

C programming lab questions with answers

1.A conservationist is studying a rare species of animals whose population follows the expression $F_n = F_{n-1} + F_{n-2}$, where $n \geq 2$, with $F_0=0$ and $F_1=1$. Starting with the population sizes from the first two years, the researcher aims to predict the population for the next n years to make conservation plans. The task involves dynamically calculating future values based on a growth sequence. The predictions must be accurate and scalable as the time range increases. How would you design a C program to compute and display these projections efficiently?

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n, i; long int F0 = 0, F1 = 1, Fn;
    clrscr();

    printf("Enter the number of years for population prediction:");
    scanf("%d", &n);

    if(n < 1) {
        printf("Number of years must be at least 1");
        getch();
        return;
    }
```

```

printf("\nPopulation predictions for the next %d years:\n",
n);
if(n == 1) {
    printf("Year 1: %ld\n", F0);
} else {
    printf("Year 1: %ld\n", F0);
    printf("Year 2: %ld\n", F1);
    for(i = 3; i <= n; i++) {
        Fn = F0 + F1;
        printf("Year %d: %ld\n", i, Fn);
        F0 = F1;
        F1 = Fn;
    }
}

getch();
}

```

2. A mathematician is working on a research project involving two datasets represented as 2D array. To analyze these datasets further, the mathematician needs to combine the values of these datasets in such a way each element on combining gives the sum of the elements. Write a C program to accept the inputs, verify the datasets compatibility to perform the operation and finally display the result.

```

#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();

    printf("Enter the number of rows and columns for the first da

```

```

taset: ");
scanf("%d %d", &rows1, &cols1);

printf("Enter the number of rows and columns for the second d
ataset: ");
scanf("%d %d", &rows2, &cols2);

if (rows1 != rows2 || cols1 != cols2)
{
    printf("Error: Datasets are incompatible for additio
n.\n");
    getch();
    return;
}

int dataset1[rows1][cols1], dataset2[rows2][cols2], result[ro
ws1][cols1];

printf("Enter the elements of the first dataset:\n");
for (i = 0; i < rows1; i++) {
    for (j = 0; j < cols1; j++) {
        printf("Element [%d][%d]: ", i, j);
        scanf("%d", &dataset1[i][j]);
    }
}

printf("Enter the elements of the second dataset:\n");
for (i = 0; i < rows2; i++) {
    for (j = 0; j < cols2; j++) {
        printf("Element [%d][%d]: ", i, j);
        scanf("%d", &dataset2[i][j]);
    }
}

```

```

for (i = 0; i < rows1; i++) {
    for (j = 0; j < cols1; j++) {
        result[i][j] = dataset1[i][j] + dataset2[i][j];
    }
}

printf("Resulting dataset after addition:\n");
for (i = 0; i < rows1; i++) {
    for (j = 0; j < cols1; j++) {
        printf("%d ", result[i][j]);
    }
    printf("\n");
}

getch();
}

```

3. A factory manager struggles with tracking the daily production of widgets over a month. Each day's production varies, and the manager wants a way to display these numbers systematically and analyze trends over time. The challenge lies in automating the recording process and ensuring scalability for any given number of days. The output should provide clarity for better decision-making in adjusting production rates. How would you create a program to handle this efficiently?

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int n, i, maxDay = 0, minDay = 0; int production[31], max, min;
    float total = 0, average;
    clrscr();

```

```

printf("Enter the number of days in the month: ");
scanf("%d", &n);

if(n < 1 || n > 31) {
    printf("Invalid number of days. Enter between 1 and 31.\n");
    getch();
    return;
}

printf("Enter the daily production numbers:\n");
for(i = 0; i < n; i++) {
    printf("Day %d: ", i + 1);
    scanf("%d", &production[i]);
    total += production[i];
}

max = production[0];
min = production[0];

for(i = 1; i < n; i++) {
    if(production[i] > max) {
        max = production[i];
        maxDay = i;
    }
    if(production[i] < min) {
        min = production[i];
        minDay = i;
    }
}

average = total / n;

printf("Production Analysis\n");
printf("Total Production: %.2f\n", total);
printf("Average Production: %.2f\n", average);

```

```

printf("Maximum Production: %d on Day %d\n", max, maxDay +
1);
printf("Minimum Production: %d on Day %d\n", min, minDay +
1);

getch();
}

```

4. A space mission involves a critical countdown timer that begins at 10 seconds and ends with the word "Liftoff!" Every second needs to be displayed sequentially, and the final output should signal the start of the mission. The task requires a robust program that simulates the countdown with precision and handles edge cases, such as system interruptions. How would you develop such a timer, ensuring accuracy and reliability in a high-stakes scenario?

```

#include<stdio.h>
#include<conio.h>
void main()
{
int i;
clrscr();

printf("Countdown Timer:\n");
for(i = 10; i > 0; i--) {
    printf("%d\n", i);
    delay(1000);
}
printf("Liftoff!\n");

getch();
}

```

5. A high-tech water management facility uses pumps to fill a tank at a constant rate of 10 liters per minute. The system must stop when the tank reaches its maximum capacity, but real-time tracking of water levels has become challenging. The facility staff requires a program to simulate the filling process, update the water level regularly, and terminate once full. The program should handle various tank capacities and ensure no overflows occur. How would you implement this simulation?

