; Currency Converter in Assembly using Irvine32

.386

.model flat, stdcall

.stack 4096

ExitProcess proto, dwExitCode:dword

INCLUDE Irvine32.inc

.data

; Define conversion rates (1 unit of currency X to PKR in cents for precision)

conversionRates DWORD 28500, 130000, 28500, 2200, 100 ; USD, EUR, GBP, INR, PKR

; Currency codes

currencies BYTE "USD", 0, "EUR", 0, "GBP", 0, "INR", 0, "PKR", 0

; Messages

promptFromCurrency BYTE "Choose the source currency (0-USD, 1-EUR, 2-GBP, 3-INR, 4-PKR): ", 0

promptToCurrency BYTE "Choose the target currency (0-USD, 1-EUR, 2-GBP, 3-INR, 4-PKR): ", 0

promptAmount BYTE "Enter the amount: ", 0

invalidInputMsg BYTE "Invalid input. Please enter a valid number.", 0

resultMsg BYTE "Converted amount: ", 0

menuMsg BYTE "Currency Converter Menu:", 13, 10, "1. Convert Currency", 13, 10, "2. View Conversion Rates", 13, 10, "3. Exit", 13, 10, "Choose an option: ", 0

conversionRatesMsg BYTE "Conversion Rates (1 unit to PKR):", 13, 10, "USD: 285.00", 13, 10, "EUR: 300.00", 13, 10, "GBP: 285.00", 13, 10, "INR: 22.00", 13, 10, 0

newline BYTE 13, 10, 0

amount DWORD ?

fromCurrency DWORD ?

toCurrency DWORD ?

convertedAmount DWORD ?

menuOption DWORD ?

.code

main PROC

; Main menu loop

menuLoop:

call DisplayMenu

call ReadMenuOption

cmp eax, 1

je ConvertCurrencyOption

cmp eax, 2

je ViewConversionRates

cmp eax, 3

je ExitProgram

jmp InvalidMenuOption

ConvertCurrencyOption:

call ChooseCurrencyFrom

call ChooseCurrencyTo

call GetAmount

call ConvertCurrency

call DisplayResult

jmp menuLoop

ViewConversionRates:

call DisplayConversionRates

jmp menuLoop

InvalidMenuOption:

mov edx, OFFSET invalidInputMsg

call WriteString

call Crlf

jmp menuLoop

ExitProgram:

INVOKE ExitProcess, 0

main ENDP

DisplayMenu PROC

mov edx, OFFSET menuMsg

call WriteString

ret

DisplayMenu ENDP

ReadMenuOption PROC

call ReadInt

mov menuOption, eax

ret

ReadMenuOption ENDP

ChooseCurrencyFrom PROC

; Prompt user for source currency

mov edx, OFFSET promptFromCurrency

call WriteString

; Read integer input

call ReadInt

mov fromCurrency, eax

; Validate input (0 to 4)

cmp eax, 4

ja InvalidInput

cmp eax, 0

jb InvalidInput

ret

InvalidInput:

mov edx, OFFSET invalidInputMsg

call WriteString

call Crlf

call ChooseCurrencyFrom

ret

ChooseCurrencyFrom ENDP

ChooseCurrencyTo PROC

; Prompt user for target currency

mov edx, OFFSET promptToCurrency

call WriteString

; Read integer input

call ReadInt

mov toCurrency, eax

; Validate input (0 to 4)

cmp eax, 4

ja InvalidInput

cmp eax, 0

jb InvalidInput

ret

InvalidInput:

mov edx, OFFSET invalidInputMsg

call WriteString

call Crlf

call ChooseCurrencyTo

ret

ChooseCurrencyTo ENDP

GetAmount PROC

; Prompt user for amount

mov edx, OFFSET promptAmount

call WriteString

; Read integer input

call ReadInt

mov amount, eax

ret

GetAmount ENDP

ConvertCurrency PROC

; Conversion: amount \* (conversionRates[toCurrency] / conversionRates[fromCurrency])

mov eax, toCurrency

mov ebx, DWORD PTR conversionRates[eax\*4] ; Get conversion rate for the target currency

mov edx, fromCurrency

mov ecx, DWORD PTR conversionRates[edx\*4] ; Get conversion rate for the source currency

mov eax, amount

imul ebx ; Multiply amount by target currency conversion rate

cdq

div ecx ; Divide by source currency conversion rate

mov convertedAmount, eax

ret

ConvertCurrency ENDP

DisplayResult PROC

; Display the result message

mov edx, OFFSET resultMsg

call WriteString

; Display converted amount

call Crlf

mov eax, convertedAmount

call WriteDec

call Crlf

ret

DisplayResult ENDP

DisplayConversionRates PROC

; Display conversion rates

mov edx, OFFSET conversionRatesMsg

call WriteString

call Crlf

ret

DisplayConversionRates ENDP

END main