|  |
| --- |
| 2020년 1학기 |
| 프로그래밍 언어 레포트 |
| 13주차 학습활동 |

|  |
| --- |
| 송 지민  사회학과 201721899 |

**[과제#1] 실습#1, 실습#2 실습수행 및 실행화면 캡처**

1. 실습#1 코드 및 실행화면

#include <string.h>

#include <stdio.h>

#define SIZE 100

int str\_chr(char\* s, int c)

{

int i;

int k = 0;

for (i = 0; s[i] != '\0'; i++)

{

if (s[i] == c)

k++;

}

return k;

}

int main(void)

{

char str[SIZE];

char ch;

printf("문자열을 입력하시오: ");

gets(str);

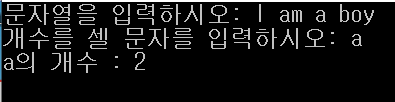
printf("개수를 셀 문자를 입력하시오: ");

ch = getchar();

printf("%c의 개수 : %d\n", ch, str\_chr(str, ch));

return 0;

}



* strchr() 함수 이용 : 문자열에서 문자를 찾아 위치를 알려주기 때문에, 문자열 안에 문자가 있을 경우 매번 j를 증가시켜 오류가 발생합니다.

#include <string.h>

#include <stdio.h>

#define SIZE 100

int main(void)

{

char str[SIZE];

char ch;

char\* p;

int i;

int j = 0;

printf("문자열을 입력하시오: ");

gets(str);

printf("개수를 셀 문자를 입력하시오: ");

ch = getchar();

for (i = 0; str[i] != '\0'; i++)

{

p = strchr(str, ch);

if (p != NULL)

j++;

else

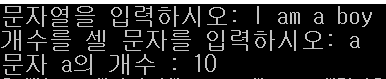
printf("문자열에 없는 문자입니다.\n");

}

printf("문자 %c의 개수 : %d", ch, j);

return 0;

}



* 다음과 같이 수정하여 정상 출력하도록 함

#include <string.h>

#include <stdio.h>

#define SIZE 100

int main(void)

{

char str[SIZE];

char ch;

char\* p;

int j = 0; // 횟수 세는 변수

printf("문자열을 입력하시오: ");

gets(str);

printf("개수를 셀 문자를 입력하시오: ");

ch = getchar();

p = strchr(str, ch); // 문자열에서 문자 찾아 포인터에 저장

while (p != NULL) // 문자가 있을 때

{

j++; // j를 증가시킴

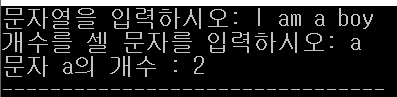
p = strchr(p + 1, ch); // 포인터 주소 1 증가시켜 다시 찾기

}

printf("문자 %c의 개수 : %d", ch, j);

return 0;

}



1. 실습#2 코드 및 실행화면

#include <string.h>

#include <stdio.h>

#define SIZE 100

void str\_upper(char\* s)

{

for (int i = 0; s[i] != '\0'; i++)

{

s[i] = s[i] - 'a' + 'A';

}

}

int main(void)

{

char str[SIZE];

printf("문자열을 입력하시오: ");

gets(str);

str\_upper(str);

printf("변환된 문자열: %s\n", str);

return 0;

}



* toupper() 함수 이용

#include <string.h>

#include <stdio.h>

#define SIZE 100

int main(void)

{

char str[SIZE];

printf("문자열을 입력하시오: ");

gets(str);

for (int i = 0; str[i] != '\0'; i++)

{

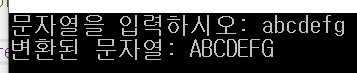
str[i] = toupper(str[i]);

}

printf("변환된 문자열: %s\n", str);

return 0;

}



**[과제#2] 문자열을 이용한 사칙연산 프로그램**

* 코드 및 실행화면

// 문자열을 이용한 사칙연산 프로그램

#include <stdio.h> // 표준 입출력 헤더 파일 선언

#include <math.h> // 여러 수학 함수들을 포함하는 C 언어 표준 라이브러리

/\* double pow(double x, double y)함수를 이용하기 위해 선언하였다.\*/

#include <stdlib.h> // 표준 라이브러리 헤더 파일 선언

/\* 문자열을 수치로 변환하는 전용함수를 사용하기 위해 선언하였다.\*/

#include <string.h> // 문자열 처리 라이브러리

/\* 복사, 결합, 토큰화, 검색과 같은 다양한 명령이 지원되는 문자열 표준 라이브러리이다.

