Problem 1: Daily Step Tracker

Time limit: 1 second

Memory limit: 64 MB

Question:

A fitness tracker logs the number of steps a user takes each day for nnn days. Write a program to count how many days the user achieved the goal of walking more than 10,000 steps.

Input:

- The first line contains an integer nnn (number of days).
- The second line contains nnn integers representing the steps taken each day.

Output:

- A single integer representing the number of days the goal was met.

Code:

c

Copy code

```c
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of days
    int steps, count = 0; // Variable for steps and counter

    for (int i = 0; i < n; i++) {
        scanf("%d", &steps); // Read steps for each day
        if (steps > 10000) {
            count++; // Increment count if steps are more than 10,000
        }
    }
    printf("%d\n", count); // Print number of days the goal was met
    return 0;
}
```

Explanation:

- The program starts by reading the number of days, *nn*n.
- It initializes a counter `count` to track how many days exceed the step goal of 10,000.
- A loop reads the steps taken each day and checks if they are greater than 10,000. If so, it increments the counter.
- Finally, it outputs the count of days that met the goal.

Problem 2: Monthly Expense Manager

Time limit: 1 second

Memory limit: 64 MB

Question:

Write a program that calculates the total monthly expenses, and if the total exceeds a given budget, print "Over Budget". Otherwise, print "Within Budget".

Input:

- The first line contains an integer *nn*n (number of expense categories).
- The second line contains *nn*n integers representing expenses.
- The third line contains an integer representing the monthly budget.

Output:

- "Over Budget" if the total expense exceeds the budget, otherwise "Within Budget".

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of expense categories
    int expense, total = 0; // Variable for each expense and total
expenses

    for (int i = 0; i < n; i++) {
        scanf("%d", &expense); // Read each expense
        total += expense; // Add to total expenses
    }

    int budget;
    scanf("%d", &budget); // Read monthly budget

    if (total > budget) {
        printf("Over Budget\n"); // Print if over budget
    } else {
        printf("Within Budget\n"); // Print if within budget
    }

    return 0;
}
```

Explanation:

- The program begins by reading the number of expense categories, nnn.
- It initializes a variable `total` to accumulate the expenses.
- A loop collects each expense and adds it to the total.
- After reading all expenses, it checks if the total exceeds the budget and prints the appropriate message.

Problem 3: Grading System

Time limit: 1 second

Memory limit: 64 MB

Question:

A teacher wants to calculate the final grade for students. If a student scores 50 or more, they pass; otherwise, they fail. Write a program that counts how many students passed.

Input:

- The first line contains an integer nnn (number of students).
- The second line contains nnn integers representing the grades.

Output:

- A single integer representing the number of students who passed.

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of students
    int grade, passCount = 0; // Variable for grades and pass counter

    for (int i = 0; i < n; i++) {
        scanf("%d", &grade); // Read each grade
        if (grade >= 50) {
            passCount++; // Increment counter if grade is 50 or more
        }
    }
    printf("%d\n", passCount); // Print number of students who passed
    return 0;
}
```

Explanation:

- The program starts by reading the number of students, nnn.
- It initializes a counter `passCount` to track passing students.
- A loop processes each student's grade, checking if it is 50 or higher. If it is, the counter increments.

- Finally, it outputs the count of students who passed.

Problem 4: Bank Loan Interest

Time limit: 1 second

Memory limit: 64 MB

Question:

A bank gives a different interest rate based on the loan amount. If the loan is greater than 50,000 units, the interest rate is 7%. If the loan is less than or equal to 50,000 units, the rate is 5%. Calculate the total interest for each loan in a list.

Input:

- The first line contains an integer nnn (number of loans).
- The second line contains nnn integers representing the loan amounts.

Output:

- The total interest for each loan.

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of loans
    int loan;

    for (int i = 0; i < n; i++) {
        scanf("%d", &loan); // Read loan amount
        float interest = (loan > 50000) ? (loan * 0.07) : (loan *
0.05); // Calculate interest
        printf("%.2f\n", interest); // Print the calculated interest
    }
    return 0;
}
```

Explanation:

- The program begins by reading the number of loans, nnn.
- A loop processes each loan amount, determining the interest based on the specified rates.
- The calculated interest is printed with two decimal precision for clarity.

Problem 5: Temperature Warnings

Time limit: 1 second

Memory limit: 64 MB

Question:

In a weather monitoring system, temperatures are recorded hourly for a week. If the temperature goes above 35°C, a heat warning is issued. Write a program that counts how many hours the heat warning was issued.

Input:

- The first line contains an integer nnn (number of hours).
- The second line contains nnn integers representing the hourly temperatures.

Output:

- A single integer representing the number of heat warnings issued.

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of hours
    int temp, warningCount = 0; // Variable for temperature and
warning counter

    for (int i = 0; i < n; i++) {
        scanf("%d", &temp); // Read temperature
        if (temp > 35) {
            warningCount++; // Increment count if temperature exceeds
35
        }
    }
    printf("%d\n", warningCount); // Print number of heat warnings
issued
    return 0;
}
```

Explanation:

- The program starts by reading the number of hours, nnn.
- It initializes a counter `warningCount` to track the number of heat warnings.
- A loop reads each temperature and increments the counter if it exceeds 35°C.
- Finally, it outputs the total count of heat warnings issued.