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int tankCapacity, currentLevel = 0;
    clrscr();

    printf("Enter the maximum tank capacity (in liters): ");
    scanf("%d", &tankCapacity);

    printf("Filling the tank...\n");

    while(currentLevel < tankCapacity) {
        currentLevel += 10;
        if(currentLevel > tankCapacity) {
            currentLevel = tankCapacity;
        }
        printf("Current water level: %d liters\n", currentLevel);
        delay(1000);
    }

    printf("Tank is full. Filling process stopped.\n");

    getch();
}
```

6. A chess tournament is being held where three players compete across three rounds, earning points based on their performance. The winner is determined by the highest total score, but ties and varying inputs have made manual calculations unreliable. The organizers want a program that dynamically computes the total scores for each player and announces the winner. Additionally, they need to see all the players' scores displayed for transparency. How would you solve this problem?

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

void main() {
    int scores[3][3], total[3] = {0, 0, 0};
    int i, j, winner, maxScore = 0;
    clrscr();

    printf("Enter the scores for 3 players across 3 rounds:\n");
    for(i = 0; i < 3; i++) {
        printf("Player %d:\n", i + 1);
        for(j = 0; j < 3; j++) {
            printf("  Round %d: ", j + 1);
            scanf("%d", &scores[i][j]);
            total[i] += scores[i][j];
        }
    }

    printf("Total scores:\n");
    for(i = 0; i < 3; i++) {
        printf("Player %d: %d\n", i + 1, total[i]);
        if(total[i] > maxScore) {
            maxScore = total[i];
            winner = i + 1;
        }
    }
}
```



```
printf("Winner: Player %d with %d points\n", winner, maxScore);

getch();
}
```

7. In a bustling city like Delhi, traffic authorities have decided to categorize the travel on road based on their license plates. All even numbered vehicles must avail on road travel only on Mondays, Wednesdays and Friday. Whereas odd numbered vehicles must board only on Tuesdays, Thursdays and Saturdays. On Sundays no private vehicles must board on road. In this regard Government issues sticker based on the travel days. Vehicles travelling on Mondays, Wednesdays and Friday are issued Dark Green coloured sticker and the other batch are issued with Parrot Green colour. Help the Government in automatic identification of sticker colour based on the constraints mentioned above through a program code.

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

void main() {
int licensePlate, day;
```

```
#include<stdio.h>
#include<conio.h>
void main()
{
int licensePlate, day;
clrscr();

printf("Enter the license plate number: ");
scanf("%d", &licensePlate);

printf("Enter the day of the week (1 for Monday, 2 for Tuesday, 3 for Wednesday, 4 for Thursday, 5 for Friday, 6 for Saturday, 7 for Sunday)");
```

```

y, ..., 7 for Sunday): ");
scanf("%d", &day);

if (day == 7) {
    printf("No private vehicles are allowed on the road on Su
ndays.\n");
} else if ((licensePlate % 2 == 0 && (day == 1 || day == 3 ||
day == 5))) {
    printf("Sticker Color: Dark Green\n");
} else if ((licensePlate % 2 != 0 && (day == 2 || day == 4 ||
day == 6))) {
    printf("Sticker Color: Parrot Green\n");
} else {
    printf("This vehicle is not allowed on the road on this d
ay.\n");
}

getch();
}

```

8. A bank approves loan applications based on specific criteria: applicants must have an annual income of at least ₹3,00,000 and no bad credit history. With thousands of applications pouring in, manual evaluations are error-prone and time-consuming. The bank seeks an automated program that evaluates eligibility dynamically and provides instant decisions. The system should be robust enough to handle exceptions, such as incomplete data. How would you create such a solution?

```

#include<stdio.h>
#include<conio.h>
void main()
{
    float annualIncome;

```

```

int badCreditHistory;
clrscr();

printf("Enter the annual income (in ₹): ");
scanf("%f", &annualIncome);

printf("Does the applicant have bad credit history? (1 for Yes, 0 for No): ");
scanf("%d", &badCreditHistory);

if (annualIncome <= 0)
{
    printf("Error: Incomplete or invalid data provided for annual income.\n");
}
else if (badCreditHistory != 0 && badCreditHistory != 1)
{
    printf("Error: Incomplete or invalid data provided for credit history.\n");
}
else if (annualIncome >= 300000 && badCreditHistory == 0)
{
    printf("Loan Approved.\n");
}
else
{
    printf("Loan Denied.\n");
}

