

7-13일 최종 보고서

보고서에 그동안 만드는데 실패한 과제들과 새로 주신 과제들을 기재하였습니다.

실패한 과제로는 1일차 4번, 5번 2일차 4번, 6일차 3번, 8일차 4번이 있었고 그중 6일차 3번은 8일차 1번으로 다시 나와 있기 때문에 제외했습니다

1일차 4번 소스 코드

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

int main()
{
    double num1, num2, result = 0;
    char operation;

    printf("Input Arithmetic Operation\n");
    scanf("%lf %c %lf", &num1, &operation, &num2);

    switch (operation)
    {
        case '+':
            result = num1 + num2;
            break;
        case '-':
            result = num1 - num2;
            break;
        case '/':
            if (num2 != 0)
            {
                result = num1 / num2;
            }
            else {
                printf("0으로 나눌 수는 없습니다.\n");
                return 1; // 프로그램을 종료합니다.
            }
            break;
        case '*':
            result = num1 * num2;
            break;
        case '^':
            result = 1;
            for (int i = 0; i < (int)num2; i++)
            {
                result *= num1;
            }
    }
}
```

```

    }
    break;
default:
    printf("잘못된 연산자입니다.\n");
    return 1; // 프로그램을 종료합니다.
}

printf("%lf %c %lf = %lf\n", num1, operation, num2, result);
return 0;

```

실행결과

```

Microsoft Visual Studio 디버그
Input Arithmetic Operation
ex) 3.4 + 8.5
     2.9 - 5.4
     3.9 * 8.0
     3.9 ^ 8

input : 3.4 * 8
3.400000 * 8.000000 = 27.200000

C:\Users\naru4\OneDrive\바탕 화면\ConsoleApplication4\x64\Debug\ConsoleApplication4.exe(프로세스 45200개)이(가) 종료되었습니다(코드: 0개).
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...

```

```

Microsoft Visual Studio 디버그
Input Arithmetic Operation
ex) 3.4 + 8.5
     2.9 - 5.4
     3.9 * 8.0
     3.9 ^ 8

input : 4 ^ 3
4.000000 ^ 3.000000 = 64.000000

C:\Users\naru4\OneDrive\바탕 화면\ConsoleApplication4\x64\Debug\ConsoleApplication4.exe(프로세스 28704개)이(가) 종료되었습니다(코드: 0개).
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...

```

1일차 5번 과제 소스코드

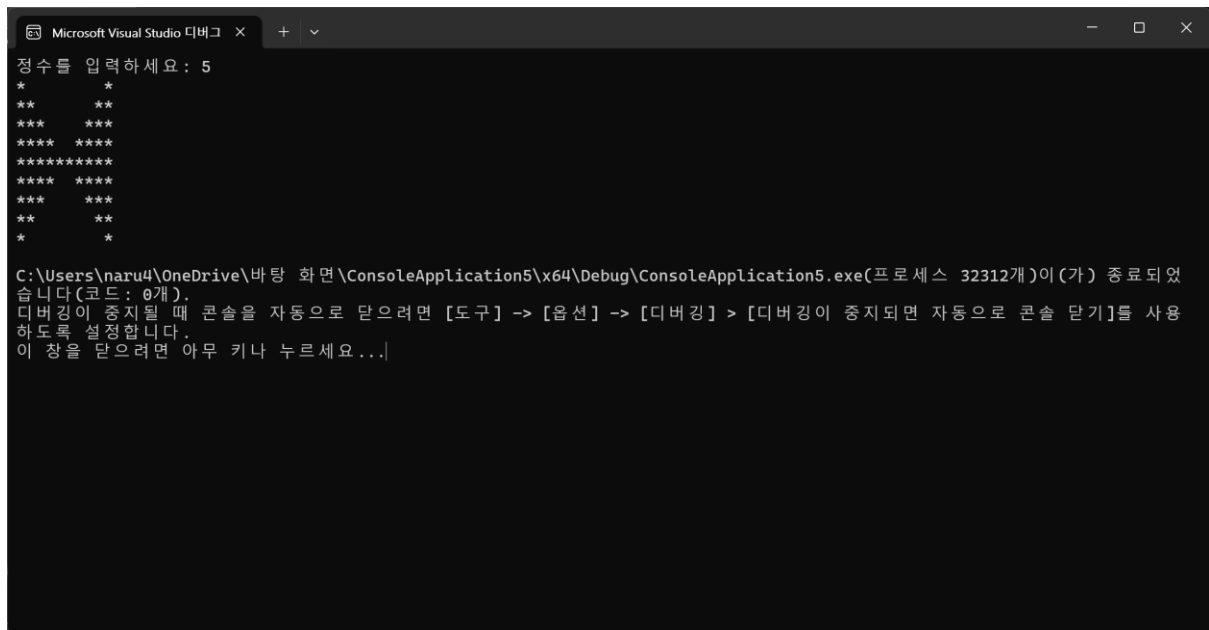
```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main()
{
    int num;
    printf("정수를 입력하세요: ");
    scanf("%d", &num);

