fopen modes

```
r  - open for reading
w  - open for writing (file need not exist)
a  - open for appending (file need not exist)
r+ - open for reading and writing, start at beginning
w+ - open for reading and writing (overwrite file)
a+ - open for reading and writing (append if file exists)
```

Reading and writing with fprintf, fscanf fputc, and fgetc

```
fscanf(fp, "%d%d", &x, &y);
fprintf(fp, "Testing...\n");
```

```
int fgetc (FILE *fp);
int fputc( int c, FILE *fp );
```

Binary file I/O - fread and fwrite, fseek

```
size_t fread(void *ptr, size_t elements_size, elements_numbers, FILE *fp);
size_t fwrite(const void *ptr, , size_t elements_size, elements_numbers, FILE *fp);

int fseek(FILE *fp, int offset, int whence);
```

Constant	Description
SEEK_SET	Beginning of file
SEEK_CUR	Current position of the file pointer
SEEK_END	End of file

Example 1

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

typedef struct {
    int id;
    int px, py;
} MyData;

int main()
{
    int i;
    FILE *fp1, *fp2;
    MyData md[10];
    fp1 = fopen("mydata.txt", "w");
    fp2 = fopen("mydata.dat", "wb");
    for (i=0; i<10; i++) {
        md[i].id=i; md[i].px=rand(); md[i].py=rand();
        fprintf(fp1, "%d %d %d\n", md[i].id, md[i].px, md[i].py);
    }
    fwrite(md, sizeof(MyData), 10, fp2);
    fclose(fp1);
    fclose(fp2);
    return 0;
}
```

Example 2

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

typedef struct {
    int id;
    int px, py;
} MyData;

int main()
{
    int i;
    FILE *fp1, *fp2;
    MyData md1[10], md2[10];
    fp1 = fopen("mydata.txt", "r");
    fp2 = fopen("mydata.dat", "rb");
    for (i=0; i<10; i++) {
        fscanf(fp1, "%d %d %d", &md1[i].id, &md1[i].px, &md1[i].py);
    }
    fread(md2, sizeof(MyData), 10, fp2);

    for (i=0; i<10; i++) {
        printf("md1[%d]=(%2d, %10d, %10d)\n", i, md1[i].id, md1[i].px, md1[i].py);
        printf("md2[%d]=(%2d, %10d, %10d)\n", i, md2[i].id, md2[i].px, md2[i].py);
    }

    fclose(fp1);
    fclose(fp2);
    return 0;
}
```

EXERCISE:

1. Instead of reading all 10 elements once, you may read 1 element each time with a loop.
2. Let user input the ID, output the record. (Use fseek to find the correct position.)