표준 라이브러리는 문자열이 NULL('\0')로 끝난다는 규칙을 사용하여 n개의 문자의 문자열은 n+1 요소의 배열로 표현된다.

문자열 처리 라이브러리에는 strlen(s), strcpy(s1,s2), strcat(s1,s2), strchr(s,c) 등의 함수가 정의되어 있다.\*/

#define SIZE 100 // 배열의 크기를 기호상수로 선언

int main(void) // 반환값이 정수형이며 매개변수가 없는 메인 함수의 선언부

{

// 문자열 선언

char str[SIZE]; // 크기가 100인 문자열, 사용자로부터 입력받은 연산을 저장

char op[SIZE]; // 크기가 100인 문자열, 사용자로부터 입력받은 연산에서 연산자를 분리해 저장

// 문자형 포인터 선언

char\* token; // 토큰을 나누어 저장

// 변수 선언 및 초기화

double x, y, result = 0; // 피연산자와 연산 결과를 저장할 변수를 선언하고 초기화

// 프로그램 안내

printf("\*\*\*연산 프로그램\*\*\*\n\n"); // 프로그램 이름 출력

printf("연산을 다음과 같은 형태로 입력하시오\n"); // 프로그램 이용 안내

printf("add(sub, mul, div, mod, pow) 1 2\n\n"); // 사용 가능한 연산 및 입력 형태 안내

gets(str); // 문자열 입력받기

/\* 한 줄의 문자열을 읽어서 문자 배열에 저장한다.

gets()함수는 비표준 함수이며 배열 크기와 상관없이 한 줄을 저장하기 때문에 안전하지 않다는 의견도 있으나 Dev C++에서 gets\_s 사용시 오류가 발생한다.\*/

// 토큰 받기

token = strtok(str, " ,\t\n"); // ""안의 구분자를 이용해 문자열의 토큰을 구분하여 포인터에 저장

strcpy(op, token); // 저장한 문자열을 op문자열에 저장

/\* strtok(s, delimit)는 문자열을 구분자를 이용하여 단어들로 분리하는 함수이며 string.h에 저장되어 있다.

문자열로부터 한 번에 하나의 토큰을 가져오기 때문에 전체 문자열에서 여러개의 토큰을 가져오기 위해서는 여러번 반복해야 한다.

사용빈도가 높은 편이며 위의 코드에서 구분자는 공백, 쉼표, 탭, 개행문자이다.

사용자가 문자열을 add 1 2를 입력하면 공백을 통해 단어가 구분되고 add, 1, 2처럼 입력하면 쉼표를 통해 단어가 구분된다.

strcpy(s1, s2) 함수는 두번째 문자열을 첫번째 문자열에 복사하는 함수이다. 순서를 헷갈리지 않도록 주의하여 사용한다.\*/

// 다음 토큰 받기

token = strtok(NULL, " ,\t\n"); // 함수를 이용하여 ""안의 구분자를 이용해 토큰을 받아 저장

x = atof(token); // 저장된 토큰을 실수로 변환하여 실수형 변수 x에 저장

/\* atof()는 문자열을 수치로 변환하는 전용 함수이며 scanf()보다 크기가 작다.

stdlib.h에 원형이 정의되어 있어 반드시 포함해야 하며 문자열을 int형으로 변환할 때는 int atoi(const char\*str),

long형으로 변환할 때는 long atol(const char\*str),double형으로 변환할 때는 double atof(const char\*str)형태로 사용한다.

