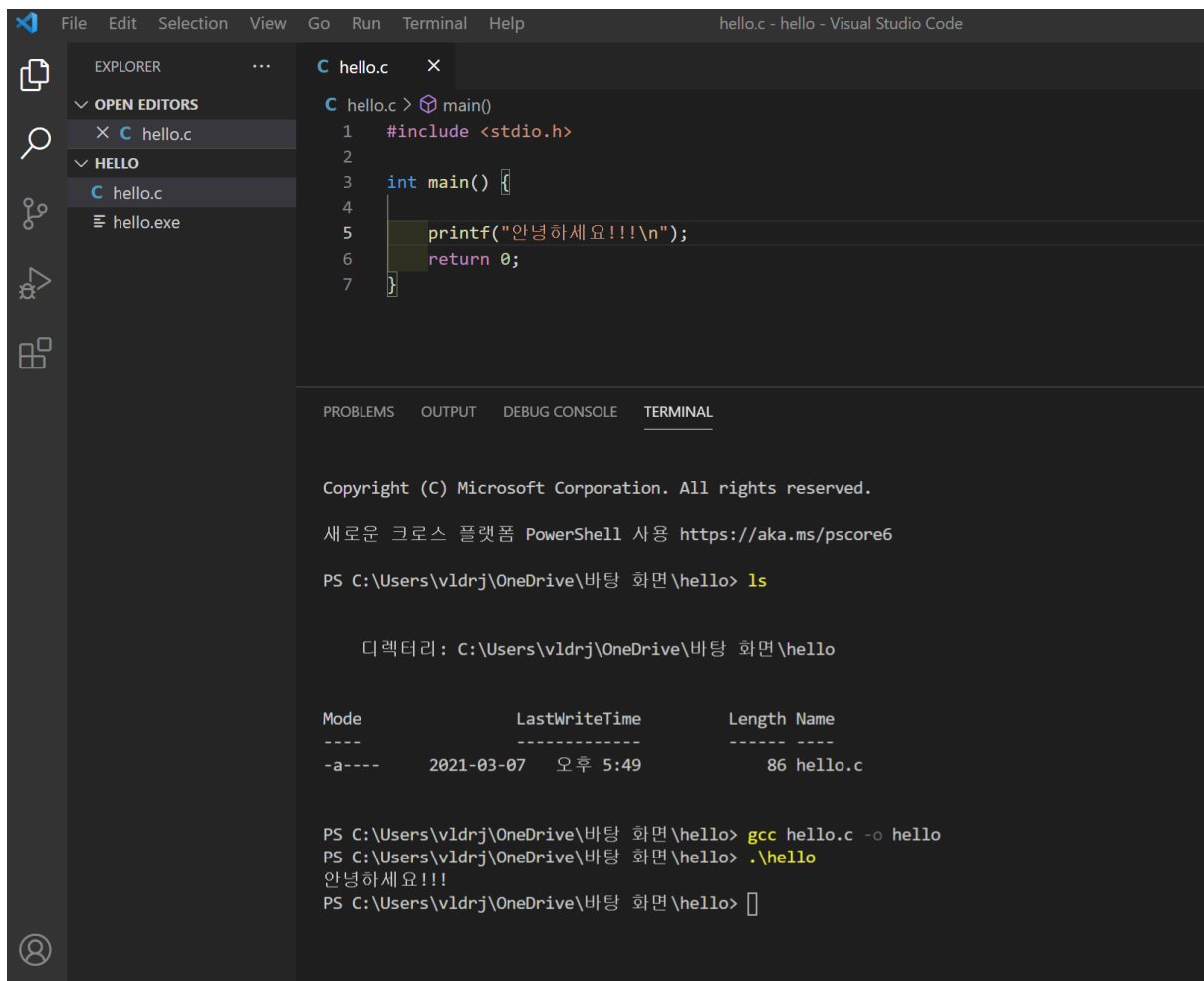


#1. C programmin 사용환경 설정



2. 소스코드_1

```
#include<stdio.h>

int main() {

    int i, j,num1, num2, num3;
    int sum1, sum2, sum3, sum4, sum5;
    double avg1, avg2, avg3, avg4, avg5;
    char grade1, grade2, grade3, grade4, grade5;

    printf("1 번 학생 : ");
    scanf("%d %d %d", &num1, &num2, &num3);
    sum1 = num1 + num2 + num3;
    avg1 = sum1 / 3.0;

    printf("2 번 학생 : ");
    scanf("%d %d %d", &num1, &num2, &num3);
    sum2 = num1 + num2 + num3;
```

```
avg2 = sum2 / 3.0;

printf("3 번 학생 : ");
scanf("%d %d %d", &num1, &num2, &num3);
sum3 = num1 + num2 + num3;
avg3 = sum3 / 3.0;

printf("4 번 학생 : ");
scanf("%d %d %d", &num1, &num2, &num3);
sum4 = num1 + num2 + num3;
avg4 = sum4 / 3.0;

printf("5 번 학생 : ");
scanf("%d %d %d", &num1, &num2, &num3);
sum5 = num1 + num2 + num3;
avg5 = sum5 / 3.0;

if(avg1 >= 90) {
    grade1 = 'A';
} else if(avg1 >= 80) {
    grade1 = 'B';
} else if(avg1 >= 70) {
    grade1 = 'C';
} else if(avg1 >= 60) {
    grade1 = 'D';
} else if(avg1 < 60) {
    grade1 = 'F';
}

if(avg2 >= 90) {
    grade2 = 'A';
} else if(avg2 >= 80) {
    grade2 = 'B';
} else if(avg2 >= 70) {
    grade2 = 'C';
} else if(avg2 >= 60) {
    grade2 = 'D';
} else if(avg2 < 60) {
    grade2 = 'F';
}

if(avg3 >= 90) {
    grade3 = 'A';
} else if(avg3 >= 80) {
    grade3 = 'B';
} else if(avg3 >= 70) {
    grade3 = 'C';
} else if(avg3 >= 60) {
```

```

        grade3 = 'D';
    } else if(avg3 < 60) {
        grade3 = 'F';
    }

    if(avg4 >= 90) {
        grade4 = 'A';
    } else if(avg4 >= 80) {
        grade4 = 'B';
    } else if(avg4 >= 70) {
        grade4 = 'C';
    } else if(avg4 >= 60) {
        grade4 = 'D';
    } else if(avg4 < 60) {
        grade4 = 'F';
    }

    if(avg5 >= 90) {
        grade5 = 'A';
    } else if(avg5 >= 80) {
        grade5 = 'B';
    } else if(avg5 >= 70) {
        grade5 = 'C';
    } else if(avg5 >= 60) {
        grade5 = 'D';
    } else if(avg5 < 60) {
        grade5 = 'F';
    }

    printf("\n");

    printf("%d 번 학생의 총점은 %d, 평균은 %.1f(등급 %c)\n", j, sum1, avg1, grade1);
    printf("%d 번 학생의 총점은 %d, 평균은 %.1f(등급 %c)\n", j, sum2, avg2, grade2);