getch();
}

```

9. An architect designing a series of temple pillars notices a pattern: the number of decorative designs on each pillar increases following the expression $F_n = F_{n-1} + F_{n-2}$, where $n \geq 2$, with $F_0=0$ and $F_1=1$. To plan effectively, the architect requires a program that calculates and displays the number of designs for the first

n pillars. The solution must be scalable, as the number of pillars often changes during the design phase. How would you design such a tool to assist the architect in their planning process?

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n, i;
    int f0 = 0, f1 = 1, fn;
    clrscr();

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

    if (n <= 0) {
        printf("Invalid input. Number of pillars must be greater than 0.\n");
    } else {
        printf("Number of designs on the first %d pillars:\n", n);
        if (n >= 1) {
            printf("%d ", f0);
        }
        if (n >= 2) {
            printf("%d ", f1);
        }

        for (i = 3; i <= n; i++) {
            fn = f0 + f1;
            printf("%d ", fn);
            f0 = f1;
            f1 = fn;
        }
    }
}
```

```

        printf("\n");
    }

    getch();
}

```

10. A restaurant faces delays in generating bills for customers due to the manual calculation of prices and GST. Each dish has a specific price, and the total bill is subject to a 5% GST. Customers often order multiple items, making manual calculations prone to errors. The restaurant needs a program to automate the entire billing process, including summing up prices and applying GST. How would you create such a solution to eliminate delays and improve efficiency?

```

#include<stdio.h>
#include<conio.h>
void main()

{
    int n, i; float price, total = 0, gst, finalBill;
    clrscr();

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

    if (n <= 0) {
        printf("Invalid input. The number of items must be greater than 0.\n");
    } else {
        printf("Enter the prices of the %d items:\n", n);

        for (i = 1; i <= n; i++) {
            printf("Price of item %d: ", i);
            scanf("%f", &price);

```

```

        if (price < 0) {
            printf("Invalid price. Please enter a non-negative value.\n");
            i--;
            continue;
        }
        total += price;
    }

    gst = total * 0.05;
    finalBill = total + gst;

    printf("Total price before GST: ₹%.2f\n", total);
    printf("GST (5%): ₹%.2f\n", gst);
    printf("Final bill amount: ₹%.2f\n", finalBill);
}

getch();
}

```

11. An urban planner proposes planting trees along a newly constructed highway to beautify the area. However, the planting is restricted to plots with even numbers due to budget constraints. With hundreds of plots to evaluate, the planner needs a program that dynamically identifies and lists all the eligible plot numbers. The solution must work for any number of plots to adapt to future projects. How would you design this system?

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int totalPlots, i;
    clrscr();

```

```

printf("Enter the total number of plots: ");
scanf("%d", &totalPlots);

if (totalPlots <= 0) {
    printf("Invalid input. The number of plots must be greater than 0.\n");
} else {
    printf("Eligible plots for planting trees (even-numbered plots):\n");
    for (i = 1; i <= totalPlots; i++) {
        if (i % 2 == 0) {
            printf("%d ", i);
        }
    }
    printf("\n");
}

getch();
}

```

12. A coding competition rewards participants for solving even-numbered problems, causing confusion among contestants about which problems qualify. With hundreds of problems in the contest, participants need a tool to quickly generate a list of all valid problem numbers. The program must dynamically calculate this list based on the total number of problems, n . How would you create such a solution to help participants strategize effectively?

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int totalProblems, i;

```

```

clrscr();

printf("Enter the total number of problems in the contest:
");
scanf("%d", &totalProblems);

if (totalProblems <= 0) {
    printf("Invalid input. The number of problems must be gre
ater than 0.\n");
} else {
    printf("Valid problem numbers (even-numbered problem
s):\n");
    for (i = 1; i <= totalProblems; i++) {
        if (i % 2 == 0) {
            printf("%d ", i);
        }
    }
    printf("\n");
}

getch();
}

```

13. A sports team manager registers jersey numbers for players before a match. The manager insists on displaying these numbers in ascending order for easy reference, but the current registration system outputs them randomly. To solve this, they need a program that accepts the jersey numbers, sorts them, and displays the sorted list. The program must handle varying numbers of players for different matches. How would you implement such a system?

```

#include<stdio.h>
#include<conio.h>

```



```

void main()
{
int jerseyNumbers[100],n,i,j,
temp;
clrscr();

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

if (n <= 0 || n > 100) {
    printf("Invalid input. The number of players must be between 1 and 100.\n");
} else {
    printf("Enter the jersey numbers of the players:\n");
    for (i = 0; i < n; i++) {
        scanf("%d", &jerseyNumbers[i]);
    }

    for (i = 0; i < n - 1; i++) {
        for (j = i + 1; j < n; j++) {
            if (jerseyNumbers[i] > jerseyNumbers[j]) {
                temp = jerseyNumbers[i];
                jerseyNumbers[i] = jerseyNumbers[j];
                jerseyNumbers[j] = temp;
            }
        }
    }

    printf("Jersey numbers in ascending order:\n");
    for (i = 0; i < n; i++) {
        printf("%d ", jerseyNumbers[i]);
    }
    printf("\n");
}
}

