#include <conio.h>

#include <stdio.h>

#include <math.h>

#include <string.h>

void dec\_bin();

void dec\_oct();

void dec\_hex();

void bin\_dec();

void bin\_oct();

void bin\_hex();

void oct\_dec();

void oct\_bin();

void oct\_hex();

void hex\_dec();

void hex\_bin();

void hex\_oct();

int main()

{

int input;

char choice;

printf ("\t\tNumber System Conversions\n");

do

{

printf ("\n\nUse numbers to navigate the menu.");

printf ("\n1.Decimal to Binary");

printf ("\n2.Decimal to Octal");

printf ("\n3.Decimal to Hexadecimal");

printf ("\n4.Binary to Decimal");

printf ("\n5.Binary to Octal");

printf ("\n6.Binary to Hexadecimal");

printf ("\n7.Octal to Decimal");

printf ("\n8.Octal to Binary");

printf ("\n9.Octal to Hexadecimal");

printf ("\n10.Hexadecimal to Decimal");

printf ("\n11.Hexadecimal to Binaary");

printf ("\n12.Hexadecimal to Octal");

printf ("\n13.Quit");

printf ("\nEnter: ");

scanf ("%d",&input);

switch(input)

{

case 1:

printf ("\n\t\tDecimal To Binary");

dec\_bin();

break;

case 2:

printf ("\n\t\tDecimal to Octal");

dec\_oct();

break;

case 3:

printf ("\n\t\tDecimal to Hexadecimal");

dec\_hex();

break;

case 4:

printf ("\n\t\tBinary to Decimal");

bin\_dec();

break;

case 5:

printf ("\n\t\tBinary to Octal");

bin\_oct();

break;

case 6:

printf ("\n\t\tBinary to Hexadecimal");

bin\_hex();

break;

case 7:

printf ("\n\t\tOctal to Decimal");

oct\_dec();

break;

case 8:

printf ("\n\t\tOctal to Binary");

oct\_bin();

break;

case 9:

printf ("\n\t\tOctal to Hexadecimal");

oct\_hex();

break;

case 10:

printf ("\n\t\tHexadecimal to Decimal");

hex\_dec();

break;

case 11:

printf ("\n\t\tHexadecimal to Binary");

hex\_bin();

break;

case 12:

printf ("\n\t\tHexadecimal to Octal");

hex\_oct();

break;

case 13:

printf ("Exiting...");

return 0;

default:

printf ("Invalid Selection.");

}

printf ("\n\nContinue? Y/N ");

choice=getch();

}

while (choice == 'y' || choice == 'Y');

return 0;

}

void dec\_bin()

{

long int dec\_num,remainder,quotient;

int binary\_num [100],i=1,j;

printf("\nEnter any decimal number: ");

scanf("%ld",&dec\_num);

quotient=dec\_num;

while(quotient!=0)

{

binary\_num[i++]=quotient%2;

quotient=quotient/2;

}

printf("Equivalent binary value of the given decimal number %d is : ",dec\_num);

for(j=i-1;j>0;j--)

{

printf("%d",binary\_num[j]);

}

}

void dec\_oct()

{

long int dec\_num,remainder,quotient;

int octal\_num[100],i=1,j;

printf("\nEnter any decimal number: ");

scanf("%ld",&dec\_num);

quotient=dec\_num;

while(quotient!=0)

{

octal\_num[i++]=quotient%8;

quotient=quotient/8;

}

printf("Equivalent octal value of given decimal number %ld is: ",dec\_num);

for(j=i-1;j>0;j--)

{

printf("%d",octal\_num[j]);

}

}

void dec\_hex()

{

long int dec\_num,remainder,quotient;

int i=1,j,k;

char hexa\_dec\_num[100];

printf("\nEnter any decimal number : ");

scanf("%ld",&dec\_num);

quotient=dec\_num;

while(quotient!=0)

{

k=quotient%16;

if(k<10)

{

k=k+48;

}

else

{

k=k+55;