Problem 6: Sales Bonus Calculation

Time limit: 1 second

Memory limit: 64 MB

Question:

A company gives its employees a bonus if their sales for the month exceed a certain threshold. For each employee, if their sales are greater than 1,000 units, they receive a 10% bonus; otherwise, they receive no bonus. Write a program to calculate the bonus for each employee.

Input:

- The first line contains an integer nnn (number of employees).
- The second line contains nnn integers representing the sales of each employee.

Output:

- The bonus for each employee.

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of employees
    int sales;

    for (int i = 0; i < n; i++) {
        scanf("%d", &sales); // Read sales for each employee
        float bonus = (sales > 1000) ? (sales * 0.10) : 0; //
Calculate bonus
        printf("%.2f\n", bonus); // Print the bonus
    }
    return 0;
}
```

Explanation:

- The program starts by reading the number of employees, nnn.
- A loop processes each employee's sales, calculating a 10% bonus if sales exceed 1,000.
- The calculated bonus is printed with two decimal precision.

Problem 7: Electricity Bill Calculation

Time limit: 1 second

Memory limit: 64 MB

Question:

Write a program to calculate the electricity bill for each household. The rate is 5 units per kWh if usage is below 100 kWh, and 10 units per kWh if usage is 100 kWh or more.

Input:

- The first line contains an integer nnn (number of households).
- The second line contains nnn integers representing the electricity usage of each household.

Output:

- The bill for each household.

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of households
    int usage;

    for (int i = 0; i < n; i++) {
        scanf("%d", &usage); // Read electricity usage
        float bill = (usage < 100) ? (usage * 5) : (usage * 10); //
Calculate bill
        printf("%.2f\n", bill); // Print the bill
    }
    return 0;
}
```

Explanation:

- The program starts by reading the number of households, nnn.

- A loop processes each household's electricity usage, calculating the bill based on the specified rates.
- The calculated bill is printed with two decimal precision.

Problem 8: Overtime Pay Calculation

Time limit: 1 second

Memory limit: 64 MB

Question:

Write a program to calculate the overtime pay for each worker. For every hour worked above 40 hours, they are paid 1.5 times their regular hourly wage.

Input:

- The first line contains an integer nnn (number of workers).
- The second line contains nnn integers representing hours worked by each worker.
- The third line contains a single integer representing the regular hourly wage.

Output:

- The overtime pay for each worker.

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of workers
    int hours;
    float wage;
    scanf("%f", &wage); // Read regular hourly wage

    for (int i = 0; i < n; i++) {
        scanf("%d", &hours); // Read hours worked
        float overtimePay = (hours > 40) ? ((hours - 40) * wage *
1.5) : 0; // Calculate overtime pay
        printf("%.2f\n", overtimePay); // Print the overtime pay
    }
    return 0;
}
```

Explanation:

- The program starts by reading the number of workers, nnn, and the regular hourly wage.
- A loop processes each worker's hours worked, calculating overtime pay if hours exceed 40.
- The calculated overtime pay is printed with two decimal precision.

Problem 9: Fuel Efficiency

Time limit: 1 second

Memory limit: 64 MB

Question:

A car's fuel efficiency is measured for nnn trips. If the fuel efficiency for any trip is below 15 km/l, it is considered inefficient. Write a program that counts how many trips were inefficient.

Input:

- The first line contains an integer nnn (number of trips).
- The second line contains nnn floating-point numbers representing fuel efficiency for each trip.

Output:

- A single integer representing the number of inefficient trips.

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of trips
    float efficiency;
    int inefficientCount = 0; // Variable for counting inefficient
trips

    for (int i = 0; i < n; i++) {
        scanf("%f", &efficiency); // Read fuel efficiency
        if (efficiency < 15) {
            inefficientCount++; // Increment count if efficiency is
below 15 km/l
        }
    }
    printf("%d\n", inefficientCount); // Print number of inefficient
trips
    return 0;
}
```

Explanation:

- The program starts by reading the number of trips, *nn*n.
- It initializes a counter `inefficientCount` to track trips with low fuel efficiency.
- A loop reads the fuel efficiency for each trip and increments the counter if it's below 15 km/l.
- Finally, it outputs the total count of inefficient trips.

Problem 10: Weight Tracker

Time limit: 1 second

Memory limit: 64 MB

Question:

Write a program that tracks a person's weight over nnn weeks. If their weight increases from the previous week, print "Gain"; if it decreases, print "Loss"; if it stays the same, print "No Change".

Input:

- The first line contains an integer nnn (number of weeks).
- The second line contains nnn integers representing the weekly weight measurements.

Output:

- For each week after the first, print "Gain", "Loss", or "No Change".