```

```
getch();  
}
```

14. A weather station monitors daily temperatures for a week to identify trends and anomalies. The station staff wants a program to compute the average temperature and determine the hottest and coldest days. With fluctuating readings, the calculations must be automated to avoid errors. The solution must also accommodate longer timeframes, such as monthly data. How would you design such a program?

```
#include<stdio.h>  
#include<conio.h>  
void main()  
{  
    int i, n;  
    float temperatures[31], sum = 0.0, average, maxTemp, minTemp;  
  
    clrscr();  
  
    printf("Enter the number of days (max 31): ");  
    scanf("%d", &n);  
  
    if (n <= 0 || n > 31) {  
        printf("Invalid input. The number of days must be between  
1 and 31.\n");  
    } else {  
        printf("Enter the temperatures for %d days:\n", n);  
        for (i = 0; i < n; i++) {  
            scanf("%f", &temperatures[i]);  
            sum += temperatures[i];  
        }  
  
        average = sum / n;
```

```

maxTemp = temperatures[0];
minTemp = temperatures[0];

for (i = 1; i < n; i++) {
    if (temperatures[i] > maxTemp) {
        maxTemp = temperatures[i];
    }
    if (temperatures[i] < minTemp) {
        minTemp = temperatures[i];
    }
}

printf("Average Temperature: %.2f\n", average);
printf("Hottest Day Temperature: %.2f\n", maxTemp);
printf("Coldest Day Temperature: %.2f\n", minTemp);
}

getch();
}

```

15. A fire alarm system in a skyscraper categorizes temperatures into three levels: normal, warning, and danger. Based on the category, the system triggers alerts or activates sprinklers to ensure safety. The management needs a program that evaluates input temperatures dynamically and determines the appropriate category. The program should be easy to update as thresholds may change over time. How would you design this system?

```

#include<stdio.h>
#include<conio.h>
void main()
{
    float temperature; float normalLimit = 35.0, warningLimit = 5

```

```

0.0;
clrscr();

printf("Enter the current temperature: ");
scanf("%f", &temperature);

if (temperature < 0) {
    printf("Invalid input. Temperature cannot be negative.\n");
} else if (temperature <= normalLimit) {
    printf("Temperature Level: Normal\n");
    printf("Status: System Stable.\n");
} else if (temperature <= warningLimit) {
    printf("Temperature Level: Warning\n");
    printf("Status: Trigger Warning Alert!\n");
} else {
    printf("Temperature Level: Danger\n");
    printf("Status: Activate Sprinklers Immediately!\n");
}

getch();
}

```

16. A supermarket calculates bills by summing up the prices of purchased items and applying a fixed GST rate of 18%. Customers often buy multiple items, and manual calculations lead to errors during busy hours. The supermarket seeks a program that automates the billing process, ensuring accuracy even during high customer traffic. The program must handle varying GST rates for promotional periods. How would you design such a solution?

```

#include<stdio.h>
#include<conio.h>
void main()

```

```

{
    int numItems;
    float price, total = 0.0, gstRate, gstAmount, finalAmount;
    clrscr();

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

    printf("Enter the GST rate (in percentage, e.g., 18 for 18%
    %): ");
    scanf("%f", &gstRate);

    for (int i = 1; i <= numItems; i++) {
        printf("Enter price for item %d: ", i);
        scanf("%f", &price);
        total += price;
    }
    gstAmount = (gstRate / 100) *
    finalAmount = total + gstAmount;

    printf("Total price before GST: %.2f\n", total);
    printf("GST @ %.2f%%: %.2f\n", gstRate, gstAmount);
    printf("Total price after GST: %.2f\n", finalAmount);

    getch();
}

```

17. A prestigious art gallery hosts an exhibition where every painting is taxed at 18% GST. Buyers often purchase multiple items, making it difficult to compute the total payable amount manually. The gallery requires a program that dynamically calculates the total price, including GST, for any number of selected items. How would you create a system to assist the staff in streamlining this process?

```

#include<stdio.h>
#include<conio.h>

void main() {
    int numItems;
    float price, total = 0.0, gstRate = 18.0, gstAmount, finalAmount;
    clrscr();

    printf("Enter the number of paintings: ");
    scanf("%d", &numItems);

    for (int i = 1; i <= numItems; i++) {
        printf("Enter price for painting %d: ", i);
        scanf("%f", &price);
        total += price;
    }

    gstAmount = (gstRate / 100) * total;
    finalAmount = total + gstAmount;

    printf("Total price before GST: %.2f\n", total);
    printf("GST @ %.2f%%: %.2f\n", gstRate, gstAmount);
    printf("Total price after GST: %.2f\n", finalAmount);

    getch();
}

