```
/*################################################################*/
/* HW05_Samet_Sait_Talayhan_101044044.c                   */
/* ----------------------                                         */
/* Created on March 31, 2013, 1:52 PM by Samet Sait Talayhan.    */
/*                                                                */
/* Description                                                    */
/* ----------------------
 * This implementation a new representation usin 3 integers "s",
 * "i", "f" to represent sign, integer, and the fractional parts.
/* ----------------------
/*                                                                */
/*################################################################*/

/*################################################################*/
/*                           Includes                             */
/*################################################################*/
#include <stdio.h>
#include <math.h>       /* for math functions i.e pow,sin */

/*################################################################*/
/*                           Defines                              */
/*################################################################*/
#define PI 3.141592653589793238462643383

/*----------------------------------------------------------------*/
/*                    Typedef Declaration                         */
/*----------------------------------------------------------------*/
typedef enum{
    add,subtract,multiply,divide        /* add = 0, subtract = 1, .. */
}operator_t;

/*----------------------------------------------------------------*/
/*                    Function Prototypes                         */
/*----------------------------------------------------------------*/
double calculatePI();
int castFromDoubleToInteger(double castedNumber,int *s,int *i, int *f);
int castFromIntegerToDouble(double *castedNumber,int s,int i, int f);
int calculateNumberOfDigits(double theNumber);
int addTwoNumber(int s1, int i1, int f1,
                 int s2, int i2, int f2,
                 int *outS, int *totalI, int *totalF);
int subtractTwoNumber(int s1, int i1, int f1,
                 int s2, int i2, int f2,
                 int *outS, int *totalI, int *totalF);
int multiplyTwoNumber(int s1, int i1, int f1,
                 int s2, int i2, int f2,
                 int *outS, int *totalI, int *totalF);
int divideTwoNumber(int s1, int i1, int f1,
                 int s2, int i2, int f2,
                 int *outS, int *totalI, int *totalF);
void function(operator_t operator);


/*################################################################*/
/* int main()                                                     */
/* ----------                                                     */
/* Return                                                         */
/* ----------                                                     */
/*0 on success                                                    */
/*################################################################*/
int
main(int argc, char const *argv[])
{
    double test = 1357.4005;
    int i=0,s=0,f=0;

    castFromDoubleToInteger(test,&s,&i,&f);
    printf("%d\t%d\t%d\n",s,i,f);

    test = 0;

    castFromIntegerToDouble(&test,s,i,f);
    printf("%.4f\n",test);

    /* Test function */
    addTwoNumber(1,125,2500,1,175,7501,&s,&i,&f);
    castFromIntegerToDouble(&test,s,i,f);
    printf("\n\nTest Functions\n\n%.4f\n",test);

    subtractTwoNumber(1,125,2500,1,175,7501,&s,&i,&f);
    castFromIntegerToDouble(&test,s,i,f);
    printf("%.4f\n",test);

    multiplyTwoNumber(1,125,2500,1,175,7501,&s,&i,&f);
    castFromIntegerToDouble(&test,s,i,f);
    printf("%.4f\n",test);

    divideTwoNumber(1,125,2500,1,175,7501,&s,&i,&f);
    castFromIntegerToDouble(&test,s,i,f);
    printf("%.4f\n",test);

    return 0;
}
/*----------------------------------------------------------------*/
/*               Function Implementations                         */
/*----------------------------------------------------------------*/
/* castFromDoubleToInteger Function                               */
/* -------------------                                            */
/* This function to cast from double to 3 integer representation.
/*----------------------------------------------------------------*/
int castFromDoubleToInteger(double castedNumber,int *s,int *i, int *f)
{
    int numberOfDigitsOfCastedNumber = calculateNumberOfDigits(castedNumber);
    if (castedNumber < 0)
    {
        *s = -1;
        *i = ceil(castedNumber);
        *f = -(castedNumber - (double)*i) * 10000;
    }
    else if (castedNumber == 0) /* Special case */
    {
        *s = +1;
        *i = 0;
        *f = 0;
        return 0;
    }
    else
    {
        *s = +1;
        *i = floor(castedNumber);
        *f = (castedNumber - (double)*i) * 10000;
    }

}
/*----------------------------------------------------------------*/
/*               Function Implementations                         */
/*----------------------------------------------------------------*/
/* castFromIntegerToDouble Function                               */
/* -------------------                                            */
/* This function to cast from 3 integer to double representation.
/*----------------------------------------------------------------*/
int castFromIntegerToDouble(double *castedNumber,int s,int i, int f)
{
    int numberOfDigitsOfI = calculateNumberOfDigits((double)i);
    *castedNumber = (double)(i);
    *castedNumber += (double)(f)/10000;

    if( s < 0)
    {
        *castedNumber = -(*castedNumber);
    }

    return 0;
}
/*----------------------------------------------------------------*/
/*               Function Implementations                         */
/*----------------------------------------------------------------*/
/* calculateNumberOfDigits Function                               */
/* -------------------                                            */
/* This function calculate number of digits that the given number.