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of weeks
    int weights[100]; // Array to store weights for n weeks

    for (int i = 0; i < n; i++) {
        scanf("%d", &weights[i]); // Read weekly weight measurements
    }

    // Compare each week with the previous week
    for (int i = 1; i < n; i++) {
        if (weights[i] > weights[i - 1]) {
            printf("Gain\n"); // Print "Gain" if weight increased
        } else if (weights[i] < weights[i - 1]) {
            printf("Loss\n"); // Print "Loss" if weight decreased
        } else {
            printf("No Change\n"); // Print "No Change" if weight
stayed the same
        }
    }
    return 0;
}
```

Explanation:

- The program begins by reading the number of weeks, nnn, and stores each week's weight in an array.
- A loop compares each week's weight with the previous week's weight starting from the second week.
- Based on the comparison, it prints "Gain", "Loss", or "No Change" for each week.

Problem 11: Store Discount Calculator

Time limit: 1 second

Memory limit: 64 MB

Question:

A store gives a 10% discount if the total purchase is above 1,000 units. Calculate the total bill for nnn customers, applying the discount if applicable.

Input:

- The first line contains an integer nnn (number of customers).
- The second line contains nnn integers representing the total purchase of each customer.

Output:

- The total bill for each customer after applying the discount (if applicable).

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of customers
    int purchase;

    for (int i = 0; i < n; i++) {
        scanf("%d", &purchase); // Read total purchase
        float bill = (purchase > 1000) ? (purchase * 0.90) : purchase;
// Calculate bill after discount
        printf("%.2f\n", bill); // Print the total bill
    }
    return 0;
}
```

Explanation:

- The program starts by reading the number of customers, nnn.
- For each customer, it reads the total purchase and checks if it exceeds 1,000 units.
- If it does, a 10% discount is applied; otherwise, the full purchase amount is used.
- Finally, the program prints the adjusted total bill.

Problem 12: Tax Calculation

Time limit: 1 second

Memory limit: 64 MB

Question:

A government tax rate is applied as follows: if a person's income is greater than 50,000 units, they are taxed 20%; otherwise, they are taxed 10%. Calculate the tax for nnn people.

Input:

- The first line contains an integer nnn (number of people).
- The second line contains nnn integers representing each person's income.

Output:

- The tax for each person.

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of people
    int income;

    for (int i = 0; i < n; i++) {
        scanf("%d", &income); // Read income
        float tax = (income > 50000) ? (income * 0.20) : (income *
0.10); // Calculate tax
        printf("%.2f\n", tax); // Print the tax
    }
    return 0;
}
```

Explanation:

- The program begins by reading the number of people, nnn.
- It processes each income, applying the corresponding tax rate based on the specified thresholds.
- Finally, it prints the calculated tax for each person.

Problem 13: Blood Pressure Monitor

Time limit: 1 second

Memory limit: 64 MB

Question:

A health clinic measures a patient's blood pressure every day for nnn days. If the systolic pressure is above 120 or diastolic is above 80, print "High". If both values are below or equal to the limits, print "Normal".

Input:

- The first line contains an integer nnn (number of days).
- Each of the next nnn lines contains two integers representing systolic and diastolic pressure for that day.

Output:

- For each day, print "High" or "Normal".

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of days
    int systolic, diastolic;

    for (int i = 0; i < n; i++) {
        scanf("%d %d", &systolic, &diastolic); // Read systolic and
diastolic pressure
        if (systolic > 120 || diastolic > 80) {
            printf("High\n"); // Print "High" if pressure is above
limits
        } else {
            printf("Normal\n"); // Print "Normal" if within limits
        }
    }
    return 0;
}
```

Explanation:

- The program starts by reading the number of days, nnn.
- For each day, it reads the systolic and diastolic pressures, checking against the defined thresholds.

- It prints "High" for elevated readings and "Normal" for normal readings.

Problem 14: Bus Fare Calculation

Time limit: 1 second

Memory limit: 64 MB

Question:

A bus charges different fares based on the passenger's age. If the age is less than 12, the fare is 5 units; if the age is 60 or above, the fare is 7 units; otherwise, the fare is 10 units. Write a program to calculate the fare for each passenger.

Input:

- The first line contains an integer nnn (number of passengers).
- The second line contains nnn integers representing the age of each passenger.

Output:

- The fare for each passenger.

Code:

c

Copy code

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of passengers
    int age;

    for (int i = 0; i < n; i++) {
        scanf("%d", &age); // Read age of passenger
        int fare = (age < 12) ? 5 : (age >= 60 ? 7 : 10); // Determine
fare based on age
        printf("%d\n", fare); // Print the fare
    }
    return 0;
}
```

Explanation:

- The program begins by reading the number of passengers, nnn.
- It checks each passenger's age to determine the fare according to the specified rules.
- The fare is then printed for each passenger.

Problem 15: Discount Eligibility

Time limit: 1 second

Memory limit: 64 MB

Question:

Write a program to check if a customer is eligible for a discount based on their purchase history. If the total number of purchases made in a month is greater than 5, the customer gets a 20% discount on their next purchase; otherwise, no discount is applied.

Input:

- The first line contains an integer nnn (number of customers).
- The second line contains nnn integers representing the number of purchases each customer made.

Output:

- For each customer, print the discount amount (20% or 0%).

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of customers
    int purchases;

    for (int i = 0; i < n; i++) {
        scanf("%d", &purchases); // Read number of purchases
        float discount = (purchases > 5) ? 0.20 : 0.00; // Determine
discount eligibility
        printf("%.2f\n", discount); // Print the discount amount
    }
    return 0;
}
```

Explanation:

- The program starts by reading the number of customers, nnn.
- It checks the number of purchases for each customer and determines if they qualify for a discount.
- The discount amount is printed accordingly.

