n", n);
    for (int i = 0; i < n; i++) {
        printf("Price %d: ", i+1);
        scanf("%f", (p + i));
    }


    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (*(p + j) > *(p + j + 1)) {
                temp = *(p + j);
                *(p + j) = *(p + j + 1);
                *(p + j + 1) = temp;
            }
        }
    }

    printf("\nSorted Prices: ");
    for (int i = 0; i < n; i++) {
        printf("%.2f", *(p + i));
        if (i != n - 1) printf(", ");
    }

    return 0;
}
```

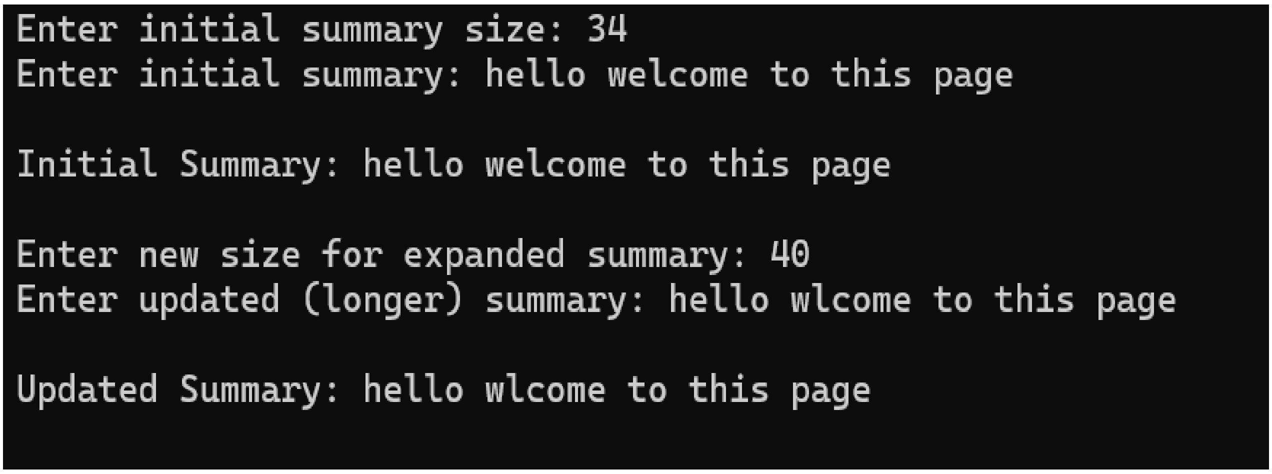


```
Enter number of items: 5
Enter 5 prices:
Price 1: 50.5
Price 2: 20.1
Price 3: 35.0
Price 4: 40.2
Price 5: 10.0

