

BETA



Ykwan2

@Ykwan2 (0)

 Upgrade

 Edit

Repls Posts

Pick a repl to showcase!

Pick



EXPLORER



C hello.c



연습 코딩

> .vscode

> hello.dSYM

> normal.dSYM

≡ arrayV

C arrayV.c

≡ arry

≡ cpv

≡ hello

C hello.c

C hello.cpp

C normalV.c

≡ spv

C SPV.c

C hello.c > main()

```
1  #include <stdio.h>
2
3  int main() {
4      printf("안녕하세요!!!\n");
5      return 0;
6  }
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL



```
/Users/youngkwanch0/.zprofile:1: no such file or directory: /opt/homebrew/bin/brew
youngkwanch0@joyeong-gwan-ui-MacBookPro 연습 코딩 % gcc hello.c -o hello
youngkwanch0@joyeong-gwan-ui-MacBookPro 연습 코딩 % ./hello
안녕하세요 !!!
youngkwanch0@joyeong-gwan-ui-MacBookPro 연습 코딩 %
```

zsh \xe1\x8...

zsh \xe1\x8...



> OUTLINE

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(등급 B)

2번 학생의 총점은 240, 평균은 80.0(등급 B)

3번 학생의 총점은 225, 평균은 75.0(등급 C)

4번 학생의 총점은 210, 평균은 70.0(등급 C)

5번 학생의 총점은 170, 평균은 56.7(등급 E)

1. 일반 Version

```
#include <stdio.h>

int main() {
    int a1, a2, a3, b1, b2, b3, c1, c2, c3, d1, d2, d3, e1, e2, e3;
    int total1, total2, total3, total4, total5;
    double ave1, ave2, ave3, ave4, ave5;

    printf("1 번 학생 : ");
    scanf("%d %d %d", &a1, &a2, &a3);
    printf("\n2 번 학생 : ");
    scanf("%d %d %d", &b1, &b2, &b3);
    printf("\n3 번 학생 : ");
    scanf("%d %d %d", &c1, &c2, &c3);
    printf("\n4 번 학생 : ");
    scanf("%d %d %d", &d1, &d2, &d3);
    printf("\n5 번 학생 : ");
    scanf("%d %d %d", &e1, &e2, &e3);

    total1=a1+a2+a3;
    total2=b1+b2+b3;
    total3=c1+c2+c3;
    total4=d1+d2+d3;
    total5=e1+e2+e3;
    ave1=total1/3.0;
    ave2=total2/3.0;
    ave3=total3/3.0;
    ave4=total4/3.0;
    ave5=total5/3.0;

    printf("\n1 번 학생의 총점은 %d, 평균은 %.1f", total1, ave1);
    if (ave1>=90.0) printf("(등급 A)\n");
    if (ave1>=80.0 && ave1<90.0) printf("(등급 B)\n");
    if (ave1>=70.0 && ave1<80.0) printf("(등급 C)\n");
    if (ave1>=60.0 && ave1<70.0) printf("(등급 D)\n");
    if (ave1>=50.0 && ave1<60.0) printf("(등급 E)\n");
    if (ave1<50.0) printf("(등급 F)\n");
    printf("\n2 번 학생의 총점은 %d, 평균은 %.1f", total2, ave2);
    if (ave2>=90.0) printf("(등급 A)\n");
    if (ave2>=80.0 && ave2<90.0) printf("(등급 B)\n");
    if (ave2>=70.0 && ave2<80.0) printf("(등급 C)\n");
    if (ave2>=60.0 && ave2<70.0) printf("(등급 D)\n");
    if (ave2>=50.0 && ave2<60.0) printf("(등급 E)\n");
```