    for (int i = 1; i <= num; i++)
    {
        for (int j = 1; j <= i; j++)
        {
            printf("x");
        }
        for (int j = 1; j <= 2 * (num - i); j++)
        {
            printf(" ");
        }
        for (int j = 1; j <= i; j++)
        {
            printf("x");
        }
        printf("\n");
    }
    for (int i = num - 1; i >= 1; i--)
    {
        for (int j = 1; j <= i; j++)
        {
            printf("x");
        }
        for (int j = 1; j <= 2 * (num - i); j++)
        {
            printf(" ");
        }
        for (int j = 1; j <= i; j++)
        {
            printf("x");
        }
        printf("\n");
    }

    return 0;
}
```

실행결과



```
Microsoft Visual Studio 디버그
정수들 입력하세요 : 5
*
**
***
****
*****
****
***
**
*

C:\Users\naru4\OneDrive\바탕 화면\ConsoleApplication5\x64\Debug\ConsoleApplication5.exe(프로세스 32312개)이(가) 종료되었습니다(코드: 0개).
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...
```

2일차 과제 4번 소스코드

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

int check(int N, int B, int A[])
{
    int count = 0;
    int length;
    int t;
    int i;
    for (length = 1; length <= N; length += 2) //길이가 홀수여야 중앙값이 내부에
    포함됨
    {
        for (t = 0; t <= N - length; t++) // 길이에 따른 부분수열 종류
        {
            int subA[100];
            for (i = 0; i < length; i++)
            {
                subA[i] = A[t + i]; //부분함수 생성
            }

            for (i = 0; i < length - 1; i++)
            {
                for (int j = 0; j < length - i - 1; j++)
                {
                    if (subA[j] > subA[j + 1]) //부분 수열 정렬
                    {
                        int temp = subA[j];
```

```

        subA[j] = subA[j + 1];
        subA[j + 1] = temp;
    }
}
}
int mid = length / 2;
if (subA[mid] == B)
{
    count++;
}
}
}

return count;
}

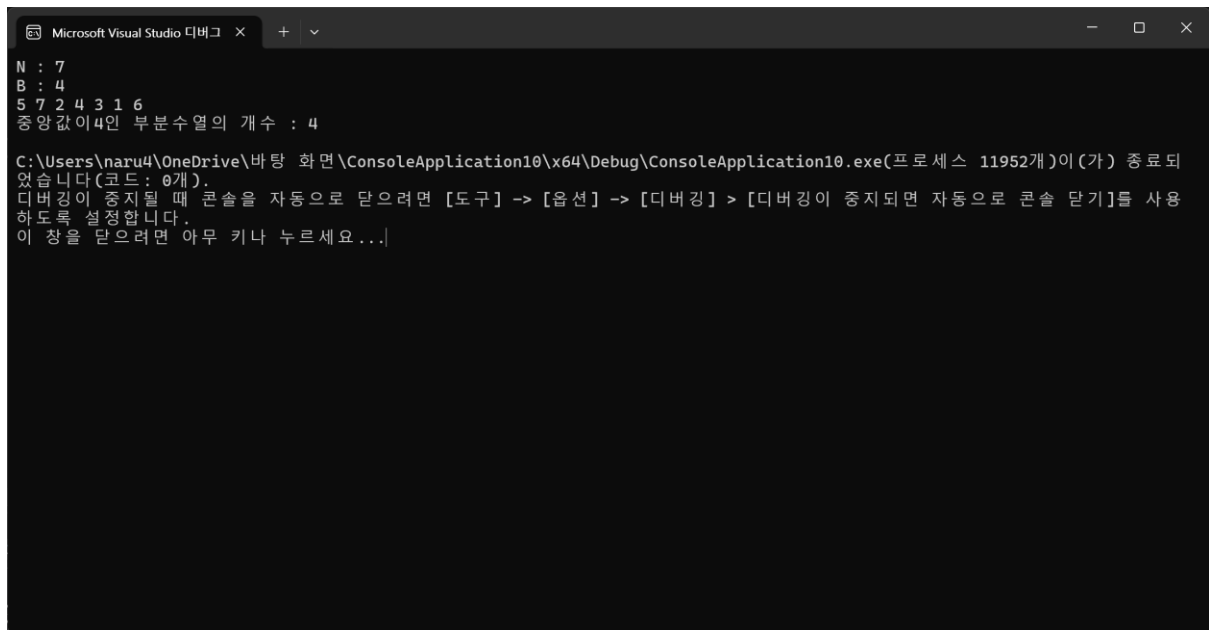
int main()
{
    int N, B;
    int A[100] = { 0, };

    printf("N : ");
    scanf("%d", &N);
    printf("B : ");
    scanf("%d", &B);
    for (int i = 0; i < N; i++)
    {
        scanf("%d", &A[i]);
    }
    int result = check(N, B, A);
    printf("중앙값이%d인 부분수열의 개수 : %d\n",B,result);

    return 0;
}