파이썬에서는 int(str)과 같은 코드를 통해 같은 기능을 구현할 수 있다.\*/

token = strtok(NULL, " ,\t\n"); // 함수를 이용하여 ""안의 구분자를 이용해 토큰을 받아 저장

y = atof(token); // 저장된 토큰을 실수로 변환하여 실수형 변수 y에 저장

/\* 매개변수로 들어온 문자열을 앞에서부터 읽어 공백이나 숫자가 아닌 문자가 올 때까지 숫자로 변환해 주는 함수이므로 여기서는 첫번째 숫자 다음 공백 또는 구분자가 나타나 변환이 종료된다.

atoi 함수가 표현할 수 있는 최댓값과 최솟값이 정해져 있기 때문에 오버플로우와 언더플로우를 주의해야 하며 받은 만약 문자열에 숫자가 없는 경우 0을 반환한다.\*/

// 연산자 문자열에 따라 계산 다르게 하기

/\* strcmp(s1, s2)는 문자열 s1과 s2를 비교하여 사전적 순서에서 s1이 앞에 있으면 음수를, 같으면 0을, 뒤에 있으면 양수를 반환한다. \*/

if (strcmp(op, "add") == 0) // 연산자 문자열을 add와 비교했을 때 두 문자열이 같은 경우

{

result = x + y; // add는 더하기이므로 두 값을 더해 결과에 저장

}

if (strcmp(op, "sub") == 0) // 연산자 문자열을 sub와 비교했을 때 두 문자열이 같은 경우

{

result = x - y; // sub는 빼기이므로 첫 번째 수에서 두 번째 수를 빼서 결과에 저장

}

if (strcmp(op, "mul") == 0) // 연산자 문자열을 mul과 비교했을 때 두 문자열이 같은 경우

{

result = x \* y; // mul은 곱하기이므로 두 수를 곱하여 결과에 저장

}

if (strcmp(op, "div") == 0 || strcmp(op, "mod") == 0) // 연산자 문자열을 div 혹은 mod와 비교했을 때 같은 경우

// 나누기와 나머지 연산 모두 나누는 수에 0이 위치할 수 없기 때문에 코드 중복을 막기 위해 ||을 사용

{

if (y == 0) // 나누는 수가 0이 될 때

{

printf("\n분모는 0이될 수 없습니다.\n"); // 오류메시지 출력

printf("분모를 다시 입력해주세요. "); // 나누는 수를 다시 입력받음

scanf("%lf", &y); // 실수 형태로 다시 입력받아 저장하기

}

if (strcmp(op, "div") == 0) // 연산자 문자열을 div와 비교했을 때 같은 경우

result = x / y; // x를 y로 나누어 결과에 저장

if (strcmp(op, "mod") == 0) // 연산자 문자열을 mod와 비교했을 때 같은 경우

result = (int)x % (int)y; // 나머지 연산을 실시하여 결과에 저장

/\* 나머지 연산은 정수로만 가능하므로 형변환 연산자를 이용하여 두 실수를 정수로 변환(강제 형변환)한다.

이 때 둘 중 하나의 실수만 변환하여도 오류가 발생하므로 둘 다 정수로 변환해야 한다.

형변환과 관련하여 만약 두 개의 정수 중 하나를 실수로 변환하여 나눗셈 연산을 실시하는 경우 승급변환(자동 변환)이 일어나 실수끼리의 연산이 이루어진다.\*/

}

if (strcmp(op, "pow") == 0) // 연산자 문자열을 pow와 비교했을 때 같은 경우

{

result = pow(x, y); // x의 y승을 결과에 저장

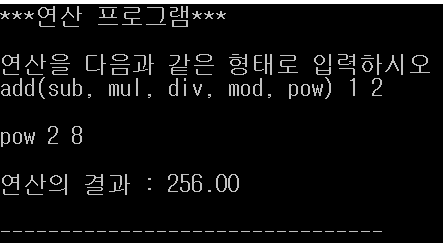
/\* pow()함수는 math.h 헤더 파일에 정의되어 있는 함수이며 반환값은 실수이다.\*/

}

printf("\n연산의 결과 : %.2lf\n", result); // 연산 결과 소수점 두번째자리까지 출력하기

return 0; // 0을 반환하여 끝내기

}



* 기본 프로그램 및 실행화면

// 문자열을 이용한 기본 사칙연산 프로그램

// 피연산자를 정수형으로 변환

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <math.h>

#define SIZE 100

int main(void)