Problem 16: Customer Feedback Rating

Time limit: 1 second

Memory limit: 64 MB

Question:

A company collects customer feedback ratings. If the rating is below 3, print "Needs Improvement"; if it's between 3 and 4, print "Satisfactory"; if it's above 4, print "Excellent".

Input:

- The first line contains an integer nnn (number of feedback ratings).
- The second line contains nnn integers representing the ratings.

Output:

- For each rating, print the corresponding message.

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of ratings
    int rating;

    for (int i = 0; i < n; i++) {
        scanf("%d", &rating); // Read feedback rating
        if (rating < 3) {
            printf("Needs Improvement\n"); // Print for low ratings
        } else if (rating >= 3 && rating < 5) {
            printf("Satisfactory\n"); // Print for satisfactory
ratings
        } else {
            printf("Excellent\n"); // Print for high ratings
        }
    }
    return 0;
}
```

Explanation:

- The program starts by reading the number of ratings, nnn.
- For each rating, it checks its value and prints the corresponding message based on the specified ranges.

Problem 17: Monthly Expense Analysis

Time limit: 1 second

Memory limit: 64 MB

Question:

Write a program to categorize monthly expenses. If the expense exceeds 1,000 units, print "High Expense"; if it's between 500 and 1,000 units, print "Moderate Expense"; otherwise, print "Low Expense".

Input:

- The first line contains an integer nnn (number of months).
- The second line contains nnn integers representing monthly expenses.

Output:

- For each month, print the corresponding message.

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of months
    int expense;

    for (int i = 0; i < n; i++) {
        scanf("%d", &expense); // Read monthly expense
        if (expense > 1000) {
            printf("High Expense\n"); // Print for high expenses
        } else if (expense >= 500) {
            printf("Moderate Expense\n"); // Print for moderate
expenses
        } else {
            printf("Low Expense\n"); // Print for low expenses
        }
    }
    return 0;
}
```

Explanation:

- The program begins by reading the number of months, nnn.
- It evaluates each month’s expense and categorizes it into high, moderate, or low.
- Finally, it prints the corresponding message for each month.

Problem 18: Weekly Workout Tracker

Time limit: 1 second

Memory limit: 64 MB

Question:

Write a program to check weekly workout consistency. If the user exercises more than 5 times a week, print "Very Active"; if it’s between 3 and 5 times, print "Moderately Active"; otherwise, print "Inactive".

Input:

- The first line contains an integer nnn (number of weeks).
- The second line contains nnn integers representing workout frequency.

Output:

- For each week, print the corresponding message.

Code:

c
Copy code

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of weeks
    int workouts;

    for (int i = 0; i < n; i++) {
        scanf("%d", &workouts); // Read workout frequency
        if (workouts > 5) {
            printf("Very Active\n"); // Print for very active
        } else if (workouts >= 3) {
            printf("Moderately Active\n"); // Print for moderately
active
        } else {
            printf("Inactive\n"); // Print for inactive
        }
    }
    return 0;
}
```

Explanation:

- The program starts by reading the number of weeks, nnn.
- For each week, it checks the number of workouts and categorizes the activity level.
- The corresponding message is printed for each week.

Problem 19: Household Energy Consumption

Time limit: 1 second

Memory limit: 64 MB

Question:

Write a program to evaluate household energy consumption. If consumption exceeds 300 kWh, print "High Consumption"; if it's between 150 kWh and 300 kWh, print "Average Consumption"; otherwise, print "Low Consumption".

Input:

- The first line contains an integer nnn (number of households).
- The second line contains nnn integers representing energy consumption.

Output:

- For each household, print the corresponding message.

Code:

c

Copy code

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of households
    int consumption;

    for (int i = 0; i < n; i++) {
        scanf("%d", &consumption); // Read energy consumption
        if (consumption > 300) {
            printf("High Consumption\n"); // Print for high
consumption
        } else if (consumption >= 150) {
            printf("Average Consumption\n"); // Print for average
consumption
        } else {
            printf("Low Consumption\n"); // Print for low consumption
        }
    }
    return 0;
}
```

Explanation:

- The program starts by reading the number of households, *nn*n.
- It checks each household's energy consumption and categorizes it accordingly.
- Finally, the program prints the relevant message for each household.

Problem 20: Student Score Evaluation

Time limit: 1 second

Memory limit: 64 MB

Question:

Write a program to evaluate student scores. If the score is below 40, print "Fail"; if it’s between 40 and 60, print "Pass"; if it’s above 60, print "Excellent".

Input:

- The first line contains an integer nnn (number of students).
- The second line contains nnn integers representing scores.

Output:

- For each student, print the corresponding message.

