#include <iostream>

#include <string>

#include <cmath>

using namespace std;

// Función que muestra el menú de opciones disponibles para el usuario

void showMenu() {

cout << "Seleccione una opción:" << endl;

cout << "1. Convertir de decimal a binario" << endl;

cout << "2. Convertir de binario a decimal" << endl;

cout << "3. Convertir de octal a decimal" << endl;

cout << "4. Convertir de hexadecimal a decimal" << endl;

cout << "5. Convertir de binario a hexadecimal" << endl;

cout << "6. Convertir de binario a octal" << endl;

cout << "7. Convertir de decimal a hexadecimal" << endl;

cout << "8. Convertir de decimal a octal" << endl;

cout << "9. Convertir de hexadecimal a binario" << endl;

cout << "10. Convertir de hexadecimal a octal" << endl;

cout << "11. Convertir de octal a binario" << endl;

cout << "12. Convertir de octal a hexadecimal" << endl;

cout << "13. Salir" << endl;

}

// Función que convierte un número decimal a binario

void decimalABinario() {

int numero\_decimal;

cout << "Ingrese un número decimal: ";

cin >> numero\_decimal; // Se solicita al usuario ingresar un número decimal

string binario = "";

// Si el número decimal es 0, se asigna "0" a la cadena binario

if (numero\_decimal == 0) binario = "0";

// Se realiza la conversión de decimal a binario

while (numero\_decimal > 0) {

binario = to\_string(numero\_decimal % 2) + binario; // Se obtiene el residuo de la división entre 2 y se concatena

numero\_decimal = numero\_decimal / 2; // Se divide el número decimal por 2

}

cout << "En binario es " << binario << endl; // Se muestra el resultado en binario

}

// Función que convierte un número binario a decimal

void binarioADecimal() {

string numero\_binario;

int numero\_decimal = 0;

int potencia = 0;

cout << "Ingrese un número binario: ";

cin >> numero\_binario; // Se solicita al usuario ingresar un número binario

// Se recorre la cadena binaria de derecha a izquierda

for (int i = numero\_binario.length() - 1; i >= 0; i--) {

if (numero\_binario[i] == '1') {

numero\_decimal += pow(2, potencia); // Si el dígito es '1', se suma 2^potencia al número decimal

}

potencia++; // Se incrementa la potencia

}

cout << "En decimal es " << numero\_decimal << endl; // Se muestra el resultado en decimal

}

// Función que convierte un número octal a decimal

void octalADecimal() {

string numero\_octal;

int numero\_decimal = 0;

int potencia = 0;

cout << "Ingrese un número en octal: ";

cin >> numero\_octal; // Se solicita al usuario ingresar un número en octal

// Se recorre la cadena octal de derecha a izquierda

for (int i = numero\_octal.length() - 1; i >= 0; i--) {

numero\_decimal += (numero\_octal[i] - '0') \* pow(8, potencia); // Se suma el valor del dígito multiplicado por 8^potencia al número decimal

potencia++; // Se incrementa la potencia

}

cout << "En decimal es " << numero\_decimal << endl; // Se muestra el resultado en decimal

}

// Función que convierte un número hexadecimal a decimal

void hexaADecimal() {

string numero\_hexadecimal;

int numero\_decimal = 0;

int potencia = 0;

cout << "Ingrese un número en hexadecimal: ";

cin >> numero\_hexadecimal; // Se solicita al usuario ingresar un número en hexadecimal

// Se recorre la cadena hexadecimal de derecha a izquierda

for (int i = numero\_hexadecimal.length() - 1; i >= 0; i--) {

char c = toupper(numero\_hexadecimal[i]); // Se convierte el carácter a mayúscula

int digito;

// Se determina el valor decimal del dígito hexadecimal

if (c >= '0' && c <= '9') {

digito = c - '0'; // Si el dígito es numérico, se resta '0'

} else {

digito = c - 'A' + 10; // Si el dígito es una letra, se calcula su valor correspondiente

}

numero\_decimal += digito \* pow(16, potencia); // Se suma el valor del dígito multiplicado por 16^potencia al número decimal

potencia++; // Se incrementa la potencia

}

cout << "En decimal es " << numero\_decimal << endl; // Se muestra el resultado en decimal

}

// Función que convierte un número binario a hexadecimal

void binarioAHexadecimal() {

string numero\_binario;

cout << "Ingrese un número binario: ";

cin >> numero\_binario; // Se solicita al usuario ingresar un número binario

int numero\_decimal = 0;

int potencia = 0;

// Se convierte el número binario a decimal

for (int i = numero\_binario.length() - 1; i >= 0; i--) {

if (numero\_binario[i] == '1') {

numero\_decimal += pow(2, potencia); // Si el dígito es '1', se suma 2^potencia al número decimal

}

potencia++; // Se incrementa la potencia

}

string hexadecimal = "";

// Si el número decimal es 0, se asigna "0" a la cadena hexadecimal

if (numero\_decimal == 0) hexadecimal = "0";

