

```

/* file_basic.c */
#include <stdio.h>

int main(void)
{
    int state;

    /* 파일의 개방 */
    FILE * file = fopen("Test.txt", "wt");
    if(file==NULL){
        printf("file open error!\n");
        return 1;
    }

    /* 파일의 종결 */
    state=fclose(file);
    if(state!=0){
        printf("file close error!\n");
        return 1;
    }

    return 0;
}

```

파일 접근 모드 +	데이터 입,출력 모드	= 파일 개방 모드
r	b	"rb"
w	t	"wt"
a	t	"at"
r	t	"rt"
...	...	...

```
/* fputs_puts.c */
#include <stdio.h>

int main(void)
{
    int state;

    /* 파일의 개방 */
    FILE * file = fopen("Test.txt", "wt");
    if(file==NULL){
        printf("file open error!\n");
        return 1;
    }

    /* puts 함수의 활용 */
    puts("Don't Worry!");

    /* fputs 함수의 활용 */
    fputs("Don't Worry!\n", stdout);

    /* fputs 함수의 활용 */
    fputs("Don't Worry!\n", file);

    /* 파일의 종결 */
    state=fclose(file);
    if(state!=0){
        printf("file close error!\n");
        return 1;
    }

    return 0;
}
```

```

/* fgets.c */
#include <stdio.h>

int main(void)
{
    int state;
    char buf[30];

    /* 파일의 개방 */
    FILE * file = fopen("Test.txt", "rt");
    if(file==NULL){
        printf("file open error!\n");
        return 1;
    }

    /* fgets 함수의 활용 */
    fputs("데이터 입력 : ", stdout);
    fgets(buf, sizeof(buf), stdin);
    puts(buf);

    /* fgets 함수의 활용 */
    fgets(buf, sizeof(buf), file);
    puts(buf);

    /* 파일의 종결 */
    state=fclose(file);
    if(state!=0){
        printf("file close error!\n");
        return 1;
    }

    return 0;
}

```

```

/* formatted_io.c */
#include <stdio.h>

int main(void)
{
    int state;          // error 체크 변수.
    int i, j;          // for 루프 변수.
    int a=0, b=0, c=0; // fscanf 변수.
    char c1=0, c2=0;  // fscanf 변수.

    /* 쓰기 전용 파일 개방 */
    FILE * file = fopen("Test.txt", "wt");
    if(file==NULL){
        printf("file open error!\n");
        return 1;
    }

    /* fprintf 함수의 활용 */
    for(i=2; i<10; i++)
        for(j=1; j<10; j++)
            fprintf(file, "%d X %d = %d \n", i, j, i*j);

    /* 파일의 종결 */
    state=fclose(file);
    if(state!=0){
        printf("file close error!\n");
        return 1;
    }

    /* 읽기 전용 파일 개방 */
    file = fopen("Test.txt", "rt");
    if(file==NULL){
        printf("file open error!\n");
        return 1;
    }
}

```

```
/* fscanf 함수의 활용 */
for(i=2; i<10; i++)
    for(j=1; j<10; j++)
    {
        fscanf(file, "%d %c %d %c %d", &a, &c1, &b, &c2, &c);

        printf("%d %c %d %c %d \n", a, c1, b, c2, c);
    }

/* 파일의 종결 */
state=fclose(file);
if(state!=0){
    printf("file close error!\n");
    return 1;
}

return 0;
}
```

```

/* eof_pro3.c */
#include <stdio.h>

int main(void)
{
    int state;           // error 체크 변수.
    char ch;

    /* 읽기 전용 파일 개방 */
    FILE* file = fopen("Test.txt", "wb");
    if(file==NULL){
        printf("file open error!\n");
        return 1;
    }

    fputc('1', file);
    fputc((char)255, file); // char 형으로 형 변환
    fputc('2', file);

    /* 파일의 종결 */
    state=fclose(file);
    if(state!=0){
        printf("file close error!\n");
        return 1;
    }

    /* 읽기 전용 파일 개방 */
    file = fopen("Test.txt", "rb");
    if(file==NULL){
        printf("file open error!\n");
        return 1;
    }

    while(1)
    {
        ch=fgetc(file);
    }
}

```

```
        if(feof(file)!=0)
            break;
        printf("data : %c \n", ch);
    }

    /* 파일의 종결 */
    state=fclose(file);
    if(state!=0){
        printf("file close error!\n");
        return 1;
    }

    return 0;
}
```

```
/* ProbArray.c */
```

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

```
void function(int);
```

```
int main (void)
```

```
{
```

```
    int m;
```

```
    int arr[m];
```

```
    scanf("%d", &m);
```

```
    arr[0] = 1;
```

```
    return 0;
```

```
}
```

```
/* ProbArray2.c */
```

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

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

```
int main (void)
```

```
{
```

```
    int m;
```

```
    int *array;
```

```
    fputs("배열의 크기를 입력하세요 :", stdout);
```

```
    scanf("%d", &m);
```

```
    array = (int*) malloc(sizeof(int) * m);
```

```
    array[0] = 1;
```

```
    return 0;
```

```
}
```