**Week-5 C programs**

1. **this program simulates a temperature monitoring system. It iterates over an array of temperature readings and uses a loop to check each value. If a reading exceeds predefined threshold, the program issues a alert and terminates further checks**

**Code:**

#include <stdio.h>

int main() {

float temperatures[] = {22.5, 23.1, 24.8, 26.5, 28.9, 30.2};

int num\_readings = sizeof(temperatures) / sizeof(temperatures[0]);

float threshold = 28.0;

for (int i = 0; i < num\_readings; i++) {

printf("Checking temperature: %.1f°C\n", temperatures[i]);

if (temperatures[i] > threshold) {

printf("ALERT! Temperature %.1f°C exceeds the threshold of %.1f°C.\n", temperatures[i], threshold);

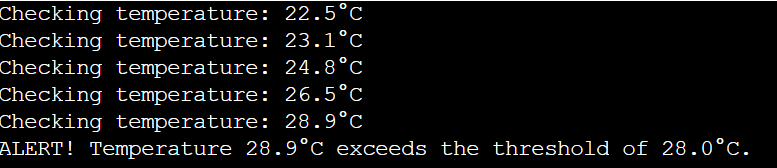
break;

}

}

return 0;

}

**Output**: 

1. **This simulation models an ATM machine. A user must enter a 4-digit pin with a maximum of three attempts. Once authenticated, the user can request a withdrawal. This program checks for a valid amount and sufficient funds before processing the transaction.**

**Code:**

#include <stdio.h>

#define MAX\_ATTEMPTS 3

int authenticate(const char \*correctPin) {

char enteredPin[5];

int attempts = 0;

while (attempts < MAX\_ATTEMPTS) {

printf("Enter your 4-digit PIN: ");

scanf("%4s", enteredPin);

if (strcmp(enteredPin, correctPin) == 0) {

printf("Authentication successful!\n");

return 1;

} else {

attempts++;

printf("Incorrect PIN. %d attempts remaining.\n", MAX\_ATTEMPTS - attempts);

}

}

printf("Too many incorrect attempts. Card blocked.\n");

return 0;

}

void withdraw(float \*balance) {

float amount;

printf("Enter amount to withdraw: ");

scanf("%f", &amount);

if (amount <= 0) {

printf("Invalid amount. Withdrawal must be positive.\n");

} else if (amount > \*balance) {

printf("Insufficient funds. Your balance is %.2f.\n", \*balance);

} else {

\*balance -= amount;

printf("Transaction successful. You have withdrawn %.2f.\n", amount);

printf("Your new balance is %.2f.\n", \*balance);

}

}

int main() {

const char \*correctPin = "1234";

float balance = 1000.00;

int authenticated = 0;

authenticated = authenticate(correctPin);

if (authenticated) {

withdraw(&balance);

} else {

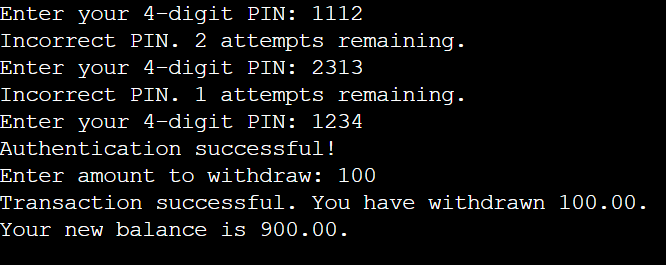
printf("Unable to authenticate user.\n");

}

return 0;

}

**Output:**

****

1. **write a program in C for voting eligibility using nested if statements to validate the age output. if age is negative, it flags an error. If the age is 18 it notifies the user that this is thier first time voting, if greater than 18, it simply confirms eligibility.**

**Code:**

#include <stdio.h>

int main() {

int age;

printf("Enter your age: ");

scanf("%d", &age);

if (age < 0) {

printf("Error: Age cannot be negative!\n");

} else {

if (age == 18) {

printf("This is your first time voting!\n");

} else if (age > 18) {

printf("You are eligible to vote.\n");

} else {

printf("You are not eligible to vote.\n");

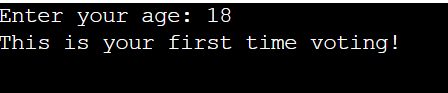
}

}

return 0;

}

**Output**:



1. **A traffic light simulation controls the flow of traffic at an intersection. The program cycles through red , yellow and green light. If an emergency vehicle is detected, the light immediately turns green.**

**Code:**

#include <stdio.h>

#include <stdbool.h>

#include <unistd.h>

void displayLightStatus(const char \*color) {

printf("The light is now %s.\n", color);

}

int main() {

int cycleCount = 0;

bool emergencyDetected = false;

while (1) {

printf("Is there an emergency vehicle? (1 for Yes, 0 for No): ");

scanf("%d", &emergencyDetected);

if (emergencyDetected) {

displayLightStatus("Green (Emergency Vehicle Detected)");

emergencyDetected = false;

} else {

if (cycleCount % 3 == 0) {

displayLightStatus("Red");

sleep(3);

} else if (cycleCount % 3 == 1) {

displayLightStatus("Yellow");

sleep(1);

} else {

displayLightStatus("Green");

sleep(3);

}

cycleCount++;

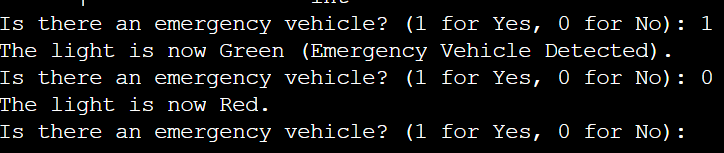
}

}

return 0;

}

**Output**:



1. **A teacher enters five student marks. IF a student score below 35, the evaluation stops immediately, and the system reports that the student has failed.**

**Code:**

#include <stdio.h>

int main() {

int marks[5];

int i;

for (i = 0; i < 5; i++) {

printf("Enter marks for student %d: ", i + 1);

scanf("%d", &marks[i]);

if (marks[i] < 35) {

printf("Student %d has failed. Evaluation stops here.\n", i + 1);

break;

}

}

return 0;

}

**Output**:

