**WAP in C to take two input in binary, decimal, octal and hexadecimal number. Convert it to other. Also perform addition, subtraction, multiplication & division.**

**#include <math.h>**

**#include <string.h>**

**#include <stdio.h>**

**#include <conio.h>**

**double calculator3binarytodecimal(double);**

**void calculator3decimaltobinary(double);**

**void calculator3decimaltooctal(double);**

**void calculator3decimaltohexadecimal(double);**

**double calculator3octaltodecimal(double);**

**double calculator3hexadecimaltodecimal(char \*);**

**int count;**

**long long int i,j,k,l,n,a,b,c,d,f,g,h;**

**double a1,a2,b1,b2,c1,c2,d1,d2,e1,e2,f1,f2,g1,g2;**

**char hexa1[100]= {0},hexa2[100]= {0},binary[100];**

**int main()**

**{**

**system("mode 700");**

**while(1)**

**{**

**system("cls");**

**printf("\n\n\t\t\tEnter the key for required type of input:");**

**printf("\n\n\t\t\t1 Binary number\t\t\t2 Decimal number");**

**printf("\n\n\t\t\t3 Octal number\t\t\t4 Hexadecimal number");**

**printf("\n\n\t\t\t\tEnter a number from 1 to 4:\t");**

**scanf("%d",&a);**

**system("cls");**

**if (a==1)**

**{**

**printf("\n\n\t\t\tEnter first binary number: ");**

**scanf("%lf",&a1);**

**printf("\n\t\t\tEnter second binary number: ");**

**scanf("%lf",&a2);**

**printf("\n\n\t\t\t%lf to decimal is %lf\n",a1,calculator3binarytodecimal(a1));**

**printf("\n\n\t\t\t%lf to decimal is %lf\n",a2,calculator3binarytodecimal(a2));**

**printf("\n\n\t\t\t%lf to octal is ",a1);**

**calculator3decimaltooctal(calculator3binarytodecimal(a1));**

**printf("\n\n\t\t\t%lf to octal is ",a2);**

**calculator3decimaltooctal(calculator3binarytodecimal(a2));**

**printf("\n\n\t\t\t%lf to hexadecimal is ",a1);**

**calculator3decimaltohexadecimal(calculator3binarytodecimal(a1));**

**printf("\n\n\t\t\t%lf to hexadecimal is ",a2);**

**calculator3decimaltohexadecimal(calculator3binarytodecimal(a2));**

**printf("\n\n\t\t\t%lf + %lf = ",a1,a2);**

**calculator3decimaltobinary(calculator3binarytodecimal(a1)+calculator3binarytodecimal(a2));**

**printf("\n\n\t\t\t%lf - %lf = ",a1,a2);**

**calculator3decimaltobinary(calculator3binarytodecimal(a1)-calculator3binarytodecimal(a2));**

**printf("\n\n\t\t\t%lf - %lf = ",a2,a1);**

**calculator3decimaltobinary(calculator3binarytodecimal(a2)-calculator3binarytodecimal(a1));**

**printf("\n\n\t\t\t%lf \* %lf = ",a1,a2);**

**calculator3decimaltobinary(calculator3binarytodecimal(a1)\*calculator3binarytodecimal(a2));**

**printf("\n\n\t\t\t%lf / %lf = ",a1,a2);**

**calculator3decimaltobinary(calculator3binarytodecimal(a1)/calculator3binarytodecimal(a2));**

**printf("\n\n\t\t\t%lf / %lf = ",a2,a1);**

**calculator3decimaltobinary(calculator3binarytodecimal(a2)/calculator3binarytodecimal(a1));**

**}**

**if (a==2)**

**{**

**printf("\n\n\t\t\tEnter decimal number: ");**

**scanf("%lf",&a1);**

**printf("\n\n\t\t\t%lf to binary is ",a1);**

**calculator3decimaltobinary(a1);**

**printf("\n\n\t\t\t%lf to octal is ",a1);**

**calculator3decimaltooctal(a1);**

**printf("\n\n\t\t\t%lf to hexadecimal is ",a1);**

**calculator3decimaltohexadecimal(a1);**

