

8

You

수학 연산을 나타내는 문자열을 처리해 해당 연산을 계산하는 C 언어 프로그램을 다음 순서에 맞게 작성해줘.

- 함수 main

- 무한 루프로 아래 작업들을 반복
  - + 실행 예제와 같이 수학 연산 문자열을 입력
  - + 입력된 문자열은 크기 100의 배열 s에 저장
  - + 배열 s에 저장된 문자열이 "-1"이면 무한 루프를 빠져나감
  - + 함수 eval\_exp를 이용해 연산을 계산하고 결과를 실행 예제와 같이 출력

- 함수 int eval\_exp(char s[])

- 라이브러리 함수 strtok을 이용해 연산자와 2개의 Operand를 문자형 배열 op(크기 10), 정수형 변수 x, y에 저장

- Delimiter는 별도 배열을 이용하지 않고 " "를 이용
- 연산자는 "add", "sub", "mult", "div", "mod"가 에러 없이 입력된다고 가정
- 연산자에 따라 x와 y의 덧셈, 뺄셈, 곱셈, 나눗셈, 나머지 계산의 결과를 반환

- 실행 예제

수학 연산을 입력하시오: 24 add 7

연산의 결과: 31

수학 연산을 입력하시오: 24 sub 7

연산의 결과: 17

수학 연산을 입력하시오: 24 mult 7

연산의 결과: 168

수학 연산을 입력하시오: 24 div 7

연산의 결과: 3

수학 연산을 입력하시오: 24 mod 7

연산의 결과: 3

수학 연산을 입력하시오: -1

토큰 분리할 때 처음은 x 다음 op 다음 y로 분리해줘



ChatGPT

아래는 요구사항에 맞게 작성된 C 언어 프로그램의 예제 코드입니다.

c



Copy code

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

int eval_exp(char s[]);

int main() {
    char s[100];

    while (1) {
        printf("수학 연산을 입력하시오: ");
        fgets(s, sizeof(s), stdin);
        s[strcspn(s, "\n")] = '\0'; // Remove the newline character

        if (strcmp(s, "-1") == 0) {
            break; // Exit the infinite loop if input is "-1"
        }

        int result = eval_exp(s);
        printf("연산의 결과: %d\n", result);
    }

    return 0;
}

int eval_exp(char s[]) {
    char *token;
    char op[10];
    int x, y;
```

```

// Get the first operand
token = strtok(s, " ");
x = atoi(token);

// Get the operator
token = strtok(NULL, " ");
strcpy(op, token);

// Get the second operand
token = strtok(NULL, " ");
y = atoi(token);

// Perform the calculation based on the operator
if (strcmp(op, "add") == 0) {
    return x + y;
} else if (strcmp(op, "sub") == 0) {
    return x - y;
} else if (strcmp(op, "mult") == 0) {
    return x * y;
} else if (strcmp(op, "div") == 0) {
    if (y != 0) {
        return x / y;
    } else {
        printf("Error: Division by zero\n");
        return 0;
    }
} else if (strcmp(op, "mod") == 0) {
    if (y != 0) {
        return x % y;
    } else {
        printf("Error: Modulo by zero\n");
        return 0;
    }
} else {
    printf("Error: Invalid operator\n");
    return 0;
}
}

```

이 코드는 사용자에게 수학 연산을 입력받고, 입력이 "-"인 경우 루프를 빠져나가며, 그렇지 않은 경우 `eval\_exp` 함수를 호출하여 연산을 수행하고 결과를 출력합니다. 함수 내에서는 `strtok` 함수를 사용하여 입력 문자열을 토큰으로 분리하고, 분리된 토큰을 이용하여 연산을 수행합니다.