

<전체 소스 코드>

```
// 2019100991 이선아

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

typedef struct {
    char name[20];
    char callNum[40];
    char gender[10];
    int age;
} Friend;

typedef struct {
    int count; // 총 친구 수
    int max; // 최고령
    int min; // 최저령
    int mCount; // 남성 수
    int fCount; // 여성 수
} Info;

void initInfo(Info*);
void printMenu();
int checkSize(Friend*, Info*, int);
void inputFriend(Friend*, Info*);
void calculate(Friend*, Info*);
void friendInfo(Friend*, Info*);

int main() {
    int defaultSize = 50; // 기본값
    Friend *friend = malloc(sizeof(Friend)*defaultSize);
    Info* info = malloc(sizeof(Info));
    int num = 0;
    int searchFlag;
    int delFlag;
    char findName[20];
    char delName[20];

    info->count = 0;
    initInfo(info);

    while(1) {
        printMenu();

        scanf("%d", &num);

        switch(num) {
            case 1 :
                printf("1번 개별 신상 정보 입력을 선택하셨습니다.\n");
```

```

inputFriend(friend, info);
defaultSize = checkSize(friend, info, defaultSize);

break;

case 2 :
    printf("2번 전체 신상 정보 출력을 선택하셨습니다.Wn");
    for(int i = 0; i < info->count; i++) {
        printf("이름 : %s, 휴대폰 번호 : %s, 성별 : %s, 나이 : %dWn",
friend[i].name, friend[i].callNum, friend[i].gender, friend[i].age);
    }

    friendInfo(friend, info);

    break;

case 3 :
    searchFlag = 1;
    printf("3번 개별 신상 정보 조회를 선택하셨습니다.Wn");
    printf("조회할 이름 입력 : ");
    scanf("%s", findName);

    for(int i = 0; i < info->count; i++) {
        if (!strcmp(findName, friend[i].name)) {
            printf("이름 : %s, 휴대폰 번호 : %s, 성별 : %s, 나이 : %dWn",
friend[i].name, friend[i].callNum, friend[i].gender, friend[i].age);
            searchFlag = 0; // 일치하는 이름이 있다면 플래그를 0으로 설정.
        }
    }
    if(searchFlag) // = (searchFlag = 1)
        printf("[%s]을/를 조회할 수 없습니다.Wn", findName);

    break;

case 4 :
    delFlag = 1;
    printf("4번 개별 신상 정보 삭제를 선택하셨습니다.Wn");
    printf("삭제할 이름 입력 : ");
    scanf("%s", delName);

    for(int i = 0; i < info->count; i++) {
        if (!strcmp(delName, friend[i].name)) {
            delFlag = 0; // 일치하는 이름이 있다면 플래그를 0으로 설정.
            for (int j = i; j < info->count - 1; j++) {
                friend[j] = friend[j + 1];
            }
            info->count--;
            i--;

            printf("[%s] 학생의 정보를 삭제했습니다.Wn", delName);
        }
    }
}

```

```

        if(delFlag) // = (delFlag = 1)
            printf("[%s] 학생의 정보가 없습니다.\n", delName);

        break;

    case 5 :
        printf("프로그램을 종료합니다.\n");

        free(friend);
        free(info);

        return 0;
    }
}

// 메뉴 보기
void printMenu() {
    printf("-----\n");
    printf("[메뉴]\n1)개별 신상 정보 입력\n2)전체 신상 정보 출력\n3)개별 신상 정보 조회\n4)개별 신상 정보 삭제\n5)프로그램 종료\n");
    printf("-----\n");
    printf("원하시는 메뉴를 선택해주세요 ");
}

// 동적 할당
int checkSize(Friend* friend, Info* info, int size) {
    if (info->count == size) {
        size *= 2; // 기본값 50
        realloc(friend, sizeof(Friend) * size);
    }
    return size;
}

// 계산을 위한 초기값
void initInfo(Info* info) {
    info->max = 0;
    if (info->count == 0)
        info->min = 0;
    else
        info->min = 99999;
    info->mCount = 0;
    info->fCount = 0;
}

// 메뉴 1번 개별 신상 정보 입력
void inputFriend(Friend* friend, Info* info) {
    printf("영문 이름을 입력해 주세요 : ");
    scanf("%s", friend[info->count].name);
    printf("휴대폰 번호를 입력해 주세요 (- 필수 입력) : ");
    scanf("%s", friend[info->count].callNum);
}

```

```

printf("성별을 입력해주세요 (남자는 'M', 여자는 'F') : ");
scanf("%s", friend[info->count].gender);
printf("나이를 입력해주세요 (숫자만 입력) : ");
scanf("%d", &friend[info->count].age);
printf("입력이 완료되었습니다.\n");

info->count++; // 총 친구 수++
}

// 신상 정보 계산기
void calculate(Friend* friend, Info* info) {
    initInfo(info);

    for (int i = 0; i < info->count; i++) {
        if (friend[i].age > info->max)
            info->max = friend[i].age;
        if (friend[i].age < info->min)
            info->min = friend[i].age;
        if (strcmp(friend[i].gender, "M") == 0)
            info->mCount++;
        else if (strcmp(friend[i].gender, "F") == 0)
            info->fCount++;
    }
}

// 메뉴 2번 전체 신상 정보 출력
void friendInfo(Friend* friend, Info* info) {
    calculate(friend, info);

    printf("총 %d명\n", info->count);
    printf("남자 : %d명\n", info->mCount);
    printf("여자 : %d명\n", info->fCount);
    printf("최고령 : %d세\n", info->max);
    printf("최저령 : %d세\n", info->min);
}