**}**

**if (a==3)**

**{**

**printf("\n\n\t\t\tEnter first octal number: ");**

**scanf("%lf",&a1);**

**printf("\n\t\t\tEnter second octal number: ");**

**scanf("%lf",&a2);**

**printf("\n\n\t\t\t%lf to binary is ",a1);**

**calculator3decimaltobinary(calculator3octaltodecimal(a1));**

**printf("\n\n\t\t\t%lf to binary is ",a2);**

**calculator3decimaltobinary(calculator3octaltodecimal(a1));**

**printf("\n\n\t\t\t%lf to decimal is %lf",a1,calculator3octaltodecimal(a1));**

**printf("\n\n\t\t\t%lf to decimal is %lf",a2,calculator3octaltodecimal(a2));**

**printf("\n\n\t\t\t%lf to hexadecimal is ",a1);**

**calculator3decimaltohexadecimal(calculator3octaltodecimal(a1));**

**printf("\n\n\t\t\t%lf to hexadecimal is ",a2);**

**calculator3decimaltohexadecimal(calculator3octaltodecimal(a2));**

**printf("\n\n\t\t\t%lf + %lf = ",a1,a2);**

**calculator3decimaltooctal(calculator3octaltodecimal(a1)+calculator3octaltodecimal(a2));**

**printf("\n\n\t\t\t%lf - %lf = ",a1,a2);**

**calculator3decimaltooctal(calculator3octaltodecimal(a1)-calculator3octaltodecimal(a2));**

**printf("\n\n\t\t\t%lf - %lf = ",a2,a1);**

**calculator3decimaltooctal(calculator3octaltodecimal(a2)-calculator3octaltodecimal(a1));**

**printf("\n\n\t\t\t%lf \* %lf = ",a1,a2);**

**calculator3decimaltooctal(calculator3octaltodecimal(a1)\*calculator3octaltodecimal(a2));**

**printf("\n\n\t\t\t%lf / %lf = ",a1,a2);**

**calculator3decimaltooctal(calculator3octaltodecimal(a1)/calculator3octaltodecimal(a2));**

**printf("\n\n\t\t\t%lf / %lf = ",a2,a1);**

**calculator3decimaltooctal(calculator3octaltodecimal(a2)/calculator3octaltodecimal(a1));**

**}**

**if (a==4)**

**{**

**printf("\n\n\t\t\tEnter first hexadecimal number: ");**

**scanf("%s",&hexa1);**

**printf("\n\t\t\tEnter second hexadecimal number: ");**

**scanf("%s",&hexa2);**

**printf("\n\n\t\t\t%s to binary is ",hexa1);**

**calculator3decimaltobinary(calculator3hexadecimaltodecimal(hexa1));**

**printf("\n\n\t\t\t%s to binary is ",hexa2);**

**calculator3decimaltobinary(calculator3hexadecimaltodecimal(hexa2));**

**printf("\n\n\t\t\t%s to decimal is %lf",hexa1,calculator3hexadecimaltodecimal(hexa1));**

**printf("\n\n\t\t\t%s to decimal is %lf",hexa2,calculator3hexadecimaltodecimal(hexa2));**

**printf("\n\n\t\t\t%s to octal is ",hexa1);**

**calculator3decimaltooctal(calculator3hexadecimaltodecimal(hexa1));**

**printf("\n\n\t\t\t%s to octal is ",hexa2);**

**calculator3decimaltooctal(calculator3hexadecimaltodecimal(hexa2));**

**printf("\n\n\t\t\t%s + %s = ",hexa1,hexa2);**

**calculator3decimaltohexadecimal(calculator3hexadecimaltodecimal(hexa1)+calculator3hexadecimaltodecimal(hexa2));**

**printf("\n\n\t\t\t%s - %s = ",hexa1,hexa2);**

**calculator3decimaltohexadecimal(calculator3hexadecimaltodecimal(hexa1)-calculator3hexadecimaltodecimal(hexa2));**

**printf("\n\n\t\t\t%s - %s = ",hexa2,hexa1);**

**calculator3decimaltohexadecimal(calculator3hexadecimaltodecimal(hexa2)-calculator3hexadecimaltodecimal(hexa1));**

**printf("\n\n\t\t\t%s \* %s = ",hexa1,hexa2);**