Sorted Prices: 10.00, 20.10, 35.00, 40.20, 50.50
Process returned 0 (0x0)   execution time : 42.736 s
Press any key to continue.
```

PRACTICAL 31 : The system must dynamically adjust the memory allocation to accommodate the longer text.

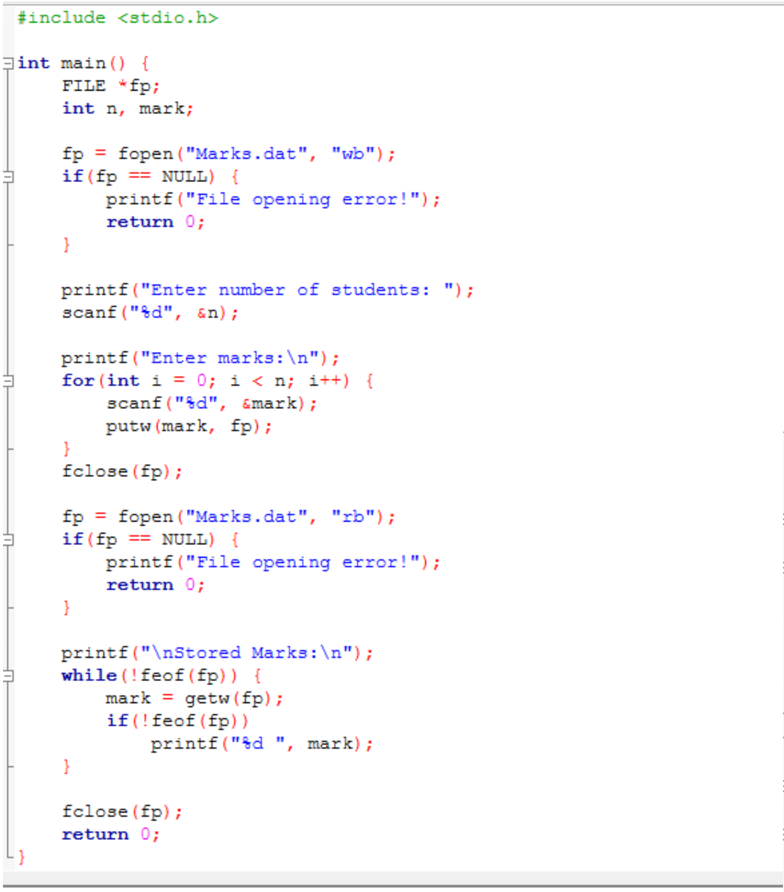


```c
#include <stdio.h>
#include <stdlib.h>

int main() {
    char *summary;
    int size1, size2;

    printf("Enter initial summary size: ");
    scanf("%d", &size1);
    getchar();
summary = (char *)calloc(size1, sizeof(char));

    if (summary == NULL) {
        printf("Memory not allocated!\n");
        return 1;
    }

    printf("Enter initial summary: ");
    fgets(summary, size1, stdin);

    printf("\nInitial Summary: %s\n", summary);

    printf("Enter new size for expanded summary: ");
    scanf("%d", &size2);
    getchar();
 summary = realloc(summary, size2);

    if (summary == NULL) {
        printf("Memory reallocation failed!\n");
        return 1;
    }

    printf("Enter updated (longer) summary: ");
    fgets(summary, size2, stdin);
  printf("\nUpdated Summary: %s\n", summary);
  free(summary);
   return 0;
}
```

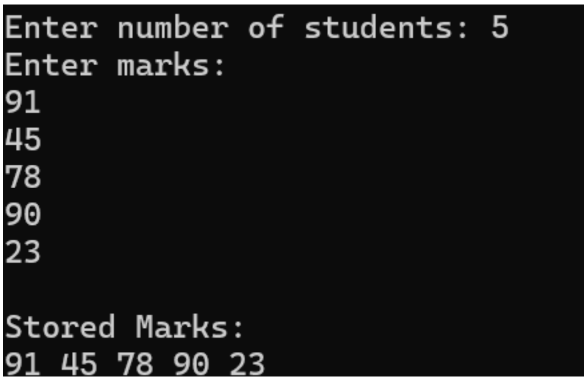


```
Enter initial summary size: 34
Enter initial summary: hello welcome to this page

Initial Summary: hello welcome to this page

Enter new size for expanded summary: 40
Enter updated (longer) summary: hello wlcome to this page