/*----------------------------------------------------------------*/
int calculateNumberOfDigits(double theNumber)
{
    if (theNumber>=1000 && theNumber<10000)
    {
        return 4;
    }
    else if (theNumber>=100 && theNumber<1000)
    {
        return 3;
    }
    else if (theNumber>=10 && theNumber<100)
    {
        return 2;
    }
    else if (theNumber>=0 && theNumber<10)
    {
        return 1;
    }
    else
    {
        printf("Invalid double number: %f|\tError code -1!\n",theNumber);
        return -1;
    }
}
/*----------------------------------------------------------------*/
/*               Function Implementations                         */
/*----------------------------------------------------------------*/
/* addTwoNumber Function                                      */
/* -------------------                                            */
/* This function to add 2 variables (3 integer representation)
/*----------------------------------------------------------------*/
int addTwoNumber(int s1, int i1, int f1,
                 int s2, int i2, int f2,
                 int *outS, int *totalI, int *totalF)
{
    double number1;
    double number2;
    double sum;
    /* convert from 3 integer to double */
    castFromIntegerToDouble(&number1,s1,i1,f1);
    castFromIntegerToDouble(&number2,s2,i2,f2);
    sum = number1 + number2;
    /* re-convert from double to 3 integer */
    castFromDoubleToInteger(sum,outS,totalI,totalF);
}
/*----------------------------------------------------------------*/
/*               Function Implementations                         */
/*----------------------------------------------------------------*/
/* subtractTwoNumber Function                                     */
```

```c
/* -------------------                                            */
/* This function to subtract 2 variables (3 integer representation)
/*----------------------------------------------------------------*/
int subtractTwoNumber(int s1, int i1, int f1,
                 int s2, int i2, int f2,
                 int *outS, int *totalI, int *totalF)
{
    double number1;
    double number2;
    double sum;
    /* convert from 3 integer to double */
    castFromIntegerToDouble(&number1,s1,i1,f1);
    castFromIntegerToDouble(&number2,s2,i2,f2);
    sum = number1 - number2;
    /* re-convert from double to 3 integer */
    castFromDoubleToInteger(sum,outS,totalI,totalF);
}
/*----------------------------------------------------------------*/
/*               Function Implementations                         */
/*----------------------------------------------------------------*/
/* multiplyTwoNumber Function                                     */
/* -------------------                                            */
/* This function to multiply 2 variables (3 integer representation)
/*----------------------------------------------------------------*/
int multiplyTwoNumber(int s1, int i1, int f1,
                 int s2, int i2, int f2,
                 int *outS, int *totalI, int *totalF)
{
    double number1;
    double number2;
    double sum;
    /* convert from 3 integer to double */
    castFromIntegerToDouble(&number1,s1,i1,f1);
    castFromIntegerToDouble(&number2,s2,i2,f2);
    sum = number1 * number2;
    /* re-convert from double to 3 integer */
    castFromDoubleToInteger(sum,outS,totalI,totalF);
}
/*----------------------------------------------------------------*/
/*               Function Implementations                         */
/*----------------------------------------------------------------*/
/* divideTwoNumber Function                                   */
/* -------------------                                            */
/* This function to divide 2 variables (3 integer representation)
/*----------------------------------------------------------------*/
int divideTwoNumber(int s1, int i1, int f1,
                 int s2, int i2, int f2,
                 int *outS, int *totalI, int *totalF)
 {
    double number1;
    double number2;
    double sum;
    /* convert from 3 integer to double */
    castFromIntegerToDouble(&number1,s1,i1,f1);
    castFromIntegerToDouble(&number2,s2,i2,f2);
    sum = number1 / number2;
    /* re-convert from double to 3 integer */
    castFromDoubleToInteger(sum,outS,totalI,totalF);
}

void function(operator_t operator)
{
}
/*----------------------------------------------------------------*/
/*               Function Implementations                         */
/*----------------------------------------------------------------*/
/* calculatePI Function                                           */
/* -------------------                                            */
/* This function calculate PI value by given formulation.
/*----------------------------------------------------------------*/
double calculatePI()
{
    int terms,i;
    double pi = 0.0, error = 0.0;

    printf("Enter terms: ");
    scanf("%d",&terms);

    pi += 3.0;
    for(i=0;i<terms;++i)
    {
        pi += 4 * ( pow(-1,i)/(2*i*(2*i+1)*(2*i+2)) );
    }

    error = ((pi - PI)/PI) * 100;

    return error;
}
/*################################################################*/
/*        End of HW05_Samet_Sait_Talayhan_101044044.c             */
/*################################################################*/
```