```

18. A company tracks its employees' salaries and wants to reward the top three earners with bonuses. Sorting through hundreds of salary records manually has proven inefficient and error-prone. The company seeks a program that accepts

employee names and salaries, sorts them, and displays the top three highest earners. How would you design this system to ensure speed and accuracy?

```
#include<stdio.h>
#include<conio.h>
struct Employee {char name[50];
float salary;};
void main()
{
int n;struct Employee emp[100], temp;
clrscr();

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

for(int i = 0; i < n; i++) {
    printf("\nEnter name of employee %d: ", i + 1);
    fflush(stdin);
    gets(emp[i].name);
    printf("Enter salary of %s: ", emp[i].name);
    scanf("%f", &emp[i].salary);
}

for(int i = 0; i < n - 1; i++) {
    for(int j = i + 1; j < n; j++) {
        if(emp[i].salary < emp[j].salary) {
            temp = emp[i];
            emp[i] = emp[j];
            emp[j] = temp;
        }
    }
}

printf("Top 3 highest earners:\n");
```

```

for(int i = 0; i < 3 && i < n; i++) {
    printf("%d. %s - %.2f\\n", i + 1, emp[i].name, emp[i].salary);
}

getch();
}

```

19. An architect is designing a square floor plan and wants to calculate the total length of decorative tiles along the main diagonal of the plan. The plan is represented as a square matrix, with each value indicating the length of tiles in a section. Write a C program to accept the square matrix, compute the sum of its diagonal elements, and display the result to help the architect finalize the design.

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int n, i, j, sum = 0;
    clrscr();

    printf("Enter the size of the matrix: ");
    scanf("%d", &n);

    int matrix[n][n];

    printf("Enter the elements of the matrix:\\n");
    for(i = 0; i < n; i++) {
        for(j = 0; j < n; j++) {
            scanf("%d", &matrix[i][j]);
        }
    }
}

```



```

for(i = 0; i < n; i++) {
    sum += matrix[i][i];
}

printf("The sum of the diagonal elements is: %d", sum);

getch();
}

```

20. An online bookstore categorizes books into three price ranges: under ₹500, between ₹500 and ₹1000, and above ₹1000. Customers often want to browse by these categories, but the current system lacks such a feature. The bookstore seeks a program that organizes books into the defined categories and displays them efficiently. How would you implement this system to enhance customer satisfaction

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int n, i;
    float price;
    clrscr();

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

    float prices[n];

    printf("Enter the prices of the books:\n");
    for(i = 0; i < n; i++) {
        scanf("%f", &prices[i]);
    }
}

```

```

printf("Books under ₹500:\n");
for(i = 0; i < n; i++) {
    if(prices[i] < 500) {
        printf("₹%.2f\n", prices[i]);
    }
}

printf("Books between ₹500 and ₹1000:\n");
for(i = 0; i < n; i++) {
    if(prices[i] >= 500 && prices[i] <= 1000) {
        printf("₹%.2f\n", prices[i]);
    }
}

printf("Books above ₹1000:\n");
for(i = 0; i < n; i++) {
    if(prices[i] > 1000) {
        printf("₹%.2f\n", prices[i]);
    }
}

getch();
}

```

21. An online learning platform stores course details, including names, categories, and ratings. Students frequently search for courses in specific categories, but the platform lacks a way to filter and rank courses easily. The platform seeks a program that allows users to search for courses by category and displays the top-rated ones. How would you design such a tool to improve the browsing experience?

```

#include<stdio.h>
#include<conio.h>
#include<string.h>
struct Course
{
char name[50];
char category[30];
float rating;
}
void main()
{
int n, i, j;
char searchCategory[30];
struct Coursecourses[50];
struct Coursetemp;
clrscr();

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

printf("Enter the course details (name, category, rating):\n");
for(i = 0; i < n; i++) {
printf("Course %d\n", i + 1);
printf("Enter name: ");
getchar();
fgets(courses[i].name, 50, stdin);
courses[i].name[strcspn(courses[i].name, "\n")] = 0;

printf("Enter category: ");
fgets(courses[i].category, 30, stdin);
courses[i].category[strcspn(courses[i].category, "\n")] =
0;

printf("Enter rating: ");

```

```

        scanf("%f", &courses[i].rating);
    }

    printf("\nEnter the category to search for: ");
    getchar();
    fgets(searchCategory, 30, stdin);
    searchCategory[strcspn(searchCategory, "\n")] = 0;

    printf("\nCourses in the category '%s' sorted by rating:\n",
searchCategory);

    for(i = 0; i < n - 1; i++) {
        for(j = i + 1; j < n; j++) {
            if(courses[i].rating < courses[j].rating) {
                temp = courses[i];
                courses[i] = courses[j];
                courses[j] = temp;
            }
        }
    }

    int found = 0;
    for(i = 0; i < n; i++) {
        if(strcmp(courses[i].category, searchCategory) == 0) {
            printf("\nCourse Name: %s\n", courses[i].name);
            printf("Rating: %.2f\n", courses[i].rating);
            found = 1;
        }
    }

    if(!found) {
        printf("\nNo courses found in the category '%s'.\n", sear
chCategory);
    }

