CHAPTER TWENTY

More Issues In Input/Output

[A] Answer the following:

(a) How will you use the following program to

- Copy the contents of one file into another.
- Create a new file and add some text to it.
- Display the contents of an existing file.

```
# include <stdio.h>
int main( )
{
    char ch, str[ 10 ] ;
    while ( ( ch = getc ( stdin ) ) != -1 )
        putc ( ch, stdout ) ;
    return 0 ;
}
```

Answer:

```
# include <stdio.h>
int main( )
{
```

```
    char ch, str[ 10 ], tar[ 10 ] ;

    /* fp1: pointer to the source file */
    /* ft: pointer to the target file */
    FILE *fp1, *ft ;

    /* copy the content of one file into another */
    while ( ( ch = getc ( fp1 ) ) != EOF )
        putc ( ch, ft ) ;

    /* creat a new file and add some text to it */
    while ( ( ch = getc ( stdin ) ) != EOF )
        putc ( ch, ft ) ;

    /* display the content of the existing file */
    while ( ( ch = getc ( fp1 ) ) != EOF )
        putc ( ch, stdout ) ;

    return 0 ;
}
```

(b) State True or False:

1. We can send arguments at command line even if we define **main( )** function without parameters.

   Answer: False

2. To use standard file pointers we don't need to open the file using **fopen( )**.

   Answer: True

3. The zeroth element of the **argv** array is always the name of the executable file.

   Answer: False

(c) Point out the errors, if any, in the following program:

```
# include <stdio.h>
int main ( int  ac, char  ( * ) av[  ] )
{
    printf ( "%d\n", ac ) ;
    printf ( "%s\n", av[ 0 ] ) ;
    return 0 ;
}
```

Declaration of **av** should be **char *av[ ]**.

[B] Attempt the following:

(a) Write a program using command line arguments to search for a word in a file and replace it with the specified word. The usage of the program is shown below.

```
C> change <old word> <new word> <filename>
```

Program:

```
# include <stdio.h>
# include <string.h>

FILE *fs, *ft ;
int newstr ( char  *, char  *, char * ) ;

int main ( int argc, char *argv[ ] )
{
    char  str[ 80 ] ;
    char *arg1, *arg2, *fname  ;
    int newstr ( char  *s, char  *t, char *n ) ;
    int c = 0 ;

    /* check if proper no of arguments are passed */
    if ( argc != 4 )
    {
        puts ( "Improper number of arguments\n" ) ;
        exit ( 1 ) ;
```

```
    }

    fs = fopen ( argv[ 3 ], "r+" ) ;
    ft = fopen ( "temp.c", "w" ) ;

    if ( fs == NULL )
    {
        puts ( "Unable to open the file\n" ) ;
        exit ( 2 ) ;
    }

    if ( strlen ( argv[ 1 ] ) > strlen ( argv[ 2 ] ) )
    {
        printf ( "Number of characters in strings mismatch\n" ) ;
        exit ( 3 ) ;
    }

    arg1 = argv[ 1 ] ;
    arg2 = argv[ 2 ] ;
    fname = argv[ 3 ] ;

    while ( fgets ( str, 79, fs ) != NULL )
        c = newstr ( str, arg1, arg2 ) ;

    fclose ( fs ) ;
    fclose ( ft ) ;
    printf ( "%d replacements done\n", c ) ;
    remove ( fname ) ;
    rename ( "temp.c", fname ) ;

    return 0 ;
}

int newstr ( char *pt, char *t, char *n )
{
    static int count = 0 ;
    char *p, *temp, *news, *p1 ;
```

```
    /* find the entered string in our array */
    p1 = pt ;
    do
    {
        p = strstr ( p1, t ) ;

        if ( p == NULL )
            break ;

        /* copy the remaining string */
        news = p + strlen ( t ) ;
        strcpy ( temp, news ) ;

        /* replace the old string */
        strcpy ( p, n ) ;

        /* finally append the remaining part */
        strcat ( p, temp ) ;
        count++ ;

        p1 = p1 + strlen ( t ) ;
    } while ( 1 ) ;

    fputs ( pt, ft ) ;
    return count ;
}
```

(b) Write a program that can be used at command prompt as a calculating utility. The usage of the program is shown below.

C> calc <switch> <n> <m>

Where, **n** and **m** are two integer operands. **switch** can be any one of the arithmetic or comparison operators. If arithmetic operator is supplied, the output should be the result of the operation. If comparison operator is supplied then the output should be **True** or **False**.

Program:

```
/* To perform the given arithmetic operation on the two integers */
# include <stdio.h>
# include <string.h>

int main ( int argc, char *argv[ ] )
{
    int  i, first, second, result ;

    /* arithmetic and logical operators */
    char *str[7] = { "+", "-", "*", "/", "%", "&&", "||" } ;

    /* check if proper no of arguments are passed */
    if ( argc != 4 )
    {
        puts ( "Improper number of arguments\n" ) ;
        exit ( 1 ) ;
    }

    /* check if the entered operator is valid */
    for ( i = 0 ; i <= 6 ; i++ )
    {
        if ( strcmp ( argv[ 2 ], str[ i ] ) == 0 )
            break ;
    }

    if ( i == 7 )
    {
        printf ( "\nNot a valid operator\n" ) ;
        exit ( 2 ) ;
    }

    first = atoi ( argv[ 1 ] ) ;
    second = atoi ( argv[ 3 ] ) ;
    printf ( "\nResult of the operation is:\n" ) ;

    switch ( i )
```

```
{
    case 0 :
        result = first + second ;
        printf ( "%d\n", result ) ;
        break ;
    case 1 :
        result = first - second ;
        printf ( "%d\n", result ) ;
        break ;
    case 2 :
        result = first * second ;
        printf ( "%d\n", result ) ;
        break ;
    case 3 :
        result = first / second ;
        printf ( "%d\n", result ) ;
        break ;
    case 4 :
        result = first % second ;
        printf ( "%d\n", result ) ;
        break ;
    case 5 :
        result = first && second ;
        result == 0 ? printf ( "False\n" ) : printf ( "True\n" ) ;
        break ;
    case 6 :
        result = first || second ;
        result == 0 ? printf ( "False\n" ) : printf ( "True\n" ) ;
        break ;
    }

    return 0 ;
}
```