// Se realiza la conversión de decimal a hexadecimal

while (numero\_decimal > 0) {

int resto = numero\_decimal % 16; // Se obtiene el residuo de la división entre 16

if (resto < 10) {

hexadecimal = to\_string(resto) + hexadecimal; // Si el residuo es menor a 10, se concatena el dígito

} else {

hexadecimal = char(resto - 10 + 'A') + hexadecimal; // Si el residuo es mayor o igual a 10, se concatena la letra correspondiente

}

numero\_decimal /= 16; // Se divide el número decimal por 16

}

cout << "En hexadecimal es " << hexadecimal << endl; // Se muestra el resultado en hexadecimal

}

// Función que convierte un número binario a octal

void binarioAOctal() {

string numero\_binario;

cout << "Ingrese un número binario: ";

cin >> numero\_binario; // Se solicita al usuario ingresar un número binario

int numero\_decimal = 0;

int potencia = 0;

// Se convierte el número binario a decimal

for (int i = numero\_binario.length() - 1; i >= 0; i--) {

if (numero\_binario[i] == '1') {

numero\_decimal += pow(2, potencia); // Si el dígito es '1', se suma 2^potencia al número decimal

}

potencia++; // Se incrementa la potencia

}

string octal = "";

// Si el número decimal es 0, se asigna "0" a la cadena octal

if (numero\_decimal == 0) octal = "0";

// Se realiza la conversión de decimal a octal

while (numero\_decimal > 0) {

int resto = numero\_decimal % 8; // Se obtiene el residuo de la división entre 8

octal = to\_string(resto) + octal; // Se concatena el dígito octal

numero\_decimal /= 8; // Se divide el número decimal por 8

}

cout << "En octal es " << octal << endl; // Se muestra el resultado en octal

}

// Función que convierte un número decimal a hexadecimal

void decimalAHexadecimal() {

int numero\_decimal;

cout << "Ingrese un número decimal: ";

cin >> numero\_decimal; // Se solicita al usuario ingresar un número decimal

string hexadecimal = "";

// Si el número decimal es 0, se asigna "0" a la cadena hexadecimal

if (numero\_decimal == 0) hexadecimal = "0";

// Se realiza la conversión de decimal a hexadecimal

while (numero\_decimal > 0) {

int resto = numero\_decimal % 16; // Se obtiene el residuo de la división entre 16

if (resto < 10) {

hexadecimal = to\_string(resto) + hexadecimal; // Si el residuo es menor a 10, se concatena el dígito

} else {

hexadecimal = char(resto - 10 + 'A') + hexadecimal; // Si el residuo es mayor o igual a 10, se concatena la letra correspondiente

}

numero\_decimal /= 16; // Se divide el número decimal por 16

}

cout << "En hexadecimal es " << hexadecimal << endl; // Se muestra el resultado en hexadecimal

}

// Función que convierte un número decimal a octal

void decimalAOctal() {

int numero\_decimal;

cout << "Ingrese un número decimal: ";

cin >> numero\_decimal; // Se solicita al usuario ingresar un número decimal

string octal = "";

// Si el número decimal es 0, se asigna "0" a la cadena octal

if (numero\_decimal == 0) octal = "0";

// Se realiza la conversión de decimal a octal

while (numero\_decimal > 0) {

int resto = numero\_decimal % 8; // Se obtiene el residuo de la división entre 8

octal = to\_string(resto) + octal; // Se concatena el dígito octal

numero\_decimal /= 8; // Se divide el número decimal por 8

}

cout << "En octal es " << octal << endl; // Se muestra el resultado en octal

}

// Función que convierte un número hexadecimal a binario

void hexadecimalABinario() {

string numero\_hexadecimal;

cout << "Ingrese un número en hexadecimal: ";

cin >> numero\_hexadecimal; // Se solicita al usuario ingresar un número en hexadecimal

int numero\_decimal = 0;

int potencia = 0;

// Se convierte el número hexadecimal a decimal

for (int i = numero\_hexadecimal.length() - 1; i >= 0; i--) {

char c = toupper(numero\_hexadecimal[i]); // Se convierte el carácter a mayúscula

int digito;

// Se determina el valor decimal del dígito hexadecimal

if (c >= '0' && c <= '9') {

digito = c - '0'; // Si el dígito es numérico, se resta '0'

} else {

digito = c - 'A' + 10; // Si el dígito es una letra, se calcula su valor correspondiente

}

numero\_decimal += digito \* pow(16, potencia); // Se suma el valor del dígito multiplicado por 16^potencia al número decimal

potencia++; // Se incrementa la potencia

}

string binario = "";

// Si el número decimal es 0, se asigna "0" a la cadena binario

if (numero\_decimal == 0) binario = "0";

// Se realiza la conversión de decimal a binario

while (numero\_decimal > 0) {

binario = to\_string(numero\_decimal % 2) + binario; // Se obtiene el residuo de la división entre 2 y se concatena

numero\_decimal /= 2; // Se divide el número decimal por 2

}

cout << "En binario es " << binario << endl; // Se muestra el resultado en binario

}

// Función que convierte un número hexadecimal a octal

void hexadecimalAOctal() {

string numero\_hexadecimal;

cout << "Ingrese un número en hexadecimal: ";

cin >> numero\_hexadecimal; // Se solicita al usuario ingresar un número en hexadecimal

int numero\_decimal = 0;

int potencia = 0;