```

## <주요 실행 결과 이미지>

### 1) 메뉴 1번 실행 (친구 3명 추가(동일 이름 포함))

```
[seona@seona:~$ ./salee
```

```
-----  
[메뉴]
```

- 1)개 별 신 상 정 보 입 력
- 2)전 체 신 상 정 보 출 력
- 3)개 별 신 상 정 보 조 회
- 4)개 별 신 상 정 보 삭 제
- 5)프 로 그 램 종 료

```
-----  
[원하시는 메뉴를 선택해주세요 1
```

```
1번 개별 신상 정보 입력을 선택하셨습니다.
```

```
[영문 이름을 입력해주세요 : seona
```

```
[휴대폰 번호를 입력해주세요 (- 필수 입력) : 010-3800-7129
```

```
[성별을 입력해주세요 (남자는 'M', 여자는 'F') : F
```

```
[나이를 입력해주세요 (숫자만 입력) : 23
```

```
입력이 완료되었습니다.
```

```
-----  
[메뉴]
```

- 1)개 별 신 상 정 보 입 력
- 2)전 체 신 상 정 보 출 력
- 3)개 별 신 상 정 보 조 회
- 4)개 별 신 상 정 보 삭 제
- 5)프 로 그 램 종 료

```
-----  
원하시는 메뉴를 선택해주세요 █
```

```
[원하시는 메뉴를 선택해주세요 1
```

```
1번 개별 신상 정보 입력을 선택하셨습니다.
```

```
[영문 이름을 입력해주세요 : seungcheol
```

```
[휴대폰 번호를 입력해주세요 (- 필수 입력) : 010-1234-5678
```

```
[성별을 입력해주세요 (남자는 'M', 여자는 'F') : M
```

```
[나이를 입력해주세요 (숫자만 입력) : 25
```

```
입력이 완료되었습니다.
```

```
-----  
[메뉴]
```

- 1)개 별 신 상 정 보 입 력
- 2)전 체 신 상 정 보 출 력
- 3)개 별 신 상 정 보 조 회
- 4)개 별 신 상 정 보 삭 제
- 5)프 로 그 램 종 료

```
-----  
원하시는 메뉴를 선택해주세요 █
```

```
[원하시는 메뉴를 선택해주세요 1
```

```
1번 개별 신상 정보 입력을 선택하셨습니다.
```

```
[영문 이름을 입력해주세요 : seona
```

```
[휴대폰 번호를 입력해주세요 (- 필수 입력) : 010-0000-0000
```

```
[성별을 입력해주세요 (남자는 'M', 여자는 'F') : F
```

```
[나이를 입력해주세요 (숫자만 입력) : 20
```

```
입력이 완료되었습니다.
```

## 2) 메뉴 2번 실행(현재 상태)

[원하시는 메뉴를 선택해주세요 2

2번 전체 신상 정보 출력을 선택하셨습니다 .

이름 : seona, 휴대폰 번호 : 010-3800-7129, 성별 : F, 나이 : 23

이름 : seungcheol, 휴대폰 번호 : 010-1234-5678, 성별 : M, 나이 : 25

이름 : seona, 휴대폰 번호 : 010-0000-0000, 성별 : F, 나이 : 20

총 3명

남자 : 1명

여자 : 2명

최고령 : 25세

최저령 : 20세

## 3) 메뉴 3번 실행 (동일 이름 조회)

[원하시는 메뉴를 선택해주세요 3

3번 개별 신상 정보 조회를 선택하셨습니다 .

[조회할 이름 입력 : seona

이름 : seona, 휴대폰 번호 : 010-3800-7129, 성별 : F, 나이 : 23

이름 : seona, 휴대폰 번호 : 010-0000-0000, 성별 : F, 나이 : 20

## 4) 메뉴 4번 실행 (동일 이름 2명 삭제)

[원하시는 메뉴를 선택해주세요 4

4번 개별 신상 정보 삭제를 선택하셨습니다 .

[삭제할 이름 입력 : seona

[seona] 학생의 정보를 삭제했습니다 .

[seona] 학생의 정보를 삭제했습니다 .

## 4) 메뉴 4번 재실행 (삭제 후 동일 이름 삭제 케이스)

[원하시는 메뉴를 선택해주세요 4

4번 개별 신상 정보 삭제를 선택하셨습니다 .

[삭제할 이름 입력 : seona

[seona] 학생의 정보가 없습니다 .

## 5) 메뉴 3번 재실행 (삭제한 친구 조회)

[원하시는 메뉴를 선택해주세요 3

3번 개별 신상 정보 조회를 선택하셨습니다 .

[조회할 이름 입력 : seona

[seona]을 /를 조회할 수 없습니다 .

## 6) 메뉴 2번 재실행

[원하시는 메뉴를 선택해주세요 2

2번 전체 신상 정보 출력을 선택하셨습니다 .

이름 : seungcheol, 휴대폰 번호 : 010-1234-5678, 성별 : M, 나이 : 25

총 1명

남자 : 1명

여자 : 0명

최고령 : 25세

최저령 : 25세

## 7) 메뉴 5번 실행

[원하시는 메뉴를 선택해주세요 5

프로그램을 종료합니다 .

seona@seona:~\$ █