Updated Summary: hello wlcome to this page
```

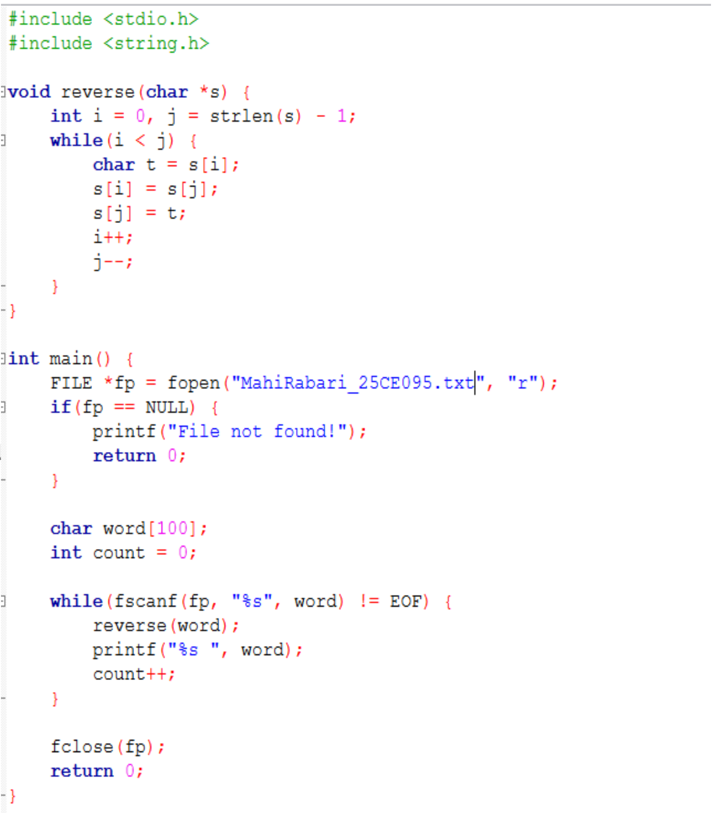
PRACTICAL 32 :



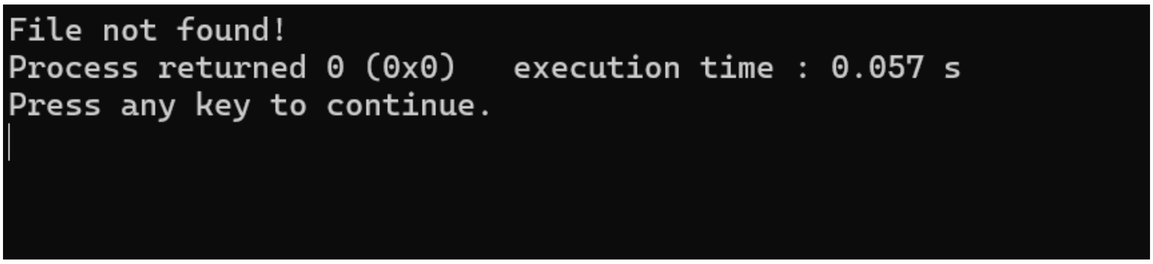
```c
#include <stdio.h>

int main() {
    FILE *fp;
    int n, mark;

    fp = fopen("Marks.dat", "wb");
    if(fp == NULL) {
        printf("File opening error!");
        return 0;
    }

    printf("Enter number of students: ");
    scanf("%d", &n);

    printf("Enter marks:\n");
    for(int i = 0; i < n; i++) {
        scanf("%d", &mark);
        putw(mark, fp);
    }
    fclose(fp);

    fp = fopen("Marks.dat", "rb");
    if(fp == NULL) {
        printf("File opening error!");
        return 0;
    }

    printf("\nStored Marks:\n");
    while(!feof(fp)) {
        mark = getw(fp);
        if(!feof(fp))
            printf("%d ", mark);
    }

    fclose(fp);
    return 0;
}
```



```
Enter number of students: 5
Enter marks:
91
45
78
90
23

Stored Marks:
91 45 78 90 23
```



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

void reverse(char *s) {
    int i = 0, j = strlen(s) - 1;
    while(i < j) {
        char t = s[i];
        s[i] = s[j];
        s[j] = t;
        i++;
        j--;
    }
}

int main() {
    FILE *fp = fopen("MahiRabari_25CE095.txt", "r");
    if(fp == NULL) {
        printf("File not found!");
        return 0;
    }

    char word[100];
    int count = 0;

    while(fscanf(fp, "%s", word) != EOF) {
        reverse(word);
        printf("%s ", word);
        count++;
    }

    fclose(fp);
    return 0;
}
```



```
File not found!
Process returned 0 (0x0)   execution time : 0.057 s
Press any key to continue.
```