

```

/*#####*/
/* HW02_Samet_Sait_Talayhan_101044044.c */
/* ----- */
/* Created on March 7, 2013, 8:16 PM by Samet Sait Talayhan. */
/* ----- */
/* Description */
/* ----- */
/* This program calculates and print the result of the operations */
/* written in a text file. */
/* Example of a text file pattern "4_25*50+115-1" */
/* ----- */
/* #####- Bonus Part only doesnt work file has 5 integer.! */
/* #####- Otherwise works clearly. */
/* ----- */
/* ----- */
/*#####*/
/*#####*/
/* Includes */
/*#####*/
#include <stdio.h>
#include <math.h>

/*#####*/
/* Defines */
/*#####*/
#define TRUE 1
#define FALSE 0

/*-----*/
/* Function Prototypes */
/*-----*/
int convertToInt1d(char ch);
int convertToInt2d(char ch1, char ch2);
int convertToInt3d(char ch1, char ch2, char ch3);
int isDigit(char ch);
int isSign(char ch);
int isPlusSign(char ch);
int isMinusSign(char ch);
int isMultiplicationSign(char ch);
int isDivisionSign(char ch);
int convertToInt(char ch);
int calculate(int leftInt, int rightInt, char sign);
int uniteToNumber(int size, char chr1, char chr2, char chr3);

/*#####*/
/* int main() */
/* ----- */
/* Return */
/* ----- */
/*0 on success */
/*#####*/
int
main(void)
{
    /* Declaring Variables */
    int numOfint = 0; /* first char in the text. */
    int int1Done = FALSE;
    int int2Done = FALSE;
    int int3Done = FALSE;
    int int4Done = FALSE;
    int int5Done = FALSE;
    int result, bonusPartResult, temp; /* temp process for operator precedence */
    char int1Chr1, int1Chr2, int1Chr3; int int1Size=1;
    char int2Chr1, int2Chr2, int2Chr3; int int2Size=1;
    char int3Chr1, int3Chr2, int3Chr3; int int3Size=1;
    char int4Chr1, int4Chr2, int4Chr3; int int4Size=1;

```

```
char int5Chr1, int5Chr2, int5Chr3; int int5Size=1;

char sign1, sign2, sign3, sign4;    /* Maximum number of operators */
char whiteSpaceChar;
char numOfChr;

/*Declaring File pointers */
FILE *inFile = fopen("text.txt","r");

fscanf(inFile,"%c",&numOfChr);
numOfint = convertToInt(numOfChr);
fscanf(inFile,"%c",&whiteSpaceChar);

if (numOfint == 2)
{
    /* Read char by char an integer from the file. */
    fscanf(inFile,"%c",&int1Chr1);
    fscanf(inFile,"%c",&int1Chr2);
    if (isSign(int1Chr2))
    {
        int1Done = TRUE;
        int1Size = 1;
        sign1 = int1Chr2;
    }

    if (int1Done != TRUE)
    {
        fscanf(inFile,"%c",&int1Chr3);
        if (isSign(int1Chr3))
        {
            int1Done = TRUE;
            int1Size = 2;
            sign1 = int1Chr3;
        }
        else /* Otherwise absolutly become sign1 */
        {
            fscanf(inFile,"%c",&sign1);
            int1Done = TRUE;
            int1Size = 3;
        }
    }

    /* Read second integer */
    fscanf(inFile,"%c",&int2Chr1);
    fscanf(inFile,"%c",&int2Chr2);
    if ( !isSign(int2Chr2) && !isDigit(int2Chr2) )
    {
        int2Done = TRUE;
        int2Size = 1;
    }

    if (int2Done != TRUE)
    {
        fscanf(inFile,"%c",&int2Chr3);
        if ( !isSign(int2Chr3) && !isDigit(int2Chr3) )
        {
            int2Done = TRUE;
            int2Size = 2;
        }
        else /* Otherwise absolutly become sign1 */
        {
            int2Done = TRUE;
            int2Size = 3;
        }
    }

    /* Calculate Operation */
    int1Done = uniteToNumber(int1Size,int1Chr1,int1Chr2,int1Chr3);
```

```
    int2Done = uniteToNumber(int2Size,int2Chr1,int2Chr2,int2Chr3);
    result = calculate(int1Done, int2Done, sign1);
}
else if (numOfint == 3)
{
    /* Read char by char an integer from the file. */
    fscanf(inFile,"%c",&int1Chr1);
