**CODING**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

// Structure for currency rate

typedef struct

{

char code[4]; // Currency code like USD, INR, EUR

float rate; // Exchange rate relative to base currency

} Currency;

// Linked list node for conversion history

typedef struct HistoryNode

{

char from[4];

char to[4];

float amount;

float result;

struct HistoryNode\* next;

} HistoryNode;

#define NUM\_CURRENCIES 5

Currency currencies[NUM\_CURRENCIES] = {

{"INR", 1.0},

{"USD", 0.012},

{"EUR", 0.011},

{"GBP", 0.0095},

{"JPY", 1.75}

};

HistoryNode\* historyHead = NULL;

// Function to find the exchange rate of a currency code

float getRate(char code[])

{

for (int i = 0; i< NUM\_CURRENCIES; i++) {

if (strcmp(currencies[i].code, code) == 0) {

return currencies[i].rate;

}

}

return -1.0; // Error code for currency not found

}

// Function to add a conversion to history

void addToHistory(char from[], char to[], float amount, float result)

{

HistoryNode\*newNode =(HistoryNode\*)malloc(sizeof(HistoryNode));

strcpy(newNode->from, from);

strcpy(newNode->to, to);

newNode->amount = amount;

newNode->result = result;

newNode->next = historyHead;

historyHead = newNode;

}

// Function to display history

void displayHistory()

{

HistoryNode\* temp = historyHead;

if (temp == NULL)

{

printf("No conversions done yet.\n");

return;

}

printf("\n--- Conversion History ---\n");

while (temp != NULL)

{

printf("%.2f %s => %.2f %s\n", temp->amount, temp->from, temp->result, temp->to);

temp = temp->next;

}

printf("---------------------------\n");

}

// Function to perform currency conversion

void convertCurrency()

{

char from[4], to[4];

float amount, result;

printf("Enter source currency code (e.g., INR): ");

scanf("%s", from);

printf("Enter target currency code (e.g., USD): ");

scanf("%s", to);

printf("Enter amount: ");

scanf("%f", &amount);

float fromRate = getRate(from);

float toRate = getRate(to);

if (fromRate == -1 || toRate == -1)

{

printf("Invalid currency code entered.\n");

return;

}

result = (amount \* toRate) / fromRate;

printf("Converted Amount: %.2f %s\n", result, to);

addToHistory(from, to, amount, result);

}

// Function to free linked list memory

void freeHistory()

{

HistoryNode\* temp;

while (historyHead != NULL)

{

temp = historyHead;

historyHead = historyHead->next;

free(temp);

}

}

int main()

{

int choice;

do

{

printf("\n--- Currency Converter ---\n");

printf("1. Convert Currency\n");

printf("2. View History\n");

printf("3. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice)

{

case 1: convertCurrency();

break;

case 2: displayHistory();

break;

case 3: freeHistory(); printf("Exiting...\n");

break;

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

}

} while (choice != 3);

return 0;

}