

컴퓨터프로그래밍 및 실습

P1

일반 연산, 확장 연산 계산기



202220761_김가을

I 문제 정의

- 모드 입력값 1 또는 2 입력 받기
- 일반 연산의 경우 연산자 1개와 두 개의 정수 입력받기, 확장 연산의 경우 연산자 2개와 세 개의 정수 입력받기 (쉽표로 구분하여)
- 연산자의 종류를 나눠서 코드 작성
- 0으로 나눌 경우 경고문 출력 후 종료
- 계산은 1회만 실행
- 확장 연산의 경우 연산자 우선순위를 고려하여 계산
- 소수점 3자리까지 출력하므로 서식문자 %.3f 사용
- 연산자 우선순위 고려

II 구현방법

```
int main(void)
{
    int mode;
    printf("Enter the calculator mode (1:Normal, 2:Extended): ");
    scanf("%d",&mode);
```

main함수 안에 모드 입력값을 받을 int형 변수 mode 선언 후
scanf 함수를 통해서 1또는 2의 값을 입력받아 저장

```
if(mode==1)//일반 연산
{
    int a,b;
    char oper;
    printf("Put two integers: ");
    scanf("%d,%d",&a,&b);
    printf("Put the operator: ");
    scanf(" %c",&oper);
```

일반 연산을 실시하기 위해서 main 함수안에 if문 사용해서 mode를 1로 나누고
2개의 정수를 받을 int형 변수 a,b 선언 연산자를 받을 char형 변수 oper 선언
scanf 함수를 통해서 값을 입력받는다.

```

switch(oper)
{
case '+':
    printf("The result is %.3f",Add(a,b));
    break;
case '-':
    printf("The result is %.3f",Subtraction(a,b));
    break;
case '*':
    printf("The result is %.3f",Multiplication(a,b));
    break;
case '%':
    if(b!=0)
        printf("The result is %.3f",Remainder(a,b));
    else
        printf("Zero division error");
    break;
case '/':
    if(b!=0)
        printf("The result is %.3f",Divide(a,b));
    else
        printf("Zero division error");
}

```

switch 문을 통해서 연산자의 종류에 따라 경우의 수를 나누고
여러 가지 함수를 통해 계산값을 출력한다

여기서 / % 연산자의 경우 나누는 수가 0이 되었을 때 오류문을 출력하기 위해 if문을 통해
경우를 나누어서 결과값을 출력해준다.

계산 결과를 위해 만든 함수들

피연산자로 정수만 사용하는 %을 위해서 Remainder 함수만 입력 매개변수를 int형으로
사용

```

double Add (double n1,double n2)
{
    return n1+n2;
}

double Subtraction(double n1,double n2)
{
    return n1-n2;
}

double Divide(double n1,double n2)
{
    return n1/n2;
}

double Remainder(int n1,int n2)
{
    return n1%n2;
}

double Multiplication(double n1,double n2)
{
    return n1*n2;
}

```

```

else if(mode==2)//확장 연산
{
    int a,b,c;
    double result;
    char oper1,oper2;

    printf("Put three integers: ");
    scanf("%d,%d,%d",&a,&b,&c);
    printf("Put two operators: ");
    getchar();
    scanf("%c, %c",&oper1,&oper2);

```

확장 연산의 경우도 if문으로 mode 1과 구분지어 주고 정수 3개와 연산자 2개를 입력받을 변수를 선언한 뒤 이를 scanf를 통해 입력받는다.

일반연산과 달리 확장 연산은 연산자가 2개이고 연산자 우선순위를 고려해야 하므로 중간 연산값을 저장해줄 double 형 변수 result를 선언한다.

```

switch(oper2)
{
case '+':
    result=Add(b,c);
    if(oper1=='+')
        printf("The result is %.3f",Add(a,result));
    else if(oper1=='-')
        printf("The result is %.3f",Subtraction(a,result));
    else if(oper1=='*'){
        result=Multiplication(a,b);
        printf("The result is %.3f",Add(result,c));
    }
    else if(oper1=='/'){
        if(b!=0){
            result=Divide(a,b);
            printf("The result is %.3f",Add(result,c));
        }
        else
            printf("Zero division error");
    }
    else
        if(b!=0){
            result=Remainder(a,b);
            printf("The result is %.3f",Add(result,c));
        }
        else
            printf("Zero division error");
    break;

```

switch문을 통해서 2번째 연산자를 기준으로 경우의 수를 나눠주고 case 안에서 if문을 통해 첫 번째 연산자의 경우의 수를 나누어 결과값을 출력한다.

연산자 우선순위에 따라 2번의 연산을 필요로 하므로 중간연산값을 저장하는 result 변수를 사용해서 우선순위가 먼저인 연산자의 계산값을 먼저 계산하고 result 값을 각각의 함수로 전달해 나머지 계산값을 출력한다.

일반연산과 동일하게 % /연산이 있는 경우 나누는 수가 0일 경우 오류문을 띄우기 위해 if문을 중첩하여 사용한다.

다른 case들도 같은 방식으로 코드를 작성한다.

```
if((b!=0)&&(c!=0)){  
    result=Remainder(a,b);  
    printf("The result is %.3f",Remainder(result,c));  
}  
else  
    printf("Zero division error");
```

특이점은 두 번째 연산자와 첫 번째 연산자의 조합이 / %, % /인 경우는 두 번째와 세 번째 정수 중 하나의 0만 있어도 오류문을 출력해야 하기 때문에 논리연산자를 사용해서 경우를 나눈다.