    printf("%d 번 학생의 총점은 %d, 평균은 %.1f(등급 %c)\n", j, sum3, avg3, grade3);
    printf("%d 번 학생의 총점은 %d, 평균은 %.1f(등급 %c)\n", j, sum4, avg4, grade4);
    printf("%d 번 학생의 총점은 %d, 평균은 %.1f(등급 %c)\n", j, sum5, avg5, grade5);

    return 0;
}

```

```

#include <stdio.h>

int main() {

    int i, j, num1, num2, num3;
    int sum[5];
    double aver[5];
    char grade[5];

    for(i = 0 ; i < 5 ; i++) {
        printf("%d 번 학생 : ", i+1);
        scanf("%d %d %d", &num1, &num2, &num3);
        sum[i] = num1 + num2 + num3;
        aver[i] = sum[i] / 3.0;

        if(aver[i] >= 90) {
            grade[i] = 'A';
        } else if(aver[i] >= 80) {
            grade[i] = 'B';
        } else if(aver[i] >= 70) {
            grade[i] = 'C';
        } else if(aver[i] >= 60) {
            grade[i] = 'D';
        } else if(aver[i] < 60) {
            grade[i] = 'F';
        }
    }
    printf("\n");

    for(j = 0; j < 5; j++) {
        printf("%d 번 학생의 총점은 %d, 평균은 %.1f(등급 %c)\n", j, sum[j], aver
[j], grade[j]);
    }

    return 0;
}

```

소스코드_3

```

#include <stdio.h>

typedef struct student {
    int score[3];
    char name[20];
    char grade[10];
} Student;

```

```

int InputScore(int i, Student *s) {
    int sum = 0;
    printf("%d 번 학생: ", i+1);

    for(int j = 0 ; j < 3 ; j++) {
        scanf("%d", &s->score[j]);
        sum += s->score[j];
    }
    return sum;
}

void OutputScore(int *sum, double *avg, Student *s) {
    for(int i = 0 ; i < 5 ; i++) {

        if(avg[i] >= 90) {
            s->grade[i] = 'A';
        } else if(avg[i] >= 80) {
            s->grade[i] = 'B';
        } else if(avg[i] >= 70) {
            s->grade[i] = 'C';
        } else if(avg[i] >= 60) {
            s->grade[i] = 'D';
        } else if(avg[i] < 60) {
            s->grade[i] = 'F';
        }

        printf("%d 학생의 총점은 %d, 평균은 %.1f(등급 %c)\n", i+1, sum[i], avg[i], s->grade[i]);
    }
}

int main() {
    Student s1;
    int seperate_sum[5];
    double seperate_avg[5];

    for(int i = 0 ; i < 5 ; i++) {
        seperate_sum[i] = InputScore(i,&s1);
        seperate_avg[i] = seperate_sum[i] / 3.0;
    }
    printf("\n");
    OutputScore(seperate_sum, seperate_avg, &s1);

    return 0;
}