    fscanf(inFile,"%c",&int1Chr2);
    if (isSign(int1Chr2))
    {
        int1Done = TRUE;
        int1Size = 1;
        sign1 = int1Chr2;
    }

    if (int1Done != TRUE)
    {
        fscanf(inFile,"%c",&int1Chr3);
        if (isSign(int1Chr3))
        {
            int1Done = TRUE;
            int1Size = 2;
            sign1 = int1Chr3;
        }
        else /* Otherwise absolutly become sign1 */
        {
            fscanf(inFile,"%c",&sign1);
            int1Done = TRUE;
            int1Size = 3;
        }
    }

    /* Read second integer */
    fscanf(inFile,"%c",&int2Chr1);
    fscanf(inFile,"%c",&int2Chr2);
    if (isSign(int2Chr2))
    {
        int2Done = TRUE;
        int2Size = 1;
        sign2 = int2Chr2;
    }

    if (int2Done != TRUE)
    {
        fscanf(inFile,"%c",&int2Chr3);
        if (isSign(int2Chr3))
        {
            int2Done = TRUE;
            int2Size = 2;
            sign2 = int2Chr3;
        }
        else /* Otherwise absolutly become sign1 */
        {
            fscanf(inFile,"%c",&sign2);
            int2Done = TRUE;
            int2Size = 3;
        }
    }

    /* Read third integer */
    fscanf(inFile,"%c",&int3Chr1);
    fscanf(inFile,"%c",&int3Chr2);
    if ( !isSign(int3Chr2) && !isDigit(int3Chr2) )
    {
        int3Done = TRUE;
        int3Size = 1;
    }
}
```

```
if (int3Done != TRUE)
{
    fscanf(inFile,"%c",&int3Chr3);
    if ( !isSign(int3Chr3) && !isDigit(int3Chr3) )
    {
        int3Done = TRUE;
        int3Size = 2;
    }
    else /* Otherwise absolutly become sign1 */
    {
        int3Done = TRUE;
        int3Size = 3;
    }
}
/* Calculate Operation */
int1Done = uniteToNumber(int1Size,int1Chr1,int1Chr2,int1Chr3);
int2Done = uniteToNumber(int2Size,int2Chr1,int2Chr2,int2Chr3);
int3Done = uniteToNumber(int3Size,int3Chr1,int3Chr2,int3Chr3);
result = calculate(int1Done, int2Done, sign1);
result = calculate(result, int3Done, sign2);

/* Bonus Part Implementation */
if (sign1 == '*' || sign1 == '/')
{
    bonusPartResult = calculate(int1Done, int2Done, sign1);
    bonusPartResult = calculate(bonusPartResult, int3Done, sign2);
}
else
{
    bonusPartResult = calculate(int2Done, int3Done, sign2);
    bonusPartResult = calculate(int1Done, bonusPartResult, sign1);
}
}
else if (numOfint == 4)
{
    /* Read char by char an integer from the file. */
    fscanf(inFile,"%c",&int1Chr1);
    fscanf(inFile,"%c",&int1Chr2);
    if (isSign(int1Chr2))
    {
        int1Done = TRUE;
        int1Size = 1;
        sign1 = int1Chr2;
    }

    if (int1Done != TRUE)
    {
        fscanf(inFile,"%c",&int1Chr3);
        if (isSign(int1Chr3))
        {
            int1Done = TRUE;
            int1Size = 2;
            sign1 = int1Chr3;
        }
        else /* Otherwise absolutly become sign1 */
        {
            fscanf(inFile,"%c",&sign1);
            int1Done = TRUE;
            int1Size = 3;
        }
    }

    /* Read second integer */
    fscanf(inFile,"%c",&int2Chr1);
    fscanf(inFile,"%c",&int2Chr2);
    if (isSign(int2Chr2))
```

```
{
    int2Done = TRUE;
    int2Size = 1;
    sign2 = int2Chr2;
}

if (int2Done != TRUE)
{
    fscanf(inFile,"%c",&int2Chr3);
    if (isSign(int2Chr3))
    {
        int2Done = TRUE;
        int2Size = 2;
        sign2 = int2Chr3;
    }
    else /* Otherwise absolutly become sign1 */
    {
        fscanf(inFile,"%c",&sign2);
        int2Done = TRUE;
        int2Size = 3;
    }
}

/* Read third integer */
fscanf(inFile,"%c",&int3Chr1);
fscanf(inFile,"%c",&int3Chr2);
if (isSign(int3Chr2))
{
    int3Done = TRUE;
    int3Size = 1;