```

실행 결과



```
Microsoft Visual Studio 디버그 x + v
N : 7
B : 4
5 7 2 4 3 1 6
중앙값이 4인 부분수열의 개수 : 4

C:\Users\naru4\OneDrive\바탕 화면\ConsoleApplication10\Debug\ConsoleApplication10.exe(프로세스 11952개)이(가) 종료되었습니다(코드: 0개).
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...
```

8일차 과제 4번 소스코드

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

typedef struct _Node
{
    char data;
    struct _Node* next;
} Node;

typedef struct _Stack
{
    Node* top;
    int size;
} Stack;

typedef struct _Queue
{
    Node* front;
    Node* rear;
    int size;
} Queue;

void _Stack(Stack* stack);
void push(Stack* stack, char data);
char pop(Stack* stack);
int isEmptyStack(Stack* stack);
```

```

void _Queue(Queue* queue);
void Enqueue(Queue* queue, char data);
char Dequeue(Queue* queue);
int isEmptyQueue(Queue* queue);

int check(char* str);

int main()
{
    char str[100];

    printf("문자열을 입력하세요: ");
    scanf_s("%[^\\n]", str, sizeof(str));

    if (check(str)==1)
    {
        printf("회문입니다.\\n");
    }
    else
    {
        printf("회문이 아닙니다.\\n");
    }

    return 0;
}

void _Stack(Stack* stack)
{
    stack->top = NULL;
    stack->size = 0;
}

void push(Stack* stack, char data)
{
    Node* newNode = (Node*)malloc(sizeof(Node));
    newNode->data = data;
    newNode->next = stack->top;
    stack->top = newNode;
    stack->size++;
}

char pop(Stack* stack)
{
    if (stack->top == NULL)
    {
        return 'W0';
    }
}

```