Code:

c

Copy code

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of students
    int score;

    for (int i = 0; i < n; i++) {
        scanf("%d", &score); // Read student score
        if (score < 40) {
            printf("Fail\n"); // Print for fail
        } else if (score <= 60) {
            printf("Pass\n"); // Print for pass
        } else {
            printf("Excellent\n"); // Print for excellent
        }
    }
    return 0;
}
```

Explanation:

- The program begins by reading the number of students, nnn.

- It evaluates each student's score and determines the appropriate message based on the specified ranges.
- Finally, the message is printed for each student.

Problem 21: Food Delivery Time Evaluation

Time limit: 1 second

Memory limit: 64 MB

Question:

Assess food delivery times. If the delivery time exceeds 30 minutes, print "Late Delivery"; if it's between 15 and 30 minutes, print "On Time"; otherwise, print "Early Delivery".

Input:

- The first line contains an integer nnn (number of deliveries).
- The second line contains nnn integers representing delivery times in minutes.

Output:

- For each delivery, print the corresponding message.

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of deliveries
    int delivery_time;

    for (int i = 0; i < n; i++) {
        scanf("%d", &delivery_time); // Read delivery time
        if (delivery_time > 30) {
            printf("Late Delivery\n"); // Print for late delivery
        } else if (delivery_time >= 15) {
            printf("On Time\n"); // Print for on time
        } else {
            printf("Early Delivery\n"); // Print for early delivery
        }
    }
    return 0;
}
```

Explanation:

- The program reads the number of deliveries, *nn*n.
- It checks each delivery time and categorizes it accordingly, printing the relevant message.

Problem 22: Grocery Store Checkout

Time limit: 1 second

Memory limit: 64 MB

Question:

Evaluate the total cost at a grocery store checkout. If the total cost exceeds 100 units, print "Apply Discount"; if it's between 50 and 100 units, print "Regular Price"; otherwise, print "Low Price".

Input:

- The first line contains an integer *nn*n (number of customers).
- The second line contains *nn*n integers representing total costs.

Output:

- For each customer, print the corresponding message.

Code:

```
c
Copy code
```

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of customers
    int total_cost;

    for (int i = 0; i < n; i++) {
        scanf("%d", &total_cost); // Read total cost
        if (total_cost > 100) {
            printf("Apply Discount\n"); // Print for applying discount
        } else if (total_cost >= 50) {
            printf("Regular Price\n"); // Print for regular price
        } else {
            printf("Low Price\n"); // Print for low price
        }
    }
    return 0;
}
```

Explanation:

- The program starts by reading the number of customers, 𝑛𝑛n.
- It checks each customer's total cost and prints the corresponding message based on the cost thresholds.

Problem 23: Daily Caloric Intake

Time limit: 1 second

Memory limit: 64 MB

Question:

Evaluate daily caloric intake. If intake exceeds 2500 calories, print "Over Limit"; if it’s between 2000 and 2500 calories, print "On Target"; otherwise, print "Under Limit".

Input:

- The first line contains an integer nnn (number of days).
- The second line contains nnn integers representing daily caloric intake.

Output:

- For each day, print the corresponding message.

Code:

c

Copy code

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of days
    int caloric_intake;

    for (int i = 0; i < n; i++) {
        scanf("%d", &caloric_intake); // Read caloric intake
        if (caloric_intake > 2500) {
            printf("Over Limit\n"); // Print for over limit
        } else if (caloric_intake >= 2000) {
            printf("On Target\n"); // Print for on target
        } else {
            printf("Under Limit\n"); // Print for under limit
        }
    }
    return 0;
}
```

Explanation:

- The program reads the number of days, nnn.
- It evaluates each day's caloric intake and prints the corresponding message based on the defined limits.

Problem 24: Employee Performance Review

Time limit: 1 second

Memory limit: 64 MB

Question:

Evaluate employee performance ratings. If the rating is below 2, print "Needs Improvement"; if it’s between 2 and 4, print "Satisfactory"; if it’s above 4, print "Outstanding".

Input:

- The first line contains an integer nnn (number of employees).
- The second line contains nnn integers representing ratings.

Output:

- For each employee, print the corresponding message.

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of employees
    int rating;

    for (int i = 0; i < n; i++) {
        scanf("%d", &rating); // Read employee rating
        if (rating < 2) {
            printf("Needs Improvement\n"); // Print for needs
improvement
        } else if (rating <= 4) {
            printf("Satisfactory\n"); // Print for satisfactory
        } else {
            printf("Outstanding\n"); // Print for outstanding
        }
    }
    return 0;
}
```

Explanation:

- The program starts by reading the number of employees, nnn.
- It checks each employee's rating and prints the relevant performance review message.

Problem 25: Daily Water Intake Tracker

Time limit: 1 second

Memory limit: 64 MB

Question:

Track daily water intake. If intake is below 2 liters, print "Drink More Water"; if it’s between 2 to 3 liters, print "Good Job"; if it exceeds 3 liters, print "Excessive Intake".

Input:

- The first line contains an integer nnn (number of days).
- The second line contains nnn floating-point numbers representing daily water intake.

Output:

- For each day, print the corresponding message.

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of days
    float water_intake;

    for (int i = 0; i < n; i++) {
        scanf("%f", &water_intake); // Read daily water intake
        if (water_intake < 2.0) {
            printf("Drink More Water\n"); // Print for low intake
        } else if (water_intake <= 3.0) {
            printf("Good Job\n"); // Print for good job
        } else {
            printf("Excessive Intake\n"); // Print for excessive
intake
        }
    }
    return 0;
}
```

Explanation:

- The program begins by reading the number of days, *nn*n.
- It evaluates each day's water intake and categorizes it accordingly, printing the relevant message.