```
if (ave2<50.0) printf("(등급 F)\n");
printf("\n3 번 학생의 총점은 %d, 평균은 %.1f", total3, ave3);
if (ave3>=90.0) printf("(등급 A)\n");
if (ave3>=80.0 && ave3<90.0) printf("(등급 B)\n");
if (ave3>=70.0 && ave3<80.0) printf("(등급 C)\n");
if (ave3>=60.0 && ave3<70.0) printf("(등급 D)\n");
if (ave3>=50.0 && ave3<60.0) printf("(등급 E)\n");
if (ave3<50.0) printf("(등급 F)\n");
printf("\n4 번 학생의 총점은 %d, 평균은 %.1f", total4, ave4);
if (ave4>=90.0) printf("(등급 A)\n");
if (ave4>=80.0 && ave4<90.0) printf("(등급 B)\n");
if (ave4>=70.0 && ave4<80.0) printf("(등급 C)\n");
if (ave4>=60.0 && ave4<70.0) printf("(등급 D)\n");
if (ave4>=50.0 && ave4<60.0) printf("(등급 E)\n");
if (ave4<50.0) printf("(등급 F)\n");
printf("\n5 번 학생의 총점은 %d, 평균은 %.1f", total5, ave5);
if (ave5>=90.0) printf("(등급 A)\n");
if (ave5>=80.0 && ave5<90.0) printf("(등급 B)\n");
if (ave5>=70.0 && ave5<80.0) printf("(등급 C)\n");
if (ave5>=60.0 && ave5<70.0) printf("(등급 D)\n");
if (ave5>=50.0 && ave5<60.0) printf("(등급 E)\n");
if (ave5<50.0) printf("(등급 F)\n");

return 0;
}
```

2. 배열 Version

```
#include <stdio.h>

int main() {
    int a[3], b[3], c[3];
    int total[5];
    double ave[5];

    for (int x= 0; x<5; x++) {
        printf("\n%d 번 학생 : ", x+1);
        scanf("%d %d %d", &a[x], &b[x], &c[x]);
    }

    for (int i=0; i<5 ; i++) {
        total[i]=a[i]+b[i]+c[i];
        ave[i]=total[i]/3.0;
    }

    for (int i=0; i<5; i++) {
        printf("\n%d 번 학생의 총점은 %d, 평균은 %.1f", i+1, total[i], ave[i]);
        if (ave[i]>=90.0) printf("(등급 A)\n");
        else if (ave[i]>=80.0 && ave[i]<90.0) printf("(등급 B)\n");
        else if (ave[i]>=70.0 && ave[i]<80.0) printf("(등급 C)\n");
        else if (ave[i]>=60.0 && ave[i]<70.0) printf("(등급 D)\n");
        else if (ave[i]>=50.0 && ave[i]<60.0) printf("(등급 E)\n");
        else printf("(등급 F)\n");
    }

    return 0;
}
```

3. 구조체, 포인터, 함수 Version

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

struct student {
    int a, b, c;
    int total;
    double ave;
    char d;
} p[5];

void averageCal(struct student* p[]) {
    for (int i=0; i<5 ; i++) {
        p[i]->ave=p[i]->total/3.0;
    }
}

void totalCal(struct student* p[]) {
    for (int i=0; i<5 ; i++) {
        p[i]->total=p[i]->a+p[i]->b+p[i]->c;
    }
}

void gradeCal(struct student* p[]) {
    for (int i=0; i<5 ; i++) {
        if (p[i]->ave>=90.0) p[i]->d='A';
        else if (p[i]->ave>=80.0 && p[i]->ave<90.0) p[i]->d='B';
        else if (p[i]->ave>=70.0 && p[i]->ave<80.0) p[i]->d='C';
        else if (p[i]->ave>=60.0 && p[i]->ave<70.0) p[i]->d='D';
        else if (p[i]->ave>=50.0 && p[i]->ave<60.0) p[i]->d='E';
        else p[i]->d='F';
    }
}

int main() {
    struct student *p[5];
    for (int i=0; i< sizeof(p) / sizeof(struct student *);i++){
        p[i]=malloc(sizeof(struct student));
    }

    for (int x= 0; x<5; x++) {
        printf("\n%d 번 학생 : ", x+1);
        scanf("%d %d %d", &p[x]->a, &p[x]->b, &p[x]->c);
    }

    totalCal(p);
```

```
averageCal(p);
gradeCal(p);

for (int i=0; i<5; i++) printf("\n%d 번 학생의 총점은 %d, 평균은 %.1f(등급 %c)\n", i+1,p[i]->total,p[i]->ave, p[i]->d);

for (int i = 0; i < sizeof(p) / sizeof(struct student *); i++)
{
    free(p[i]);
}

return 0;
}
```