```

    }
    Node* temp = stack->top;
    char data = temp->data;
    stack->top = stack->top->next;
    free(temp);
    stack->size--;
    return data;
}

int isEmptyStack(Stack* stack)
{
    return stack->size == 0;
}

void _Queue(Queue* queue)
{
    queue->front = NULL;
    queue->rear = NULL;
    queue->size = 0;
}

void Enqueue(Queue* queue, char data)
{
    Node* newNode = (Node*)malloc(sizeof(Node));
    newNode->data = data;
    newNode->next = NULL;

    if (queue->rear == NULL)
    {
        queue->front = newNode;
        queue->rear = newNode;
    }
    else
    {
        queue->rear->next = newNode;
        queue->rear = newNode;
    }
    queue->size++;
}

char Dequeue(Queue* queue)
{
    if (queue->front == NULL)
    {
        return 'W0';
    }

```



```

Node* temp = queue->front;
char data = temp->data;
queue->front = queue->front->next;

if (queue->front == NULL)
{
    queue->rear = NULL;
}

free(temp);
queue->size--;
return data;
}

int isEmptyQueue(Queue* queue)
{
    return queue->size == 0;
}

int check(char* str)
{
    Stack stack;
    Queue queue;
    _Stack(&stack);
    _Queue(&queue);

    for (int i = 0; str[i] != '\0'; i++)
    {
        push(&stack, str[i]);
        Enqueue(&queue, str[i]);
    }
    while (!isEmptyStack(&stack) && !isEmptyQueue(&queue))
    {
        if (pop(&stack) != Dequeue(&queue))
        {
            return 0;
        }
    }
    return 1;
}

```

실행결과

```
Microsoft Visual Studio 디버그 x + v
문자열을 입력하세요: oloolo
회문입니다.

C:\Users\naru4\OneDrive\바탕 화면\ConsoleApplication8\x64\Debug\ConsoleApplication8.exe(프로세스 9592개)이(가) 종료되었습니다(코드: 0개).
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...
```

```
Microsoft Visual Studio 디버그 x + v
문자열을 입력하세요: rewafdsadfqwe
회문이 아닙니다.

C:\Users\naru4\OneDrive\바탕 화면\ConsoleApplication8\x64\Debug\ConsoleApplication8.exe(프로세스 18008개)이(가) 종료되었습니다(코드: 0개).
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...
```

11일차 과제 1번 소스코드

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct
{
    int num;
    char name[30];
```

```

    char country[30];
    char prov[30];
    char city[30];
    char district[30];
    int grade;
} student;

int main()
{
    student students[100];
    int count = 0;
    int opp, opp2;

    FILE* file = fopen("students.txt", "r");
    if (file != NULL)
    {
        while (fscanf(file, "%d %s %s %s %s %s %d", &students[count].num,
students[count].name, students[count].country, students[count].prov,
students[count].city, students[count].district, &students[count].grade) != -1)
        {
            count++;
        }
        fclose(file);
    }

    while (1)
    {
        printf("숫자를 입력하세요\n학생을 추가하려면 : 1\n학생을 삭제하려면 :
2\n학생을 찾으려면 : 3\n학생을 정렬하려면 : 4\n학생 목록을 저장하려면 :
5\n종료하려면 : 6\n");
        scanf("%d", &opp);

        if (opp == 1)
        {
            if (count >= 100)
            {
                printf("학생을 더 이상 추가할 수 없습니다.\n");
            }
            else
            {
                printf("학생 번호: ");
                scanf("%d", &students[count].num);
                printf("학생 이름: ");
                scanf("%s", students[count].name);
                printf("나라: ");
                scanf("%s", students[count].country);
                printf("도: ");
            }
        }
    }
}

```

```

        scanf("%s", students[count].prov);
        printf("시: ");
        scanf("%s", students[count].city);
        printf("구: ");
        scanf("%s", students[count].district);
        printf("성적: ");
        scanf("%d", &students[count].grade);
        count++;
    }
}
else if (opp == 2)
{
    int num;
    printf("삭제할 학생의 번호를 입력하세요: ");
    scanf("%d", &num);
    for (int i = 0; i < count; i++)
    {
        if (students[i].num == num)
        {
            for (int j = i; j < count - 1; j++)
            {
                students[j] = students[j + 1];
            }
            count--;
            break;
        }
    }
}
else if (opp == 3)
{
    int num;
    printf("찾을 학생의 번호를 입력하세요: ");
    scanf("%d", &num);
    for (int i = 0; i < count; i++)
    {
        if (students[i].num == num)
        {
            printf("번호: %d, 이름: %s, 나라: %s, 도: %s, 시: %s, 구: %s, 성적: %d\n",
                students[i].num, students[i].name, students[i].country,
                students[i].prov, students[i].city, students[i].district, students[i].grade);
            break;
        }
    }
}
else if (opp == 4)
{