{

char str[SIZE];

char op[SIZE];

char\* token;

int x, y, result = 0;

printf("연산을 입력하시오:");

gets(str);

token = strtok(str, " ,\t\n");

strcpy(op, token);

token = strtok(NULL, " ,\t\n");

x = atoi(token);

token = strtok(NULL, " ,\t\n");

y = atoi(token);

if (strcmp(op, "add") == 0)

{

result = x + y;

}

if (strcmp(op, "sub") == 0)

{

result = x - y;

}

if (strcmp(op, "mul") == 0)

{

result = x \* y;

}

if (strcmp(op, "div") == 0)

{

if (y == 0)

{

printf("\n분모는 0이될 수 없습니다.\n");

printf("분모를 다시 입력해주세요. ");

scanf("%d", &y);

}

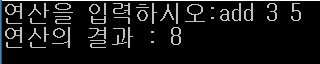
result = x / y;

}

printf("연산의 결과 : %d\n", result);

return 0;

}



**[과제#3] 소설 이어쓰기 프로그램**

* 코드 및 실행화면

// 소설 이어쓰기 프로그램

#include <stdio.h> // 표준 입출력 헤더 파일 포함 선언

#include <string.h> // 문자열 처리 라이브러리

/\* string.h는 C언어의 표준 라이브러리로 메모리 블록이나 문자열을 다룰 수 있는 함수들(복사, 합치기, 비교, 검색 등)을 포함하고 있다.

유니코드 문자열의 경우 wchar.h를 사용하며 본 프로그램에서 사용되는 함수는 strcpy(), strcmp(), strcpy()이다.\*/

#define SIZE 1000

int main(void)

{

// 변수 선언

char str1[SIZE]; // 사용자로부터 문자열을 입력받아 저장할 배열

char str2[SIZE]; // 사용자로부터 입력받은 문자열을 복사하여 저장할 배열

char\* book; // 입력받은 문자열을 이어붙여 저장할 포인터 변수

/\* 기본 알고리즘은 다음과 같다.

사용자로부터 문자열을 입력받는다. 입력받은 문자열을 이전에 입력받은 문자열과 이어붙인다. 이어붙인 문자열을 출력한다. 이어붙인 문자열을 저장한다.

이 때 사용자로부터 나중에 입력받은 문자열은 가장 뒤에 위치해야 하므로 인수를 줄 때 순서에 유의해야 한다.

또한 사용자가 입력한 문장을 누적해서 이어붙이기 위해서는 이어붙인 문장을 복사하여 저장하고 거기에 다시 문장을 이어붙이는 알고리즘을 작성해야 한다.\*/

// 반복문 이용하여 소설 쓰기

while (1) // 항상 반복(1은 True를 의미)

{

printf("문자열을 입력하세요 : "); // 문자열 입력 안내

gets(str1); // 문자열 입력받아 str1에 저장하기

if (strcmp(str1, "끝") == 0) // 만약 사용자가 "끝"을 입력하면

break; // 반복문을 탈출하여 프로그램을 끝내기

/\* strcmp()는 문자열 두개를 인수로 받아 첫번째 문자열과 두번째 문자열을 비교하여 사전 순서에 따라 값을 반환한다.

첫번째 문자열이 앞에 있으면 음수를, 두번째 문자열이 앞에 있으면 양수를 반환한다. 또한 두 문자열의 사전 순서가 같다면 0을 반환하는데,

위의 코드는 이 점을 이용하여 작성된 코드이다.

string.h에는 memcmp, strcoll, strncmp 등의 비교 함수가 포함되어 있다.\*/

book = strcat(str2, str1); // 첫번째 문자열에 두번째 문자열 이어붙여서 포인터에 저장하기

/\* 여기서 포인터가 아닌 배열을 사용하면 배열 요소를 주지 않는 것이 되므로 오류가 발생한다.

C에서는 문자열을 이어붙일 때 strcat()함수를 이용해야 하지만 파이썬 등에서는 문자열을 +를 이용해 이어붙이기도 한다.

문자열을 합치는 함수는 strcat외에 strncat이 있는데 이 함수 또한 string.h에 저장되어 있으며 두 개의 문자열을 지정한 부분만큼 합치는 함수이다.\*/

printf("현재까지의 줄거리 : %s \n\n\n", book); // 이어붙인 문자열을 출력

strcpy(str2, book); // 이어붙인 문자열을 복사하여 저장

/\* strcpy함수는 두번째 문자열을 첫번째 문자열에 복사하는 함수이다. 순서를 헷갈리지 않도록 주의해야 한다.