Problem 26: Pet Weight Check

Time limit: 1 second

Memory limit: 64 MB

Question:

Evaluate pet weight. If the pet’s weight is above the recommended limit, print "Overweight"; if it’s within the recommended range, print "Healthy Weight"; otherwise, print "Underweight".

Input:

- The first line contains an integer nnn (number of pets).
- The second line contains nnn floating-point numbers representing each pet's weight.

Output:

- For each pet, print the corresponding message.

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of pets
    float weight;

    for (int i = 0; i < n; i++) {
        scanf("%f", &weight); // Read pet's weight
        if (weight > 10.0) { // Assume 10.0 as the recommended limit
for this example
            printf("Overweight\n"); // Print for overweight
        } else if (weight >= 5.0) { // Assume 5.0 as the lower limit
for healthy weight
            printf("Healthy Weight\n"); // Print for healthy weight
        } else {
            printf("Underweight\n"); // Print for underweight
        }
    }
    return 0;
}
```

Explanation:

- The program reads the number of pets, nnn.
- It checks each pet's weight and categorizes it into overweight, healthy weight, or underweight, printing the relevant message.

Problem 27: Holiday Expense Analysis

Time limit: 1 second

Memory limit: 64 MB

Question:

Evaluate holiday expenses. If expenses exceed 500 units, print "Too Much Spending"; if it’s between 200 and 500 units, print "Moderate Spending"; otherwise, print "Low Spending".

Input:

- The first line contains an integer nnn (number of holidays).
- The second line contains nnn integers representing holiday expenses.

Output:

- For each holiday, print the corresponding message.

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of holidays
    int expenses;

    for (int i = 0; i < n; i++) {
        scanf("%d", &expenses); // Read holiday expenses
        if (expenses > 500) {
            printf("Too Much Spending\n"); // Print for too much
spending
        } else if (expenses >= 200) {
            printf("Moderate Spending\n"); // Print for moderate
spending
```

```
        } else {
            printf("Low Spending\n"); // Print for low spending
        }
    }
    return 0;
}
```

Explanation:

- The program starts by reading the number of holidays, nnn.
- It evaluates each holiday's expenses and prints the appropriate message based on the defined thresholds.

Problem 28: Food Inventory Check

Time limit: 1 second

Memory limit: 64 MB

Question:

Check food inventory. If the quantity of any item is below 5 units, print "Reorder"; if it’s between 5 and 10 units, print "Sufficient"; if it’s above 10 units, print "Stocked Up".

Input:

- The first line contains an integer nnn (number of food items).
- The second line contains nnn integers representing the quantities.

Output:

- For each item, print the corresponding message.

Code:

```
c
Copy code
#include <stdio.h>
```

```
int main() {
    int n;
    scanf("%d", &n); // Read number of food items
    int quantity;

    for (int i = 0; i < n; i++) {
        scanf("%d", &quantity); // Read quantity of food item
        if (quantity < 5) {
            printf("Reorder\n"); // Print for reorder
        } else if (quantity <= 10) {
            printf("Sufficient\n"); // Print for sufficient
        } else {
            printf("Stocked Up\n"); // Print for stocked up
        }
    }
    return 0;
}
```

Explanation:

- The program begins by reading the number of food items, nnn.
- It checks the quantity of each item and categorizes it, printing the relevant message.

Problem 29: Vehicle Speed Monitoring

Time limit: 1 second

Memory limit: 64 MB

Question:

Monitor vehicle speeds. If speed exceeds 80 km/h, print "Speeding"; if it's between 50 and 80 km/h, print "Normal Speed"; otherwise, print "Under Speed".

Input:

- The first line contains an integer nnn (number of vehicles).

- The second line contains nnn integers representing speeds.

Output:

- For each vehicle, print the corresponding message.

Code:

c
Copy code

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of vehicles
    int speed;

    for (int i = 0; i < n; i++) {
        scanf("%d", &speed); // Read vehicle speed
        if (speed > 80) {
            printf("Speeding\n"); // Print for speeding
        } else if (speed >= 50) {
            printf("Normal Speed\n"); // Print for normal speed
        } else {
            printf("Under Speed\n"); // Print for under speed
        }
    }
    return 0;
}
```

Explanation:

- The program starts by reading the number of vehicles, nnn.
- It evaluates each vehicle's speed and prints the appropriate message based on the speed thresholds.

Problem 30: Health Risk Assessment

Time limit: 1 second

Memory limit: 64 MB

Question:

Assess health risks based on BMI. If BMI is below 18.5, print "Underweight"; if it's between 18.5 and 24.9, print "Normal"; if it's between 25 and 29.9, print "Overweight"; otherwise, print "Obesity".

Input:

- The first line contains an integer nnn (number of individuals).
- The second line contains nnn floating-point numbers representing BMI values.

Output:

- For each individual, print the corresponding message.

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of individuals
    float bmi;

    for (int i = 0; i < n; i++) {
        scanf("%f", &bmi); // Read BMI value
        if (bmi < 18.5) {
            printf("Underweight\n"); // Print for underweight
        } else if (bmi < 25.0) {
            printf("Normal\n"); // Print for normal
        } else if (bmi < 30.0) {
            printf("Overweight\n"); // Print for overweight
        } else {
            printf("Obesity\n"); // Print for obesity
        }
    }
    return 0;
}
```

Explanation:

- The program begins by reading the number of individuals, 𝑛𝑛n.
- It checks each individual's BMI and categorizes it into underweight, normal, overweight, or obesity, printing the relevant message.