    sign3 = int3Chr2;
}

if (int3Done != TRUE)
{
    fscanf(inFile,"%c",&int3Chr3);
    if (isSign(int3Chr3))
    {
        int3Done = TRUE;
        int3Size = 2;
        sign3 = int3Chr3;
    }
    else /* Otherwise absolutly become sign1 */
    {
        fscanf(inFile,"%c",&sign3);
        int3Done = TRUE;
        int3Size = 3;
    }
}

/* Read fourth integer */
fscanf(inFile,"%c",&int4Chr1);
fscanf(inFile,"%c",&int4Chr2);
if ( !isSign(int4Chr2) && !isDigit(int4Chr2) )
{
    int4Done = TRUE;
    int4Size = 1;
}

if (int4Done != TRUE)
{
    fscanf(inFile,"%c",&int4Chr3);
    if (!isSign(int4Chr3) && !isDigit(int4Chr3) )
    {
        int4Done = TRUE;
        int4Size = 2;
    }
}
```

```

        else /* Otherwise absolutly become sign1 */
        {
            int4Done = TRUE;
            int4Size = 3;
        }
    }
    /* Calculate Operation */
    int1Done = uniteToNumber(int1Size,int1Chr1,int1Chr2,int1Chr3);
    int2Done = uniteToNumber(int2Size,int2Chr1,int2Chr2,int2Chr3);
    int3Done = uniteToNumber(int3Size,int3Chr1,int3Chr2,int3Chr3);
    int4Done = uniteToNumber(int4Size,int4Chr1,int4Chr2,int4Chr3);
    result = calculate(int1Done, int2Done, sign1);
    result = calculate(result, int3Done, sign2);
    result = calculate(result, int4Done, sign3);

    /* Bonus Part Implementation */
    if (sign1 == '*' || sign1 == '/')
    {
        bonusPartResult = calculate(int1Done, int2Done, sign1);
        if (sign2 == '*' || sign2 == '/')
        {
            bonusPartResult = calculate(bonusPartResult, int3Done, sign2);
            bonusPartResult = calculate(bonusPartResult, int4Done, sign3);
        }
        else
        {
            temp = calculate(int3Done, int4Done, sign3);
            bonusPartResult = calculate(bonusPartResult, temp, sign2);
        }
    }
    else
    {
        if (sign2 == '*' || sign2 == '/')
        {
            bonusPartResult = calculate(int2Done, int3Done, sign2);
            bonusPartResult = calculate(bonusPartResult, int4Done, sign3);
            bonusPartResult = calculate(int1Done, bonusPartResult, sign1);
        }
        else
        {
            bonusPartResult = calculate(int3Done, int4Done, sign3);
            bonusPartResult = calculate(int2Done, bonusPartResult, sign2);
            bonusPartResult = calculate(int1Done, bonusPartResult, sign1);
        }
    }
}
else if (numOfint == 5)
{
    /* Read char by char an integer from the file. */
    fscanf(inFile,"%c",&int1Chr1);
    fscanf(inFile,"%c",&int1Chr2);
    if (isSign(int1Chr2))
    {
        int1Done = TRUE;
        int1Size = 1;
        sign1 = int1Chr2;
    }

    if (int1Done != TRUE)
    {
        fscanf(inFile,"%c",&int1Chr3);
        if (isSign(int1Chr3))
        {
            int1Done = TRUE;
            int1Size = 2;
            sign1 = int1Chr3;
        }
    }
}

```

```
        else    /* Otherwise absolutly become sign1 */
        {
            fscanf(inFile,"%c",&sign1);
            int1Done = TRUE;
            int1Size = 3;
        }
    }

    /* Read second integer */
    fscanf(inFile,"%c",&int2Chr1);
    fscanf(inFile,"%c",&int2Chr2);
    if (isSign(int2Chr2))
    {
        int2Done = TRUE;
        int2Size = 1;
        sign2 = int2Chr2;
    }

    if (int2Done != TRUE)
    {
        fscanf(inFile,"%c",&int2Chr3);
        if (isSign(int2Chr3))
        {
            int2Done = TRUE;
            int2Size = 2;
            sign2 = int2Chr3;
        }
        else    /* Otherwise absolutly become sign1 */
        {
            fscanf(inFile,"%c",&sign2);
            int2Done = TRUE;
