



date.h

```
#include <stdio.h>

struct Date_t {
    int day;
    char mon[4];
    int year;
};

typedef struct Date_t Date;

typedef enum { Success, Fail } Result;

/* Reads a date from the open file in,
   expects dt to be allocated */
Result DateRead(FILE *in, Date *dt);

/* Prints the given date the the given
   open stream */
Result DatePrint(FILE *out, Date dt);

/* Returns the month's number of dt,
   -1 in case of error */
int DateMonth(Date dt);

/* Difference in days between the two
   given dates. -1 in case of error */
int DateDiff(Date a, Date b);
```

main.c

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

int main()
{
    Date d1, d2;

    if(DateRead(stdin, &d1)==Fail ||
       DateRead(stdin, &d2)==Fail) return 1;

    printf("The month of ");
    DatePrint(stdout, d1);
    printf(" is %d\n", DateMonth(d1));

    printf("The diff between ");
    DatePrint(stdout, d1);
    printf(" and ");
    DatePrint(stdout, d2);
    printf(" is %d days\n", DateDiff(d1, d2));

    return 0;
}
```

date.c

```
#include <string.h>
#include "date.h"

static const char *Month[] = {
    "JAN", "FEB", "MAR", "APR", "MAY", "JUN",
    "JUL", "AUG", "SEP", "OCT", "NOV", "DEC"
};

Result DateRead(FILE *in, Date *dt)
{
    if(in==NULL || dt==NULL) return Fail;

    if(fscanf(in, "%d %s %d",
              &dt->day, dt->mon, &dt->year)!=3)
        return Fail;

    if(dt->day<1 || dt->day>31) return Fail;
    if(DateMonth(*dt)==-1) return Fail;
    return Success;
}

Result DatePrint(FILE *out, Date dt)
{
    if(out == NULL) return Fail;

    if(fprintf(out, "%d %s %d",
              dt.day, dt.mon, dt.year)<0)
        return Fail;

    return Success;
}

int DateMonth(Date dt)
{
    int i;
    for(i=0; i<12; i++) {
        if(strcmp(dt.mon, Month[i])==0)
            return i+1;
    }
    return -1;
}

int DateDiff(Date a, Date b)
{
    int mon_a, mon_b, days_a, days_b;

    mon_a = DateMonth(a);
    if(mon_a == -1) return -1;
    mon_b = DateMonth(b);
    if(mon_b == -1) return -1;

    days_a = a.day + 30*mon_a + 365*a.year;
    days_b = b.day + 30*mon_b + 365*b.year;

    return days_a-days_b;
}
```

date_adt.h

```
#ifndef DATE_H
#define DATE_H

#include <stdio.h>

typedef struct Date_t *Date;

typedef enum { Success, Fail } Result;

/* Allocates a new date, NULL on error */
Date DateCreate(int day, int mon, int year);

/* Reads a date from the open file in,
 expects dt to be allocated */
Result DateRead(Date dt, FILE *in);

/* Prints the given date the the given
 open stream */
Result DatePrint(Date dt, FILE *out);

/* Returns the month's number of dt,
 -1 in case of error */
int DateMonth(Date dt);

/* Difference in days between the two
 given dates. */
Result DateDiff(Date a, Date b, int *diff);

/* Frees up the space of dt */
void DateDestroy(Date dt);

#endif
```

date_adt.c

```
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include "date_adt.h"

struct Date_t {
    int day;
    char mon[4];
    int year;
};

static const char *Month[] = {
    "JAN", "FEB", "MAR", "APR", "MAY", "JUN",
    "JUL", "AUG", "SEP", "OCT", "NOV", "DEC" };

Date DateCreate(int day, int mon, int year)
{
    Date dt = NULL;

    if(day<1 || day>31 || mon<1 || mon>12)
        return NULL;

    dt = malloc(sizeof(struct Date_t));
    if(dt == NULL) return NULL;

    dt->day = day;
    strcpy(dt->mon, Month[mon-1]);
    dt->year = year;

    return dt;
}

Result DateRead(Date dt, FILE *in)
{
    if(in==NULL || dt==NULL) return Fail;

    if(fscanf(in, "%d %s %d",
        &dt->day, dt->mon, &dt->year)!=3)
        return Fail;

    if(dt->day<1 || dt->day>31) return Fail;
    if(DateMonth(dt)==-1) return Fail;
    return Success;
}

Result DatePrint(Date dt, FILE *out)
{
    if(out==NULL || dt==NULL) return Fail;

    if(fprintf(out, "%d %s %d",
        dt->day, dt->mon, dt->year)<0)
        return Fail;

    return Success;
}

int DateMonth(Date dt)
{
    int i=0;

    if(dt==NULL) return -1;

    for(i=0; i<12; i++) {
        if(strcmp(dt->mon, Month[i])==0)
            return i+1;
    }

    return -1;
}

Result DateDiff(Date a, Date b, int *diff)
{
    int mon_a, mon_b, days_a, days_b;

    if(a==NULL || b==NULL || diff==NULL)
        return Fail;

    mon_a = DateMonth(a);
    if(mon_a == -1) return Fail;
    mon_b = DateMonth(b);
    if(mon_b == -1) return Fail;

    days_a = a->day + 30*mon_a + 365*a->year;
    days_b = b->day + 30*mon_b + 365*b->year;

    *diff = days_a-days_b;
    return Success;
}

void DateDestroy(Date dt)
{
    if(dt==NULL) return;
    free(dt);
}
```

```
Result DatePrint(Date dt, FILE *out)
{
    if(out==NULL || dt==NULL) return Fail;

    if(fprintf(out, "%d %s %d",
        dt->day, dt->mon, dt->year)<0)
        return Fail;

    return Success;
}

int DateMonth(Date dt)
{
    int i=0;

    if(dt==NULL) return -1;

    for(i=0; i<12; i++) {
        if(strcmp(dt->mon, Month[i])==0)
            return i+1;
    }

    return -1;
}

Result DateDiff(Date a, Date b, int *diff)
{
    int mon_a, mon_b, days_a, days_b;

    if(a==NULL || b==NULL || diff==NULL)
        return Fail;

    mon_a = DateMonth(a);
    if(mon_a == -1) return Fail;
    mon_b = DateMonth(b);
    if(mon_b == -1) return Fail;

    days_a = a->day + 30*mon_a + 365*a->year;
    days_b = b->day + 30*mon_b + 365*b->year;

    *diff = days_a-days_b;
    return Success;
}

void DateDestroy(Date dt)
{
    if(dt==NULL) return;
    free(dt);
}
```

main_adt.c

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

int main()
{
    Date d1, d2;
    int diff;

    d1 = DateCreate(1, 1, 1);
    if(d1==NULL) return 1;
    d2 = DateCreate(1, 1, 1);
    if(d2==NULL) return 1;

    if(DateRead(d1, stdin)==Fail ||
        DateRead(d2, stdin)==Fail) return 1;

    printf("The month of ");
    DatePrint(d1, stdout);
    printf(" is %d\n", DateMonth(d1));

    if(DateDiff(d1, d2, &diff)==Success) {
        printf("The diff between ");
        DatePrint(d1, stdout);
        printf(" and ");
        DatePrint(d2, stdout);
        printf(" is %d days\n", diff);
    }

    DateDestroy(d1);
    DateDestroy(d2);

    return 0;
}
```