Problem 31: Travel Distance Evaluation

Time limit: 1 second

Memory limit: 64 MB

Question:

Evaluate travel distances. If the distance exceeds 1000 km, print "Long Journey"; if it’s between 500 km and 1000 km, print "Moderate Journey"; otherwise, print "Short Journey".

Input:

- The first line contains an integer 𝑛𝑛n (number of trips).
- The second line contains 𝑛𝑛n integers representing travel distances.

Output:

- For each trip, print the corresponding message.

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of trips
    int distance;

    for (int i = 0; i < n; i++) {
        scanf("%d", &distance); // Read travel distance
        if (distance > 1000) {
            printf("Long Journey\n"); // Print for long journey
        } else if (distance >= 500) {
            printf("Moderate Journey\n"); // Print for moderate
journey
        } else {
            printf("Short Journey\n"); // Print for short journey
        }
    }
    return 0;
}
```

Explanation:

- The program reads the number of trips, nnn.
- It evaluates each trip's distance and prints the appropriate message.

Problem 32: Monthly Subscription Feedback

Time limit: 1 second

Memory limit: 64 MB

Question:

Evaluate monthly subscription feedback ratings. If the rating is below 3, print "Consider Canceling"; if it’s between 3 and 4, print "Satisfactory"; if it’s above 4, print "Highly Satisfied".

Input:

- The first line contains an integer nnn (number of subscriptions).
- The second line contains nnn integers representing ratings.

Output:

- For each subscription, print the corresponding message.

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of subscriptions
    int rating;

    for (int i = 0; i < n; i++) {
        scanf("%d", &rating); // Read subscription rating
        if (rating < 3) {
            printf("Consider Canceling\n"); // Print for canceling
consideration
        } else if (rating <= 4) {
            printf("Satisfactory\n"); // Print for satisfactory
        } else {
            printf("Highly Satisfied\n"); // Print for highly
satisfied
        }
    }
    return 0;
}
```

Explanation:

- The program reads the number of subscriptions, nnn.
- It evaluates each subscription's rating and prints the corresponding message.

Problem 33: Book Reading Tracker

Time limit: 1 second

Memory limit: 64 MB

Question:

Evaluate the number of books read in a month. If a person reads more than 10 books, print "Bookworm"; if it's between 5 and 10, print "Avid Reader"; otherwise, print "Casual Reader".

Input:

- The first line contains an integer nnn (number of individuals).
- The second line contains nnn integers representing the number of books read.

Output:

- For each individual, print the corresponding message.

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of individuals
    int books_read;

    for (int i = 0; i < n; i++) {
        scanf("%d", &books_read); // Read number of books read
        if (books_read > 10) {
            printf("Bookworm\n"); // Print for bookworm
        } else if (books_read >= 5) {
            printf("Avid Reader\n"); // Print for avid reader
        } else {
            printf("Casual Reader\n"); // Print for casual reader
        }
    }
    return 0;
}
```

Explanation:

- The program reads the number of individuals, nnn.
- It evaluates each individual's book reading count and prints the appropriate message.

Problem 34: Movie Rating Evaluation

Time limit: 1 second

Memory limit: 64 MB

Question:

Evaluate movie ratings. If the rating is below 3, print "Poor Movie"; if it’s between 3 and 4, print "Average Movie"; if it’s above 4, print "Must Watch".

Input:

- The first line contains an integer nnn (number of movies).
- The second line contains nnn floating-point numbers representing ratings.

Output:

- For each movie, print the corresponding message.

Code:

c

Copy code

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of movies
    float rating;

    for (int i = 0; i < n; i++) {
        scanf("%f", &rating); // Read movie rating
        if (rating < 3.0) {
            printf("Poor Movie\n"); // Print for poor movie
        } else if (rating < 4.0) {
            printf("Average Movie\n"); // Print for average movie
        } else {
            printf("Must Watch\n"); // Print for must watch
        }
    }
    return 0;
}
```

Explanation:

- The program reads the number of movies, *nn*n.
- It evaluates each movie's rating and prints the appropriate message.

Problem 35: Pet Adoption Assessment

Time limit: 1 second

Memory limit: 64 MB

Question:

Evaluate the suitability for pet adoption. If the adoption application score is below 50, print "Not Suitable"; if it’s between 50 and 80, print "Moderately Suitable"; if it’s above 80, print "Highly Suitable".

Input:

- The first line contains an integer *nn*n (number of applications).
- The second line contains *nn*n integers representing scores.

Output:

- For each application, print the corresponding message.

