숙제 2

리스트 연산 (배열)

순서

1. 프로그램 구조
2. Create 처리하기
3. Load 처리하기
4. Insert 처리하기
5. Delete 처리하기
6. Search 처리하기
7. Print 처리하기

0. 자료 구조

```
typedef struct _sinfo {
        char name[8];
        char sex;
        char city[8];
        char dept[16];
        float gpa;
        int height;
        int weight;
} sinfo;

int cnt = 0;
int n;
sinfo *slist;
```

1. 프로그램 구조 (1)

```
int main ( )
{
   FILE *fp = fopen ( "input.txt", "r+t");
   char input[512];
   char tok1[32], tok2[32], tok3[32], tok4[32], tok5[32], tok6[32],
        tok7[32], tok8[32], tok9[32];
```

1. 프로그램 구조 (2)

```
while ( fgets ( input, 512, fp ) != NULL ) {
   sscanf(input, "%s%s%s%s%s%s%s%s%s", tok1, tok2, tok3, tok4,
                    tok5, tok6, tok7, tok8, tok9);
   if ( strcmp ( tok1, "CREATE" ) == 0)
            process_create ( );
   else if ( strcmp (tok1, "LOAD") == 0 )
            process_load ( tok2 );
   else if ( strcmp(tok1, "PRINT") == 0)
            process_print ( );
   else if (strcmp(tok1, "INSERT") == 0)
            process_insert ( tok2, tok3, tok4, tok5, tok6,
                             tok7, tok8 );}
   else if (strcmp(tok1, "DELETE") == 0)
            process_delete ( tok2 );
   else if ( strcmp (tok1, "SEARCH") == 0 ) {
            process_search ( tok2 );
   else
      printf("%s is not a keyword.\n", tok1);
}
```

1. 프로그램 구조 (3)

```
    fclose ( fp );
    return 0;
}
```

2. Create 처리하기

```
void process_create ( )
{
        n = 100;
        slist = (sinfo *) calloc ( n, sizeof(sinfo) );
}
```

3. Load 처리하기

```
void process_load ( char *fn )
{
        FILE *fp2 = fopen ( fn, "r+t" );
        sinfo tinfo;
        char str[512];
        while ( fgets ( str, 512, fp2 ) != NULL ) {
                sscanf ( str, "%s %c %s %s %f %d %d", tinfo.name,
                                &tinfo.sex, tinfo.city, tinfo.dept,
                                &tinfo.gpa, &tinfo.height, &tinfo.weight);
                process_insert ( tinfo );
        }

        fclose ( fp2 );
}
```

4. Insert 처리하기 (1)

```
void process_insert ( sinfo tinfo )
{
        if ( cnt == 0 ) {
                slist[cnt++] = tinfo;
                return;
        }

        int i, j;
        for ( i = 0; i < cnt; i++ ) {
                if ( strcmp ( tinfo.name, slist[i].name ) <= 0 )
                        break;
        }

        for ( j = cnt-1; j >= i; j-- )
                slist[j+1] = slist[j];

        slist[i] = tinfo;
        cnt++;
}
```

4. Insert 처리하기 (2)

```
void process_insert ( char *tok2, char *tok3, char *tok4, char *tok5,
                      char *tok6, char *tok7, char *tok8 )
{
        sinfo tinfo;

        strcpy ( tinfo.name, tok2 );
        tinfo.sex = tok3[0];
        strcpy ( tinfo.city, tok4 );
        strcpy ( tinfo.dept, tok5 );
        tinfo.gpa = atof ( tok6 );
        tinfo.height = atoi ( tok7 );
        tinfo.weight = atoi ( tok8 );

        process_insert ( tinfo );
}
```

5. Delete 처리하기

```
void process_delete ( char *tok2 )
{
        int i, j;
        for ( i = 0; i < cnt; i++ ) {
                if ( strcmp ( slist[i].name, tok2 ) == 0 )
                        break;
        }

        if ( i == cnt ) {
                printf("No %s in the list.\n", tok2 );
                return;
        }

        for ( j = i; j < cnt-1; j++ )
                slist[j] = slist[j+1];
        cnt--;

        printf("%s is deleted.\n", tok2);
}
```

6. Search 처리하기

```
void process_search ( char *tok2 )
{
        int i;

        for ( i = 0; i < cnt; i++ ) {
                if ( strcmp ( slist[i].name, tok2) == 0 ) {
                        printf("%s found: %s %c %s %s %f %d %d\n", tok2,
                                        slist[i].name, slist[i].sex,
                                        slist[i].city, slist[i].dept,
                                        slist[i].gpa, slist[i].height,
                                        slist[i].weight);
                        return;
                }
        }

        printf("%s not found.\n", tok2);
}
```

7. Print 처리하기

```
void process_print ( )
{
        int i;

        for ( i = 0; i < cnt; i++ )
                printf("%s %c %s %s %f %d %d\n", slist[i].name,
                                slist[i].sex, slist[i].city,
                                slist[i].dept, slist[i].gpa,
                                slist[i].height, slist[i].weight);
}
```