```

```
getch();  
}
```

22. A bank needs a program to calculate the total interest earned on fixed deposits for multiple customers. Each customer has a different principal amount, interest rate, and tenure. The system must take input for these values, compute the total interest for each customer, and display the results. Additionally, the program should handle edge cases such as zero tenure or invalid inputs. How would you design such a solution to streamline the interest calculation process?

```
#include<stdio.h>  
#include<conio.h>  
void main()  
{  
    int n, i;  
    float principal, rate, tenure, interest;  
    clrscr();  
  
    printf("Enter the number of  
customers: ");  
    scanf("%d", &n);  
  
    if (n <= 0) {  
        printf("Invalid number of customers.\n");  
        getch();  
        return;  
    }  
  
    for (i = 0; i < n; i++) {  
        printf("\nCustomer %d\n", i + 1);  
  
        printf("Enter principal amount: ");  
        scanf("%f", &principal);
```

```

printf("Enter interest rate (in percentage): ");
scanf("%f", &rate);

printf("Enter tenure (in years): ");
scanf("%f", &tenure);

if (principal <= 0 || rate <= 0 || tenure < 0) {
    printf("Invalid input. Please enter positive values for principal, rate, and tenure.\n");
    continue;
}

if (tenure == 0) {
    printf("Tenure is zero, no interest will be earned.\n");
    interest = 0;
} else {
    interest = (principal * rate * tenure) / 100;
}

printf("Total interest for Customer %d: %.2f\n", i + 1, interest);
}

getch();
}

```

23. A movie theatre runs a promotional campaign offering discounted tickets for students and senior citizens. The staff often struggles to verify eligibility based on age and student status. Write a program that accepts the customer's age and whether they are a student, then determines if they qualify for a discount. The solution should ensure efficiency during peak hours and handle invalid inputs gracefully.

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int age;
    char studentStatus;
    int discountEligible = 0;
    clrscr();

    printf("Enter customer's age: ");
    scanf("%d", &age);

    if(age <= 0) {
        printf("Invalid age entered. Please enter a valid age.\n");
        getch();
        return;
    }

    printf("Is the customer a student? (Y/N): ");
    getchar();
    scanf("%c", &studentStatus);

    if(studentStatus != 'Y' && studentStatus != 'N') {
        printf("Invalid input. Please enter 'Y' for Yes or 'N' for No.\n");
        getch();
        return;
    }

    if ((age <= 18) || (age >= 60) || (studentStatus == 'Y')) {
        discountEligible = 1;
    }
}

```

```

if (discountEligible) {
    printf("The customer qualifies for a discount.\n");
} else {
    printf("The customer does not qualify for a discount.\n");
}

getch();
}

```

24. A conservationist tracks the migration patterns of a bird species over a given number of days. Each day's migration distance is recorded, but the total distance and average distance for the period must be computed. Write a program to accept the number of days and the daily distances, then calculate and display the total, average, and the day with the longest migration. How would you make the program adaptable for varying data sizes?

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int days, i, longestDay = 0;
    float totalDistance = 0, averageDistance, maxDistance = 0, dailyDistance;

    clrscr();
    printf("Enter the number of days: ");
    scanf("%d", &days);

    if (days <= 0) {
        printf("Invalid number of days.\n");
        getch();
        return;
    }
}

```



```

}

for (i = 1; i <= days; i++) {
    printf("Enter the migration distance for day %d: ", i);
    scanf("%f", &dailyDistance);

    if (dailyDistance < 0) {
        printf("Invalid distance. Please enter a non-negative
value.\n");
        i--;
        continue;
    }

    totalDistance += dailyDistance;

    if (dailyDistance > maxDistance) {
        maxDistance = dailyDistance;
        longestDay = i;
    }
}

averageDistance = totalDistance / days;
printf("\nTotal migration distance: %.2f\n", totalDistance);
printf("Average migration distance: %.2f\n", averageDistance);
printf("The longest migration was on day %d with a distance of %.2f\n", longestDay, maxDistance);

getch();
}

```

25. An electricity board monitors daily power usage for a week and wants to determine patterns of consumption. The board requires a program to accept usage data for seven days, calculate the total and average consumption, and

identify the day with the highest usage. Additionally, the program should allow for future extension to accommodate monthly data. How would you design this system?