Code:

c

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

int main() {
    int n;
    scanf("%d", &n); // Read number of applications
    int score;

    for (int i = 0; i < n; i++) {
        scanf("%d", &score); // Read adoption application score
        if (score < 50) {
            printf("Not Suitable\n"); // Print for not suitable
        } else if (score <= 80) {
            printf("Moderately Suitable\n"); // Print for moderately
suitable
        } else {
            printf("Highly Suitable\n"); // Print for highly suitable
        }
    }
    return 0;
}
```

Explanation:

- The program reads the number of applications, *nn*n.
- It evaluates each application's score and prints the corresponding message.

Problem 36: Smartphone Battery Status

Time limit: 1 second

Memory limit: 64 MB

Question:

Assess smartphone battery levels. If the battery level is below 20%, print "Charge Now"; if it’s between 20% and 50%, print "Low Battery"; if it’s above 50%, print "Battery Good".

Input:

- The first line contains an integer nnn (number of devices).
- The second line contains nnn integers representing battery levels in percentage.

Output:

- For each device, print the corresponding message.

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of devices
    int battery_level;

    for (int i = 0; i < n; i++) {
        scanf("%d", &battery_level); // Read battery level
        if (battery_level < 20) {
            printf("Charge Now\n"); // Print for charge now
        } else if (battery_level <= 50) {
            printf("Low Battery\n"); // Print for low battery
        } else {
            printf("Battery Good\n"); // Print for good battery
        }
    }
    return 0;
}
```

Explanation:

- The program reads the number of devices, nnn.
- It evaluates each device's battery level and prints the relevant message.

Problem 37: Meal Prep Evaluation

Time limit: 1 second

Memory limit: 64 MB

Question:

Evaluate meal prep consistency. If meals prepared exceed 30 in a month, print "Meal Prep Master"; if it's between 15 and 30, print "Good Effort"; otherwise, print "Needs Improvement".

Input:

- The first line contains an integer nnn (number of individuals).
- The second line contains nnn integers representing meals prepared.

Output:

- For each individual, print the corresponding message.

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of individuals
    int meals_prepared;

    for (int i = 0; i < n; i++) {
        scanf("%d", &meals_prepared); // Read number of meals prepared
        if (meals_prepared > 30) {
            printf("Meal Prep Master\n"); // Print for meal prep
master
        } else if (meals_prepared >= 15) {
            printf("Good Effort\n"); // Print for good effort
        } else {
            printf("Needs Improvement\n"); // Print for needs
improvement
        }
    }
    return 0;
}
```

Explanation:

- The program reads the number of individuals, *nn*n.
- It evaluates each individual's meal prep count and prints the appropriate message.

Problem 38: Household Cleanliness Assessment

Time limit: 1 second

Memory limit: 64 MB

Question:

Evaluate household cleanliness ratings. If the rating is below 3, print "Needs Attention"; if it’s between 3 and 4, print "Generally Clean"; if it’s above 4, print "Spotless".

Input:

- The first line contains an integer *nn*n (number of households).
- The second line contains *nn*n integers representing cleanliness ratings.

Output:

- For each household, print the corresponding message.

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of households
    int cleanliness_rating;

    for (int i = 0; i < n; i++) {
        scanf("%d", &cleanliness_rating); // Read cleanliness rating
        if (cleanliness_rating < 3) {
            printf("Needs Attention\n"); // Print for needs attention
        } else if (cleanliness_rating <= 4) {
            printf("Generally Clean\n"); // Print for generally clean
        } else {
            printf("Spotless\n"); // Print for spotless
        }
    }
    return 0;
}
```

Explanation:

- The program reads the number of households, nnn.
- It evaluates each household's cleanliness rating and prints the relevant message.

Problem 39: Shopping Cart Evaluation

Time limit: 1 second

Memory limit: 64 MB

Question:

Evaluate the total cost of items in a shopping cart. If the total cost exceeds 200 units, print "Apply Discount"; if it’s between 100 and 200 units, print "Regular Price"; otherwise, print "Budget Friendly".

Input:

- The first line contains an integer nnn (number of shopping carts).
- The second line contains nnn integers representing total costs.

Output:

- For each cart, print the corresponding message.

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of shopping carts
    int total_cost;

    for (int i = 0; i < n; i++) {
        scanf("%d", &total_cost); // Read total cost
        if (total_cost > 200) {
            printf("Apply Discount\n"); // Print for discount
application
        } else if (total_cost >= 100) {
            printf("Regular Price\n"); // Print for regular price
        } else {
            printf("Budget Friendly\n"); // Print for budget friendly
        }
    }
    return 0;
}
```

Explanation:

- The program reads the number of shopping carts, nnn.
- It evaluates each cart's total cost and prints the appropriate message.

Problem 40: Health Check-Up Frequency

Time limit: 1 second

Memory limit: 64 MB

Question:

Evaluate the frequency of health check-ups. If a person has more than 2 check-ups per year, print "Health Conscious"; if it’s between 1 and 2, print "Occasional Check-Up"; otherwise, print "Need Regular Check-Up".

Input:

- The first line contains an integer nnn (number of individuals).
- The second line contains nnn integers representing the number of check-ups per year.

Output:

- For each individual, print the corresponding message.

Code:

```
c
Copy code
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n); // Read number of individuals
    int checkups;

    for (int i = 0; i < n; i++) {
        scanf("%d", &checkups); // Read number of check-ups
        if (checkups > 2) {
            printf("Health Conscious\n"); // Print for health
conscious
        } else if (checkups >= 1) {
            printf("Occasional Check-Up\n"); // Print for occasional
check-up
        } else {
            printf("Need Regular Check-Up\n"); // Print for need
regular check-up
        }
    }
    return 0;
}
```

Explanation:

- The program reads the number of individuals, nnn.
- It evaluates each individual's check-up frequency and prints the relevant message.