            int2Size = 3;
        }
    }

    /* Read third integer */
    fscanf(inFile,"%c",&int3Chr1);
    fscanf(inFile,"%c",&int3Chr2);
    if (isSign(int3Chr2))
    {
        int3Done = TRUE;
        int3Size = 1;
        sign3 = int3Chr2;
    }

    if (int3Done != TRUE)
    {
        fscanf(inFile,"%c",&int3Chr3);
        if (isSign(int3Chr3))
        {
            int3Done = TRUE;
            int3Size = 2;
            sign3 = int3Chr3;
        }
        else    /* Otherwise absolutly become sign1 */
        {
            fscanf(inFile,"%c",&sign3);
            int3Done = TRUE;
            int3Size = 3;
        }
    }

    /* Read fourth integer */
    fscanf(inFile,"%c",&int4Chr1);
    fscanf(inFile,"%c",&int4Chr2);
    if (isSign(int4Chr2))
    {
```

```

        int4Done = TRUE;
        int4Size = 1;
        sign4 = int4Chr2;
    }

    if (int4Done != TRUE)
    {
        fscanf(inFile,"%c",&int4Chr3);
        if (isSign(int4Chr3))
        {
            int4Done = TRUE;
            int4Size = 2;
            sign4 = int4Chr3;
        }
        else /* Otherwise absolutly become sign1 */
        {
            fscanf(inFile,"%c",&sign4);
            int4Done = TRUE;
            int4Size = 3;
        }
    }

    /* Read fifth integer */
    fscanf(inFile,"%c",&int5Chr1);
    fscanf(inFile,"%c",&int5Chr2);
    if ( !isSign(int5Chr2) && !isDigit(int5Chr2) )
    {
        int5Done = TRUE;
        int5Size = 1;
    }

    if (int5Done != TRUE)
    {
        fscanf(inFile,"%c",&int5Chr3);
        if ( !isSign(int5Chr3) && !isDigit(int5Chr3) )
        {
            int5Done = TRUE;
            int5Size = 2;
        }
        else /* Otherwise absolutly become sign1 */
        {
            int3Done = TRUE;
            int3Size = 3;
        }
    }

    /* Calculate Operation */
    int1Done = uniteToNumber(int1Size,int1Chr1,int1Chr2,int1Chr3);
    int2Done = uniteToNumber(int2Size,int2Chr1,int2Chr2,int2Chr3);
    int3Done = uniteToNumber(int3Size,int3Chr1,int3Chr2,int3Chr3);
    int4Done = uniteToNumber(int4Size,int4Chr1,int4Chr2,int4Chr3);
    int5Done = uniteToNumber(int5Size,int5Chr1,int5Chr2,int5Chr3);
    result = calculate(int1Done, int2Done, sign1);
    result = calculate(result, int3Done, sign2);
    result = calculate(result, int4Done, sign3);
    result = calculate(result, int5Done, sign4);
}
else
{
    printf("Text file is not valid!\tFix iT!\n");
}

/* Close file. */
fclose(inFile);

printf("\n_____RESULT_____\n");
printf("Result is :%5d\n",result);
printf("BonusPart Result is: %5d\n",bonusPartResult );

```



```

    printf("\n_____ \n");

    return 0;
}
/*-----*/
/*          Function Implementations          */
/*-----*/
/* convertToInt1d Function                    */
/* ----- */
/* Converts the given character to its equivalent integer,
/* prints error message if the input doesnt refer a number
/*-----*/
int
convertToInt1d(char ch)
{
    return convertToInt(ch);
}
/*-----*/
/*          Function Implementations          */
/*-----*/
/* convertToInt2d Function                    */
/* ----- */
/* Converts the given characters to their equivalent integer of
/* two digit, prints error message if the input doesnt refer
/* a number.
/*-----*/
int
convertToInt2d(char ch1, char ch2)
{
    return 10*convertToInt(ch1) + convertToInt(ch2);
}
/*-----*/
/*          Function Implementations          */
/*-----*/
/* convertToInt3d Function                    */
/* ----- */
/* Converts the given characters to their equivalent integer of
/* three digit, prints error message if the input doesnt refer
/* a number.