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i, highestDay = 0;
    float totalConsumption = 0, averageConsumption, maxConsumption = 0, dailyConsumption;
    clrscr();
    printf("Enter the power consumption (in units) for 7 days:\n");

    for (i = 1; i <= 7; i++) {
        printf("Day %d: ", i);
        scanf("%f", &dailyConsumption);

        if (dailyConsumption < 0) {
            printf("Invalid consumption value. Please enter a non-negative value.\n");
            i--;
            continue;
        }

        totalConsumption += dailyConsumption;

        if (dailyConsumption > maxConsumption) {
            maxConsumption = dailyConsumption;
            highestDay = i;
        }
    }
}
```

```

averageConsumption = totalConsumption / 7;
printf("\nTotal power consumption: %.2f units\n", totalConsumption);
printf("Average power consumption: %.2f units\n", averageConsumption);
printf("The highest power consumption was on day %d with %.2f units\n", highestDay, maxConsumption);

getch();
}

```

26. A tech store keeps track of laptops available for sale, each with a price and a user rating. Customers often search for laptops with high ratings that fit their budget. Write a program to accept the price and rating for multiple laptops, then suggest the best option based on the highest rating. If multiple laptops have the same rating, prioritize the one with the lowest price. How would you design such a system?

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int n, i, bestLaptopIndex = -1;
    float price, rating, minPrice = 0, maxRating = 0;

    clrscr();
    printf("Enter the number of laptops: ");
    scanf("%d", &n);

    if (n <= 0) {
        printf("Invalid number of laptops.\n");
        getch();
        return;
    }
}

```

```

}

float prices[n], ratings[n];

for (i = 0; i < n; i++) {
    printf("\nLaptop %d:\n", i + 1);
    printf("Enter price: ");
    scanf("%f", &prices[i]);
    printf("Enter rating: ");
    scanf("%f", &ratings[i]);

    if (prices[i] < 0 || ratings[i] < 0 || ratings[i] > 5) {
        printf("Invalid input. Price should be non-negative,
and rating should be between 0 and 5.\n");
        i--;
        continue;
    }

    if ((ratings[i] > maxRating) || (ratings[i] == maxRating
&& prices[i] < minPrice)) {
        maxRating = ratings[i];
        minPrice = prices[i];
        bestLaptopIndex = i;
    }
}

if (bestLaptopIndex != -1) {
    printf("\nBest laptop recommendation:\n");
    printf("Price: %.2f\n", prices[bestLaptopIndex]);
    printf("Rating: %.2f\n", ratings[bestLaptopIndex]);
} else {
    printf("\nNo laptops available.\n");
}

```

```
getch();  
}
```

27. A water dispenser in a park refills water bottles up to a certain capacity. The dispenser adds 500 ml per cycle and stops once the bottle is full. Write a program to accept the bottle's capacity and simulate the filling process, displaying the water level after each cycle. Ensure the program handles different bottle sizes dynamically and prevents overfilling.

```
#include<stdio.h>  
#include<conio.h>  
void main()  
{  
float bottleCapacity, currentWaterLevel = 0.0;  
int cycle = 0;  
clrscr();  
  
printf("Enter the capacity of the water bottle (in ml): ");  
scanf("%f", &bottleCapacity);  
  
if (bottleCapacity <= 0) {  
    printf("Invalid bottle capacity.\n");  
    getch();  
    return;  
}  
  
printf("\nStarting the filling process...\n");  
  
while (currentWaterLevel < bottleCapacity) {  
    cycle++;  
    currentWaterLevel += 500;  
  
    if (currentWaterLevel > bottleCapacity) {
```

```

        currentWaterLevel = bottleCapacity;
    }

    printf("Cycle %d: Water level = %.2f ml\n", cycle, currentWaterLevel);
}

printf("\nBottle is now full!\n");
getch();
}

```

28. A factory monitors the production of three different products daily. The management wants a summary that displays the total production for each product at the end of the week. Write a program to accept the daily production counts for all three products, calculate their weekly totals, and display a comprehensive summary. How would you design this to ensure clarity and accuracy?

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int product1[7], product2[7], product3[7];
    int total1 = 0, total2 = 0, total3 = 0;
    int i;
    clrscr();

    printf("Enter daily production counts for Product 1:\n");
    for (i = 0; i < 7; i++) {
        printf("Day %d: ", i + 1);
        scanf("%d", &product1[i]);
        total1 += product1[i];
    }
}

```

```

printf("\nEnter daily production counts for Product 2:\n");
for (i = 0; i < 7; i++) {
    printf("Day %d: ", i + 1);
    scanf("%d", &product2[i]);
    total2 += product2[i];
}

printf("\nEnter daily production counts for Product 3:\n");
for (i = 0; i < 7; i++) {
    printf("Day %d: ", i + 1);
    scanf("%d", &product3[i]);
    total3 += product3[i];
}

printf("\n Weekly Production Summary\n");
printf("Total production for Product 1: %d\n", total1);
printf("Total production for Product 2: %d\n", total2);
printf("Total production for Product 3: %d\n", total3);

getch();
}

```

29. A transportation company tracks the travel times of its vehicles over several routes. The company needs a program to accept travel times for each route, calculate the average time, and identify the fastest and slowest routes. The program must handle varying numbers of routes and ensure clear output for performance evaluation. How would you create such a system?

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int n, i, fastestIndex = 0, slowestIndex = 0;