```

```

printf("어떤 순서로 학생을 정렬할건지 숫자를 입력하세요\n번호순 :
1\n이름순 : 2\n성적순 : 3\n");
scanf("%d", &opp2);
if (opp2 == 1)
{
    for (int i = 0; i < count - 1; i++)
    {
        for (int j = 0; j < count - i - 1; j++)
        {
            if (students[j].num > students[j + 1].num)
            {
                student temp = students[j];
                students[j] = students[j + 1];
                students[j + 1] = temp;
            }
        }
    }
}
else if (opp2 == 2)
{
    for (int i = 0; i < count - 1; i++)
    {
        for (int j = 0; j < count - i - 1; j++)
        {
            if (strcmp(students[j].name, students[j + 1].name) > 0)
            {
                student temp = students[j];
                students[j] = students[j + 1];
                students[j + 1] = temp;
            }
        }
    }
}
else if (opp2 == 3)
{
    for (int i = 0; i < count - 1; i++)
    {
        for (int j = 0; j < count - i - 1; j++)
        {
            if (students[j].grade < students[j + 1].grade)
            {
                student temp = students[j];
                students[j] = students[j + 1];
                students[j + 1] = temp;
            }
        }
    }
}

```

```

    }
    else
    {
        printf("잘못된 선택입니다.\n");
    }
    for (int i = 0; i < count; i++)
    {
        printf("번호: %d, 이름: %s, 나라: %s, 도: %s, 시: %s, 구: %s, 성적: %d\n", students[i].num, students[i].name, students[i].country, students[i].prov, students[i].city, students[i].district, students[i].grade);
    }
}
else if (opp == 5)
{
    FILE* file = fopen("students.txt", "w");
    if (file == NULL)
    {
        printf("파일을 열 수 없습니다.\n");
    }
    else {
        for (int i = 0; i < count; i++)
        {
            fprintf(file, "%d %s %s %s %s %s %d\n", students[i].num, students[i].name, students[i].country, students[i].prov, students[i].city, students[i].district, students[i].grade);
        }
        fclose(file);
    }
}
else if (opp == 6)
{
    FILE* file = fopen("students.txt", "r");
    if (file == NULL)
    {
        printf("파일을 열 수 없습니다.\n");
    }
    else
    {
        count = 0;
        while (fscanf(file, "%d %s %s %s %s %s %d", &students[count].num, students[count].name, students[count].country, students[count].prov, students[count].city, students[count].district, &students[count].grade) != -1)
        {
            count++;
        }
        fclose(file);
    }
}

```

```

    }
    else if (opp == 7)
    {
        FILE* file = fopen("students.txt", "w");
        if (file != NULL)
        {
            for (int i = 0; i < count; i++)
            {
                fprintf(file, "%d %s %s %s %s %s %d\n", students[i].num,
students[i].name, students[i].country, students[i].prov, students[i].city,
students[i].district, students[i].grade);
            }
            fclose(file);
        }
        break;
    }
    else
    {
        printf("잘못된 숫자입니다.\n");
    }
}
return 0;
}