}

hexa\_dec\_num[i++]=k;

quotient=quotient/16;

}

printf("Equivalent hexadecimal value of given decimal number %ld is : ",dec\_num);

for(j=i-1;j>0;j--)

{

printf("%c",hexa\_dec\_num[j]);

}

}

void bin\_dec()

{

long int binary\_num,binary\_numb,dec\_num=0,j=1,remainder;

printf("\nEnter any binary number: ");

scanf("%ld",&binary\_num);

binary\_numb=binary\_num;

while(binary\_num!=0)

{

remainder=binary\_num%10;

dec\_num=dec\_num+remainder\*j;

j=j\*2;

binary\_num=binary\_num/10;

}

printf("Equivalent decimal value of given binary number %ld is : %ld",binary\_numb,dec\_num);

}

void bin\_oct()

{

long int bin, oct = 0, i = 1, rem;

printf("\nEner a Binary Number ");

scanf("%ld", &bin);

while (bin != 0)

{

rem = bin % 10;

oct = oct + rem \* i;

i = i \* 2;

bin = bin / 10;

}

printf("Equivalent Octal number: %lo", oct);

}

void bin\_hex()

{

long int bin, hex = 0, i = 1, rem;

printf("\nEnter a Binary number: ");

scanf("%ld", &bin);

while (bin != 0)

{

rem = bin % 10;

hex = hex + rem \* i;

i = i \* 2;

bin = bin / 10;

}

printf("Equivalent Hexadecimal value: %lX", hex);

}

void oct\_dec()

{

long int octal\_num,octal\_numb,decimal\_num=0;

int i=0;

printf("\nEnter any octal number : ");

scanf("%ld",&octal\_num);

octal\_numb=octal\_num;

while(octal\_num!=0)

{

decimal\_num = decimal\_num + (octal\_num % 10) \* pow(8,i++);

octal\_num = octal\_num/10;

}

printf("Equivalent decimal value of given octal number %ld is : %ld",octal\_numb,decimal\_num);

}

void oct\_bin()

{

char octal\_num[1000];

long int i=0;

printf("\nEnter any octal number : ");

scanf("%s",octal\_num);

printf("Equivalent binary value of given octal number %s is : ",octal\_num);

while(octal\_num[i])

{

switch(octal\_num[i])

{

case '0':

{

printf("000");

break;

}

case '1':

{

printf("001");

break;

}

case '2':

{

printf("010");

break;

}

case '3':

{

printf("011");

break;

}

case '4':

{

printf("100");

break;

}

case '5':

{

printf("101");

break;

}

case '6':

{

printf("110");

break;

}

case '7':

{

printf("111");

break;

}

default:

{

printf("\nInvalid octal digit %c",octal\_num[i]);

}

}

i++;

}

}

void oct\_hex()

{

int OCTALVALUES[] = {0, 1, 10, 11, 100, 101, 110, 111};

long long oct, tempOct, bin=0, place=1;

char hex[65] = "";

int rem;

printf("\nEnter a Octal number: ");

scanf("%lld", &oct);

tempOct = oct;

/\*Octal to binary conversion\*/

while(tempOct > 0)

{

rem = tempOct % 10;

bin = (OCTALVALUES[rem] \* place) + bin;

tempOct /= 10;

place \*= 1000;

}

while(bin > 0)

{

rem = bin % 10000;

switch(rem)

{

case 0:

strcat(hex, "0");

break;

case 1:

strcat(hex, "1");

break;

case 10:

strcat(hex, "2");

break;

case 11:

strcat(hex, "3");

break;

case 100:

strcat(hex, "4");

break;

case 101:

strcat(hex, "5");

break;

case 110:

strcat(hex, "6");

break;

case 111:

strcat(hex, "7");

break;

case 1000:

strcat(hex, "8");

break;

case 1001:

strcat(hex, "9");

break;

case 1010:

strcat(hex, "A");

break;

case 1011:

strcat(hex, "B");

break;

case 1100:

strcat(hex, "C");

break;

case 1101:

strcat(hex, "D");

break;

case 1110:

strcat(hex, "E");

break;

case 1111:

strcat(hex, "F");

break;