```

```

float total = 0, average;
float travelTimes[100];
clrscr();

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

printf("\nEnter the travel times for each route (in hours):\n");
for (i = 0; i < n; i++) {
    printf("Route %d: ", i + 1);
    scanf("%f", &travelTimes[i]);
    total += travelTimes[i];

    if (travelTimes[i] < travelTimes[fastestIndex]) {
        fastestIndex = i;
    }
    if (travelTimes[i] > travelTimes[slowestIndex]) {
        slowestIndex = i;
    }
}

average = total / n;

printf("\nTravel Time Summary\n");
printf("Average travel time: %.2f hours\n", average);
printf("Fastest route: Route %d (%.2f hours)\n", fastestIndex + 1, travelTimes[fastestIndex]);
printf("Slowest route: Route %d (%.2f hours)\n", slowestIndex + 1, travelTimes[slowestIndex]);

getch();
}

```


30. An online grocery store tracks product categories like fruits, vegetables, and grains. Customers often search for products within a specific category, but the current system lacks efficient filtering options. Write a program using structures to store product details and allow users to search for items by category. Display all matching products in a user-friendly format. How would you design this system?

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
struct Product
{
char name[50];
char category[20];
float price;
}
void main()
{
int n, i, found = 0;
char searchCategory[20];
struct Product products[100];
clrscr();

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

for (i = 0; i < n; i++) {
printf("\nEnter details for product %d:\n", i + 1);
printf("Name: ");
fflush(stdin);
gets(products[i].name);
printf("Category (e.g., fruits, vegetables, grains): ");
gets(products[i].category);
printf("Price: ");
scanf("%f", &products[i].price);
```

```

}

printf("\nEnter the category to search for: ");
fflush(stdin);
gets(searchCategory);

printf("\nProducts in Category: %s\n", searchCategory);
for (i = 0; i < n; i++) {
    if (strcmp(products[i].category, searchCategory) == 0) {
        printf("Name: %s, Price: ₹%.2f\n", products[i].name,
products[i].price);
        found = 1;
    }
}

if (!found) {
    printf("No products found in the %s category.\n", searchC
ategory);
}

getch();
}

```

31. A car dealership tracks the prices of vehicles available in its inventory. Customers often want to compare prices to find the most affordable and the most premium vehicles. Write a program to accept the prices of all vehicles, identify the lowest and highest prices, and display them. Ensure the program is efficient and adaptable for large inventories. How would you design such a solution?

```

#include<stdio.h>
#include<conio.h>
void main()
{

```

```

int n, i;
float prices[100], minPrice, maxPrice;
clrscr();

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

if (n <= 0) {
    printf("Invalid number of vehicles. Exiting...");
    getch();
    return;
}

printf("Enter the prices of the vehicles:\n");
for (i = 0; i < n; i++) {
    printf("Vehicle %d: ₹", i + 1);
    scanf("%f", &prices[i]);

    if (prices[i] <= 0) {
        printf("Invalid price. Exiting...");
        getch();
        return;
    }
}

minPrice = maxPrice = prices[0];

for (i = 1; i < n; i++) {
    if (prices[i] < minPrice) {
        minPrice = prices[i];
    }
    if (prices[i] > maxPrice) {
        maxPrice = prices[i];
    }
}

```

```

printf("\nPrice Summary \n");
printf("Lowest Price: ₹%.2f\n", minPrice);
printf("Highest Price: ₹%.2f\n", maxPrice);

getch();
}

```

32. A physics lab is analyzing data from two experiments, each represented as a matrix of measurements. To determine the difference between the two experiments, the researchers need to subtract one matrix from the other. The lab requires a program that first ensures the dimensions of the matrices are compatible for subtraction, then calculates the resulting matrix. Write a C program to handle this task and display the output matrix in a clean, readable format.

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int rows, cols, i, j;
    int matrix1[10][10], matrix2[10][10], result[10][10];

    clrscr();

    printf("Enter the number of rows: ");
    scanf("%d", &rows);
    printf("Enter the number of columns: ");
    scanf("%d", &cols);

    if (rows <= 0 || cols <= 0 || rows > 10 || cols > 10) {
        printf("Invalid matrix dimensions. Exiting...");
    }
}

```

```

        getch();
        return;
    }

    printf("\nEnter elements of the first matrix:\n");
    for (i = 0; i < rows; i++) {
        for (j = 0; j < cols; j++) {
            printf("Element [%d][%d]: ", i + 1, j + 1);
            scanf("%d", &matrix1[i][j]);
        }
    }

    printf("\nEnter elements of the second matrix:\n");
    for (i = 0; i < rows; i++) {
        for (j = 0; j < cols; j++) {
            printf("Element [%d][%d]: ", i + 1, j + 1);
            scanf("%d", &matrix2[i][j]);
        }
    }

    for (i = 0; i < rows; i++) {
        for (j = 0; j < cols; j++) {
            result[i][j] = matrix1[i][j] - matrix2[i][j];
        }
    }

    printf("\nResultant Matrix (Matrix1 - Matrix2):\n");
    for (i = 0; i < rows; i++) {
        for (j = 0; j < cols; j++) {
            printf("%4d", result[i][j]);
        }
        printf("\n");
    }
}

```

```
getch();  
}
```