```

3. 결과화면

```
1번 학생 : 85 95 75
2번 학생 : 90 80 70
3번 학생 : 65 85 75
4번 학생 : 60 70 80
5번 학생 : 60 50 60

1번 학생의 총점은 255, 평균은 85.0(등급B)
2번 학생의 총점은 240, 평균은 80.0(등급B)
3번 학생의 총점은 225, 평균은 75.0(등급C)
4번 학생의 총점은 210, 평균은 70.0(등급C)
5번 학생의 총점은 170, 평균은 56.0(등급F)
PS C:\Users\vldrj\OneDrive\바탕 화면\CProject> |
```

4. dynalist link

<https://dynalist.io/d/-tANU0aH-vKe77GiHLCzoEeV>

- Data Type(자료형)의 종류
 - 정수 자료형
 - 문자형
 - char 1BYTE
 - 부호가 있는 char
 - 부호가 없는 (unsigned)char
 - 정수형
 - 부호가 있는 정수형
 - short 2BYTE
 - int 4BYTE
 - long 4BYTE
 - 부호가 없는 정수형
 - (unsigned)short 2BYTE
 - (unsigned)int 4BYTE
 - (unsigned)long 4BYTE
 - 실수 자료형
 - 나열형
 - enum
 - 열거형이란 말 그대로 나열한다는 뜻
 - 프로그래밍 중 자주 사용하는 변수가 있으면 이들의 값을 정수로 표현할 때 유용함
 - 무차형
 - void

- void 포인터 매개변수를 활용하면 자료형을 변환하지 않아도 모든 자료형을 함수에 추가할 수 있음.
- 조건문
 - 조건문의 종류는 if-else, switch-case 문이 대표적으로 있음
 - if -else
 - if () 조건문이 참이면 실행, 거짓이면 실행하지 않음
 - else 의 경우 거짓인 경우 실행되는 조건문
 - else if 문을 사용해 조건문이 아닐 경우 다른 조건을 탐색하도록 조절
 - switch - case
 - switch()의 입력변수의 값과 일치하는 case 입력값이 있다면 case 문의 문장이 실행됨
- 반복문(이중 반복문)
 - for 반복문
 - for(int i = 0 ; i < n ; i++)이 대표적인 예
 - 정수형 변수 i = 0 일 때부터 n-1 의 값을 가질 때까지 총 n 번 실행함
 - for 문 안에 있는 문장을 n 번 반복 실행하게 됨
 - while 반복문
 - for 문과 다르게 몇 번 반복되는 것인지 ()가 아닌 while {} 전체 구조 안에 실행해야 함
 - 보통 while 문과 if 문이 같이 쓰임
 - if 의 조건을 만족시키면 while 루프를 break 하는 사례가 대표적
 - do-while 반복문
 - 단순 while 반복문의 구조와 다르게 do{}의 구조를 실행 후 while()의 조건문을 검사해 반복을 결정한다.
 - while()의 조건문이 단순 while 반복문의 break 조건문이라고 보면 됨
 - 이중 반복문
 - 보통 for 문 안에 for 문이 있는 형식이다.
 - 바깥의 for 문이 1 회 시행될 때 안의 for 문이 n 회 모두 반복됨
 - 바깥의 for 문이 n 회 모두 반복될 때 시행이 종료됨
 - 둘 다 n 회 반복 반복문인 경우 총 반복 횟수는 n^2 이 됨
- 배열
 - ex) int score[4] = {100, 90, 80, 70}; //선언 및 초기화 문법
 - 같은 타입의 변수들로 이루어진 유한 집합 으로 정의됨
 - 배열을 구성하는 각각의 값을 배열 요소(element)라고 함
 - 배열에서의 위치를 가리키는 숫자는 인덱스(index)라고 함
- 문자열 다루기
 - 문자열(String)은 문자(Char)가 여러개 있다고 보면 됨
 - 선언 방법

- `char str[10] = "ABCD";`
 - 문자형은 `char` 이고 `str` 변수이름 선언과 `"ABCD"`로 초기화하면 됨
- 함수 만들기
 - `main` 함수에서 기능을 구현하는 부분을 따로 떼어 구현하는 것
 - 함수 문법
 - 변환자료형(`int, void, float...`) 함수명(`GetScore, ShowResult...`) (매개변수 목록(`int a, char ch...`)) { }