C Programming : 문법정리

데이터 타입

자료형 타입, 연산자, 수식

형(Type) : 변수와 상수들이 취할값과 그것에 가해질수 있는 연산을 결정한다.

연산자 : 변수와 상수를 어떻게 연산할지를 결정한다. (+, -, *, % 등)

수식 : 변수와 상수를 묶어 새로운 결과를 나타낸다

변수 데이터 형과 크기

char : 한바이트문자 (8비트)

int : 정수 (32비트)

float : 단정도 (소수점이하 정밀도 6자리 이상)

double : 배정도 (소수점이하 정밀도 10자리 이상)

조건문

연산자

$a == b$: a와 b가 같다

$a != b$: a와 b가 다르다

$a > b$: a가 b보다 크다

$a < b$: b가 a보다 크다

if문 :

if (조건) {실행;}

else문 :

if (조건) {실행1;}

else {실행2;}

else if문 :

```
if (조건1) {실행1;}

else if (조건2) {실행2;}

else if (조건3) {실행3;}

else {실행4;}
```

반복문(+이중반복문)

for문 :

```
for (실행1; 조건1; 실행3) {

    실행2;

}
```

while문 :

```
while(조건) {

    실행;

}
```

do while 문 :

```
do {

    실행;

} while(조건);
```

break 문 : 반복문을 강제로 종료 시키는 구문

continue 문 : 반복문 아래 명령문들을 스킵하고 다시 반복문 처음으로 돌아가는 구문

이중반복문 : 반복문을 중첩하여 사용하는 것

```
for(int i=1; i<=9; i++){

    for(int j=1; j<=9; j++){

        printf("%d*%d=%d ",j,i,i*j);
```

```
}  
  
printf("\n");  
  
}
```

배열

변수의 개수가 많이 필요할때 사용

변수유형 배열이름[변수개수]로 선언

실제 입력되는 변수는 이름[0]부터 이름[변수개수-1]; 까지 이다.

문자열 다루기

문자열 출력할때 : `printf("%s", 문자열의 주소);`

문자열을 출력하고 줄 바꿈을 할때 : `puts(문자열의 주소);`

문자열 입력받을때 : `scanf("%s", 문자열의 주소);`

입력받을때 띄어쓰기도 입력이 필요 할때 : `gets(문자열의 주소);`

문자열 길이를 반환해주는 함수 : `strlen(문자열의 주소);`

두 문자열을 비교하는 함수 : `strcmp(문자열1, 문자열2);`

같을때 0, 첫 번째 문자열이 사전순으로 빠를 때 = -1, 첫 번째 문자열이 사전순으로 느릴 때 = 1

문자열을 복사 할때 : `strcpy(문자열1, 문자열2) -> 문자열2를 문자열1로 복사`

두 문자열을 합칠 때 : `strcat(문자열1, 문자열2)`

함수 만들기

기본적으로 함수를 사용할 때 함수의 정의, 호출, 그리고 선언이 필요

반환값의_형식 함수_이름 (매개변수) {

명령문1;

명령문2;

return 반환값;

```
}
```

함수의 호출은 다른 함수(main())등에서 함수이름과 매개변수를 넣음

함수가 main()함수보다 아래에 위치할 경우, 함수 인식이 되지 않기에 main() 전에 "반환값의_형식 함수_이름 (매개변수);" 형식으로 선언

Link : <https://dynamlist.io/d/gFzpppaZXe0nSFCzoAfPy6DC>