```

실행결과

```

C:\Users\Wnaru4W\OneDrive\W...
숫자들 입력하세요
학생을 추가하려면 : 1
학생을 삭제하려면 : 2
학생을 찾으려면 : 3
학생을 정렬하려면 : 4
학생 목록을 저장하려면 : 5
종료하려면 : 6
1
학생 번호 : 1
학생 이름 : 정
나라 : 대
도 : 한
시 : 민
구 : 국
성적 : 90
숫자들 입력하세요
학생을 추가하려면 : 1
학생을 삭제하려면 : 2
학생을 찾으려면 : 3
학생을 정렬하려면 : 4
학생 목록을 저장하려면 : 5
종료하려면 : 6
1
학생 번호 : 2
학생 이름 : 지
나라 : 국
도 : 민
시 : 한
구 : 대
성적 : 89

```

```
C:\Users\Wnaru4\OneDrive\W... x + v
숫자 들 입력 하 세 요
학 생 을 추 가 하 려 면 : 1
학 생 을 삭 제 하 려 면 : 2
학 생 을 찾 으 려 면 : 3
학 생 을 정 령 하 려 면 : 4
학 생 목 록 을 저 장 하 려 면 : 5
종 료 하 려 면 : 6
1
학 생 번 호 : 3
학 생 이 름 : 안
나 라 : 대
도 : 한
시 : 민
구 : 서
성 적 : 76
숫 자 들 입 력 하 세 요
학 생 을 추 가 하 려 면 : 1
학 생 을 삭 제 하 려 면 : 2
학 생 을 찾 으 려 면 : 3
학 생 을 정 령 하 려 면 : 4
학 생 목 록 을 저 장 하 려 면 : 5
종 료 하 려 면 : 6
3
찾 을 학 생 의 번 호 들 입 력 하 세 요 : 2
번 호 : 2, 이 름 : 지, 나 라 : 대, 도 : 서 울, 시 : 송, 구 : 송 파 구, 성 적 : 84
숫 자 들 입 력 하 세 요
학 생 을 추 가 하 려 면 : 1
학 생 을 삭 제 하 려 면 : 2
학 생 을 찾 으 려 면 : 3
학 생 을 정 령 하 려 면 : 4
```

```
C:\Users\Wnaru4\OneDrive\W... x + v
학 생 을 정 령 하 려 면 : 4
학 생 목 록 을 저 장 하 려 면 : 5
종 료 하 려 면 : 6
4
어 떤 순 서 로 학 생 을 정 령 할 건 지 숫 자 들 입 력 하 세 요
번 호 순 : 1
이 름 순 : 2
성 적 순 : 3
3
번 호 : 1, 이 름 : 정, 나 라 : 대, 도 : 서, 시 : 서 울, 구 : 송 파 구, 성 적 : 92
번 호 : 1, 이 름 : 정, 나 라 : 대, 도 : 한, 시 : 민, 구 : 국, 성 적 : 90
번 호 : 2, 이 름 : 지, 나 라 : 국, 도 : 민, 시 : 한, 구 : 대, 성 적 : 89
번 호 : 2, 이 름 : 지, 나 라 : 대, 도 : 서 울, 시 : 송, 구 : 송 파 구, 성 적 : 84
번 호 : 3, 이 름 : 안, 나 라 : 대, 도 : 한, 시 : 민, 구 : 서, 성 적 : 76
번 호 : 4, 이 름 : 안, 나 라 : 대, 도 : 한, 시 : 민, 구 : 국, 성 적 : 67
숫 자 들 입 력 하 세 요
학 생 을 추 가 하 려 면 : 1
학 생 을 삭 제 하 려 면 : 2
학 생 을 찾 으 려 면 : 3
학 생 을 정 령 하 려 면 : 4
학 생 목 록 을 저 장 하 려 면 : 5
종 료 하 려 면 : 6
5
숫 자 들 입 력 하 세 요
학 생 을 추 가 하 려 면 : 1
학 생 을 삭 제 하 려 면 : 2
학 생 을 찾 으 려 면 : 3
학 생 을 정 령 하 려 면 : 4
학 생 목 록 을 저 장 하 려 면 : 5
종 료 하 려 면 : 6
```




| 정 대 서 서울 송파구 92
1 정 대 한 민 국 90
2 지 국 민 한 대 89
2 지 대 서울 송 송파구 84
3 안 대 한 민 서 76
4 안 대 한 민 국 67