

1- Write a program in C to encrypt a text file.

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

void main()
{
    char fname[20], ch;
    FILE *fpts, *fptt;

    printf("\n\n Encrypt a text file :\n");
    printf("-----\n");

    printf(" Input the name of file to encrypt : ");
    scanf("%s", fname);

    fpts=fopen(fname, "r");
    if(fpts==NULL)
    {
        printf(" File does not exists or error in opening..!!");
        exit(1);
    }
    fptt=fopen("temp.txt", "w");
    if(fptt==NULL)
    {
        printf(" Error in creation of file temp.txt ..!!");
        fclose(fpts);
        exit(2);
    }
    while(1)
    {
        ch=fgetc(fpts);
        if(ch==EOF)
        {
            break;
        }
        else
        {
            ch=ch+100;
            fputc(ch, fptt);
        }
    }
    fclose(fpts);
    fclose(fptt);
    fpts=fopen(fname, "w");
    if(fpts==NULL)
    {
        printf(" File does not exists or error in opening..!!");
        exit(3);
    }
    fptt=fopen("temp.txt", "r");
    if(fptt==NULL)
    {
        printf(" File does not exists or error in opening..!!");
        fclose(fpts);
        exit(4);
    }
}
```

```

while(1)
{
    ch=fgetc(fptt);
    if(ch==EOF)
    {
        break;
    }
    else
    {
        fputc(ch, fpts);
    }
}
printf(" File %s successfully encrypted ..!!\n\n", fname);
fclose(fpts);
fclose(fptt);
}

```

Sample Output:

Encrypt a text file :

Input the name of file to encrypt : test.txt
File test.txt successfully encrypted ..!!

If you read the file test.txt you will see the following :

?????H?S?????b??n

2- Write a program in C to decrypt a previously encrypted file.

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

void main()
{
    char ch, fname[20];
    FILE *fpts, *fptt;

    printf("\n\n Decrypt a text file :\n");
    printf("-----\n");

    printf(" Input the name of file to decrypt : ");
    scanf("%s", fname);

    fpts=fopen(fname, "w");
    if(fpts==NULL)
    {
        printf(" File does not exists or error in opening..!!");
        exit(7);
    }
    fptt=fopen("temp.txt", "r");
    if(fptt==NULL)
    {
        printf(" File does not exists or error in opening..!!");
        fclose(fpts);
        exit(9);
    }
    while(1)
    {
        ch=fgetc(fptt);
        if(ch==EOF)
        {
            break;
        }
        else
        {
            ch=ch-100;
            fputc(ch, fpts);
        }
    }
    printf(" The file %s decrypted successfully..!!\n\n", fname);
    fclose(fpts);
    fclose(fptt);
}
```

Sample Output:

```
Decrypt a text file :
-----
Input the name of file to decrypt : test.txt
The file test.txt decrypted successfully..!!
```

- 3- Write a program that reverses the order of the bits in an unsigned integer value. The program should input the value from the user and call function reverse Bits to print the bits in reverse order. Print the value in bits both before and after the bits are reversed to confirm that the bits are reversed properly.

```
/* Exercise 10.17 Solution */
#include<stdio.h>

/* prototypes */
unsigned reverseBits( unsignedvalue);
void displayBits( unsignedvalue);

int main() {
    unsigned a; /* unsigned integer from user */

    /* prompt user and read value */
    printf("Enter an unsigned integer: ");
    scanf("%u", &a);

    /* display bits of a before reversed */
    printf("\nBefore bits are reversed:\n");
    displayBits(a);

    /* reverse bits and display results */
    a = reverseBits(a);
    printf("\nAfter bits are reversed:\n");
    displayBits(a);

    return 0; /* indicate successful termination */
} /* end main */

/* reverseBits reverses the bits of value */
unsigned reverseBits(unsigned value) {
    unsigned mask = 1; /* bit mask */
    unsigned temp = 0; /* reversed bits */
    int i; /* loop counter */

    /* loop through bits of value */
    for (i = 0; i <= 15; i++) {
        temp <<= 1; /* right shift 1 bit */
        temp |= (value & mask); /* separate bit and place in temp */
        value >>= 1; /* left shift 1 bit */
    } /* end for */

    return temp;
} /* end function reverseBits */
```

```

/* display the bits of value */
void displayBits(unsigned value) {
    unsigned c; /* bit counter */
    unsigned displayMask = 1 << 15; /* bit mask */

    printf("%7u = ", value);

    /* loop through bits */
    for (c = 1; c <= 16; c++) {
        value & displayMask ? putchar('1') : putchar('0');
        value <<= 1; /* shift value 1 bit to the left */

        if (c % 8 == 0) { /* print a space */
            putchar(' ');
        } /* end if */

    } /* end for */

    putchar('\n');
} /* end function displayBits */
}

```

Enter an unsigned integer: 2127

Before bits are reversed:

2127 = 00001000 01001111

After bits are reversed:

61968 = 11110010 00010000