}

bin /= 10000;

}

strrev(hex);

printf("Equivalent Hexadecimal number: %s", hex);

}

void hex\_dec()

{

char hex[17];

long long dec=0, place=1;

int i = 0, val, len;

printf("\nEnter a Hexadecimal number: ");

scanf ("%s", hex);

len = strlen(hex);

len--;

/\*Iterate over each hex digit\*/

for(i=0; hex[i]!='\0'; i++)

{

if(hex[i]>='0' && hex[i]<='9')

{

val = hex[i] - 48;

}

else if(hex[i]>='a' && hex[i]<='f')

{

val = hex[i] - 97 + 10;

}

else if(hex[i]>='A' && hex[i]<='F')

{

val = hex[i] - 65 + 10;

}

dec += val \* pow(16, len);

len--;

}

printf("Equivalent Decimal number: %lld", dec);

}

void hex\_bin()

{

char bin[1000], hex[1000];

long int i = 0;

printf("\nEnter a Hexadecimal number: ");

scanf("%s", hex);

printf("Equivalent Binary Number: ");

while (hex[i])

{

switch (hex[i])

{

case '0':

printf("0000");

break;

case '1':

printf("0001");

break;

case '2':

printf("0010");

break;

case '3':

printf("0011");

break;

case '4':

printf("0100");

break;

case '5':

printf("0101");

break;

case '6':

printf("0110");

break;

case '7':

printf("0111");

break;

case '8':

printf("1000");

break;

case '9':

printf("1001");

break;

case 'A':

printf("1010");

break;

case 'B':

printf("1011");

break;

case 'C':

printf("1100");

break;

case 'D':

printf("1101");

break;

case 'E':

printf("1110");

break;

case 'F':

printf("1111");

break;

case 'a':

printf("1010");

break;

case 'b':

printf("1011");

break;

case 'c':

printf("1100");

break;

case 'd':

printf("1101");

break;

case 'e':

printf("1110");

break;

case 'f':

printf("1111");

break;

default:

printf("\nInvalid hexa digit %c ", hex[i]);

}

i++;

}

}

void hex\_oct()

{

char hex[17];

long long oct, bin, place;

int i = 0, rem, val;

printf("\nEnter a Hexadecimal number: ");

scanf ("%s", hex);

oct = 0ll;

bin = 0ll;

place = 0ll;

for(i=0; hex[i]!='\0'; i++)

{

bin = bin \* place;

switch(hex[i])

{

case '0':

bin += 0;

break;

case '1':

bin += 1;

break;

case '2':

bin += 10;

break;

case '3':

bin += 11;

break;

case '4':

bin += 100;

break;

case '5':

bin += 101;

break;

case '6':

bin += 110;

break;

case '7':

bin += 111;

break;

case '8':

bin += 1000;

break;

case '9':

bin += 1001;

break;

case 'a':

case 'A':

bin += 1010;

break;

case 'b':

case 'B':

bin += 1011;

break;

case 'c':

case 'C':

bin += 1100;

break;

case 'd':

case 'D':

bin += 1101;

break;

case 'e':

case 'E':

bin += 1110;

break;

case 'f':

case 'F':

bin += 1111;

break;

default:

printf("Invalid hexadecimal input.");

}

place = 10000;

}

place = 1;

while(bin > 0)

{

rem = bin % 1000;

switch(rem)

{

case 0:

val = 0;

break;

case 1:

val = 1;

break;

case 10:

val = 2;

break;

case 11:

val = 3;

break;

case 100:

val = 4;

break;

case 101:

val = 5;

break;

case 110:

val = 6;

break;

case 111:

val = 7;

break;

}

oct = (val \* place) + oct;

bin /= 1000;

place \*= 10;

}

printf("Equivalent Octal number = %lld", oct);

}