// Se convierte el número hexadecimal a decimal

for (int i = numero\_hexadecimal.length() - 1; i >= 0; i--) {

char c = toupper(numero\_hexadecimal[i]); // Se convierte el carácter a mayúscula

int digito;

// Se determina el valor decimal del dígito hexadecimal

if (c >= '0' && c <= '9') {

digito = c - '0'; // Si el dígito es numérico, se resta '0'

} else {

digito = c - 'A' + 10; // Si el dígito es una letra, se calcula su valor correspondiente

}

numero\_decimal += digito \* pow(16, potencia); // Se suma el valor del dígito multiplicado por 16^potencia al número decimal

potencia++; // Se incrementa la potencia

}

string octal = "";

// Si el número decimal es 0, se asigna "0" a la cadena octal

if (numero\_decimal == 0) octal = "0";

// Se realiza la conversión de decimal a octal

while (numero\_decimal > 0) {

int resto = numero\_decimal % 8; // Se obtiene el residuo de la división entre 8

octal = to\_string(resto) + octal; // Se concatena el dígito octal

numero\_decimal /= 8; // Se divide el número decimal por 8

}

cout << "En octal es " << octal << endl; // Se muestra el resultado en octal

}

// Función que convierte un número octal a binario

void octalABinario() {

string numero\_octal;

cout << "Ingrese un número en octal: ";

cin >> numero\_octal; // Se solicita al usuario ingresar un número en octal

int numero\_decimal = 0;

int potencia = 0;

// Se convierte el número octal a decimal

for (int i = numero\_octal.length() - 1; i >= 0; i--) {

numero\_decimal += (numero\_octal[i] - '0') \* pow(8, potencia); // Se suma el valor del dígito multiplicado por 8^potencia al número decimal

potencia++; // Se incrementa la potencia

}

string binario = "";

// Si el número decimal es 0, se asigna "0" a la cadena binario

if (numero\_decimal == 0) binario = "0";

// Se realiza la conversión de decimal a binario

while (numero\_decimal > 0) {

binario = to\_string(numero\_decimal % 2) + binario; // Se obtiene el residuo de la división entre 2 y se concatena

numero\_decimal /= 2; // Se divide el número decimal por 2

}

cout << "En binario es " << binario << endl; // Se muestra el resultado en binario

}

// Función que convierte un número octal a hexadecimal

void octalAHexadecimal() {

string numero\_octal;

cout << "Ingrese un número en octal: ";

cin >> numero\_octal; // Se solicita al usuario ingresar un número en octal

int numero\_decimal = 0;

int potencia = 0;

// Se convierte el número octal a decimal

for (int i = numero\_octal.length() - 1; i >= 0; i--) {

numero\_decimal += (numero\_octal[i] - '0') \* pow(8, potencia); // Se suma el valor del dígito multiplicado por 8^potencia al número decimal

potencia++; // Se incrementa la potencia

}

string hexadecimal = "";

// Si el número decimal es 0, se asigna "0" a la cadena hexadecimal

if (numero\_decimal == 0) hexadecimal = "0";

// Se realiza la conversión de decimal a hexadecimal

while (numero\_decimal > 0) {

int resto = numero\_decimal % 16; // Se obtiene el residuo de la división entre 16

if (resto < 10) {

hexadecimal = to\_string(resto) + hexadecimal; // Si el residuo es menor a 10, se concatena el dígito

} else {

hexadecimal = char(resto - 10 + 'A') + hexadecimal; // Si el residuo es mayor o igual a 10, se concatena la letra correspondiente

}

numero\_decimal /= 16; // Se divide el número decimal por 16

}

cout << "En hexadecimal es " << hexadecimal << endl; // Se muestra el resultado en hexadecimal

}

int main() {

int opcion;

while (true) {

showMenu(); // Se muestra el menú de opciones

cin >> opcion; // Se lee la opción seleccionada por el usuario

switch (opcion) {

case 1:

decimalABinario(); // Conversión de decimal a binario

break;

case 2:

binarioADecimal(); // Conversión de binario a decimal

break;

case 3:

octalADecimal(); // Conversión de octal a decimal

break;

case 4:

hexaADecimal(); // Conversión de hexadecimal a decimal

break;

case 5:

binarioAHexadecimal(); // Conversión de binario a hexadecimal

break;

case 6:

binarioAOctal(); // Conversión de binario a octal

break;

case 7:

decimalAHexadecimal(); // Conversión de decimal a hexadecimal

break;

case 8:

decimalAOctal(); // Conversión de decimal a octal

break;

case 9:

hexadecimalABinario(); // Conversión de hexadecimal a binario

break;

case 10:

hexadecimalAOctal(); // Conversión de hexadecimal a octal

break;

case 11:

octalABinario(); // Conversión de octal a binario

break;

case 12:

octalAHexadecimal(); // Conversión de octal a hexadecimal

break;

case 13:

return 0; // Salir del programa

default:

cout << "Opción no válida. Intente de nuevo." << endl; // Mensaje para opción no válida

break;

}

}

}