**calculator3decimaltohexadecimal(calculator3hexadecimaltodecimal(hexa1)\*calculator3hexadecimaltodecimal(hexa2));**

**printf("\n\n\t\t\t%s / %s = ",hexa1,hexa2);**

**calculator3decimaltohexadecimal(calculator3hexadecimaltodecimal(hexa1)/calculator3hexadecimaltodecimal(hexa2));**

**printf("\n\n\t\t\t%s / %s = ",hexa2,hexa1);**

**calculator3decimaltohexadecimal(calculator3hexadecimaltodecimal(hexa2)/calculator3hexadecimaltodecimal(hexa2));**

**}**

**getch();**

**}**

**return 0;**

**}**

**double calculator3binarytodecimal(double b1) /\*Function to return value\*/**

**{**

**b=(int)b1;**

**c1=b1-(int)b1;**

**c=0;**

**d=0;**

**while (b>0)**

**{**

**d+=(b%10)\*pow(2,c);**

**c++;**

**b/=10;**

**}**

**d1=0;**

**for(i=1; i<=10; i++)**

**{**

**c1\*=10;**

**d1+=((int)c1)\*pow(2,-i);**

**c1=c1-(int)c1;**

**}**

**d1+=d;**

**return d1;**

**}**

**void calculator3decimaltobinary(double b1)**

**{**

**c=(int)b1;**

**c1=b1-(int)b1;**

**g=0;**

**while (c!=0)**

**{**

**if (c%2==0)**

**binary[g]='0';**

**else**

**binary[g]='1';**

**c/=2;**

**g++;**

**}**

**printf("%s.",strrev(binary));**

**for(i=0; i<30; i++)**

**{**

**c1\*=2;**

**if ((int)c1==1 || (int)c1==0) printf("%lli",(int)c1);**

**c1=c1-(int)c1;**

**}**

**printf("\n");**

**}**

**void calculator3decimaltooctal(double b1)**

**{**

**b=(int)b1;**

**c1=b1-(int)b1;**

**printf("%llo.",b);**

**for(i=0; i<30; i++)**

**{**

**c1\*=8;**

**printf("%llo",(int)c1);**

**c1=c1-(int)c1;**

**}**

**printf("\n");**

**}**

**void calculator3decimaltohexadecimal(double b1)**

**{**

**b=(int)b1;**

**c1=b1-(int)b1;**

**printf("%llX.",b);**

**for(i=0; i<30; i++)**

**{**

**c1\*=16;**

**printf("%llX",(int)c1);**

**c1=c1-(int)c1;**

**}**

**printf("\n");**

**}**

**double calculator3octaltodecimal(double b1)/\*Function to return value\*/**

**{**

**b=(int)b1;**

**c1=b1-(int)b1;**

**d=0;**

**c=0;**

**while (b>0)**

**{**

**d+=(b%10)\*pow(8,c);**

**c++;**

**b/=10;**

**}**

**d1=0;**

**for(i=1; i<=10; i++)**

**{**

**c1\*=10;**

**d1+=((int)c1)\*pow(8,-i);**

**c1=c1-(int)c1;**

**}**

**d1+=d;**

**return d1;**

**}**

**double calculator3hexadecimaltodecimal(char \*hexa)/\*Function to return value\*/**

**{**

**c=0;**

**for(i=0; i<strlen(hexa); i++)**

**{**

**if (hexa[i]!='.') c++;**

**else break;**

**}**

**d=0;**

**j=0;**

**for(i=c-1; i>=0; i--)**

**{**

**if (hexa[i]>='0' && hexa[i]<='9')/\*ASCII of 0 = 48 & 9 = 57\*/**

**{**

**d+=(hexa[i]-48)\*pow(16,j);**

**}**

**else if (hexa[i]>='A' && hexa[i]<='F')/\*ASCII of A = 65 & Z = 90\*/**

**{**

**d+=(hexa[i]-55)\*pow(16,j);**

**}**

**else if(hexa[i]>='a' && hexa[i]<='f')/\*ASCII of a = 97 & z = 122\*/**

**{**

**d+=(hexa[i]-87)\*pow(16,j);**

**}**

**j++;**

**}**

**j=1;**

**d1=0;**

**for(i=c+1; i<strlen(hexa); i--)**

**{**

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

**{**

**d1+=(hexa[i]-48)\*pow(16,-j);**

**}**

**if (hexa[i]>='A' && hexa[i]<='F')**

**{**

**d1+=(hexa[i]-55)\*pow(16,-j);**

**}**

**if(hexa[i]>='a' && hexa[i]<='f')**

**{**

**d1+=(hexa[i]-87)\*pow(16,-j);**

**}**

**j++;**

**}**

**d1+=d;**

**return d1;**

**}**