/*-----*/
int
convertToInt3d(char ch1, char ch2, char ch3)
{
    return 100*convertToInt(ch1) + 10*convertToInt(ch2)
        + convertToInt(ch3);
}
/*-----*/
/*          Function Implementations          */
/*-----*/
/* isDigit Function                          */
/* ----- */
/* Returns 1 if the input refers a number and 0 otherwise
/*-----*/
int
isDigit(char ch)
{
    switch(ch)
    {
        case '0':
            return 1;
            break;
        case '1':
            return 1;
            break;
        case '2':
            return 1;
            break;
    }
}

```

```
        case '3':
            return 1;
            break;
        case '4':
            return 1;
            break;
        case '5':
            return 1;
            break;
        case '6':
            return 1;
            break;
        case '7':
            return 1;
            break;
        case '8':
            return 1;
            break;
        case '9':
            return 1;
            break;
        default :
            return 0;
    }
}

/*-----*/
/*          Function Implementations          */
/*-----*/
/* isPlusSign Function                        */
/* ----- */
/* Returns 1 if the input refers a plus sign and 0 otherwise */
/*-----*/
int
isPlusSign(char ch)
{
    if (ch == '+')
    {
        return 1;
    }
    else
    {
        return 0;
    }
}

/*-----*/
/*          Function Implementations          */
/*-----*/
/* isMinusSign Function                      */
/* ----- */
/* Returns 1 if the input refers a minus sign and 0 otherwise */
/*-----*/
int
isMinusSign(char ch)
{
    if (ch == '-')
    {
        return 1;
    }
    else
    {
        return 0;
    }
}

/*-----*/
/*          Function Implementations          */
/*-----*/
/* isMultiplicationSign Function            */
```

```
/* ----- */
/* Returns 1 if the input refers a multiplication sign and
/* 0 otherwise
/* ----- */
int
isMultiplicationSign(char ch)
{
    if (ch == '*')
    {
        return 1;
    }
    else
    {
        return 0;
    }
}
/* ----- */
/*           Function Implementations           */
/* ----- */
/* isDivisionSign Function                      */
/* ----- */
/* Returns 1 if the input refers division sign and 0 otherwise
/* ----- */
int
isDivisionSign(char ch)
{
    if (ch == '/')
    {
        return 1;
    }
    else
    {
        return 0;
    }
}
/* ----- */
/*           Function Implementations           */
/* ----- */
/* isSign Function                            */
/* ----- */
/* Returns 1 if the input refers any operator sign and 0 otherwise
/* ----- */
int
isSign(char ch)
{
    if (ch == '+')
    {
        return 1;
    }
    else if (ch == '-')
    {
        return 1;
    }
    else if (ch == '*')
    {
        return 1;
    }
    else if (ch == '/')
    {
        return 1;
    }
    else
    {
        return 0;
    }
}
/* ----- */
```

```

/*          Function Implementations          */
/*-----*/
/* convertToInt Function                      */
/*-----*/
/* Input - Character that keep integer char. ie. '4', '5' single
/* digit.
/* Output - Integer value of character
/*-----*/
int
convertToInt(char ch)
{
    switch(ch)
    {
        case '0':
            return 0;
            break;
        case '1':
            return 1;
            break;
        case '2':
            return 2;
            break;
        case '3':
            return 3;
            break;
        case '4':
            return 4;
            break;
        case '5':
            return 5;
            break;
        case '6':
            return 6;
            break;
        case '7':
            return 7;
            break;
        case '8':
            return 8;
            break;
        case '9':
            return 9;
            break;
    }
}
/*-----*/
/*          Function Implementations          */
/*-----*/
/* calculate Function                        */
/*-----*/
/* Input - leftInt and rightInt of a binary operator, and operator
/* sign as char.
/* Output - Result of the given process.
/*-----*/
int
calculate(int leftInt, int rightInt, char sign)
{
    if (isPlusSign(sign))
    {
        return (leftInt + rightInt);
    }
    else if (isMinusSign(sign))
    {
        return (leftInt - rightInt);
    }
    else if (isMultiplicationSign(sign))
    {

```

```
        return (leftInt * rightInt);
    }
    else if (isDivisionSign(sign))
    {
        return (leftInt / rightInt);
    }
}

int
uniteToNumber(int size, char chr1, char chr2, char chr3)
{
    switch(size)
    {
        case 1:
            return convertToInt1d(chr1);
            break;
        case 2:
            return convertToInt2d(chr1, chr2);
            break;
        case 3:
            return convertToInt3d(chr1, chr2, chr3);
            break;
        default:
            printf("Error Invalid size!\n");
            return -1;
    }
}

/*#####*/
/*      End of HW02_Samet_Sait_Talayhan_101044044.c      */
/*#####*/
```