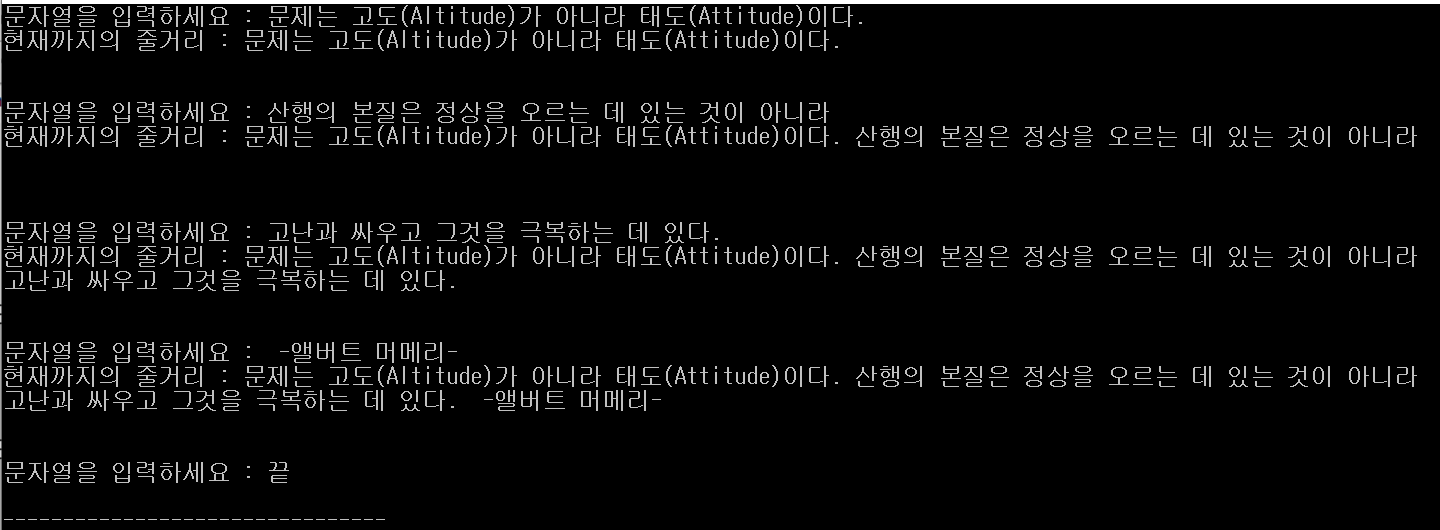
인수 외에 반환값을 저장하기 위한 변수가 필요하지 않은 것으로 보아, 반환값이 없는 것으로 예측된다.

strcpy외에 strncpy함수 또한 string.h에 정의되어 있으며 이 함수는 문자열에서 지정한 부분만큼을 복사하는 함수이다.\*/

}

return 0; // 0을 반환하여 끝내기

}



* 응용프로그램 : 가사 쓰고 가사에서 단어 찾는 프로그램

// 가사 쓰기 프로그램

#include <stdio.h>

#include <string.h>

#define SIZE 1000

int main(void)

{

char str1[SIZE];

 char str2[SIZE] = { 0 }; // 초기화하지 않았을 때 특수문자가 출력돼 초기화함

char\* book;

char s[10]; // 가사 입력받아 저장할 문자열

char\* p; // 가사에서 단어 찾아 저장할 포인터 변수

while (1)

{

printf("가사를 입력하세요 : ");

gets(str1);

if (strcmp(str1, "끝") == 0)

break;

book = strcat(str2, str1);

strcpy(str2, book);

printf("현재까지의 가사 : %s \n\n\n", book);

}

printf("\n\n검색할 단어를 입력하세요 : ");

gets(s);

p = strstr(book, s); // 문자열 book에서 문자열 s찾아서 포인터에 저장

printf("\n"); // 줄바꿈

if (p != NULL) // 만약 단어가 가사에 있다면

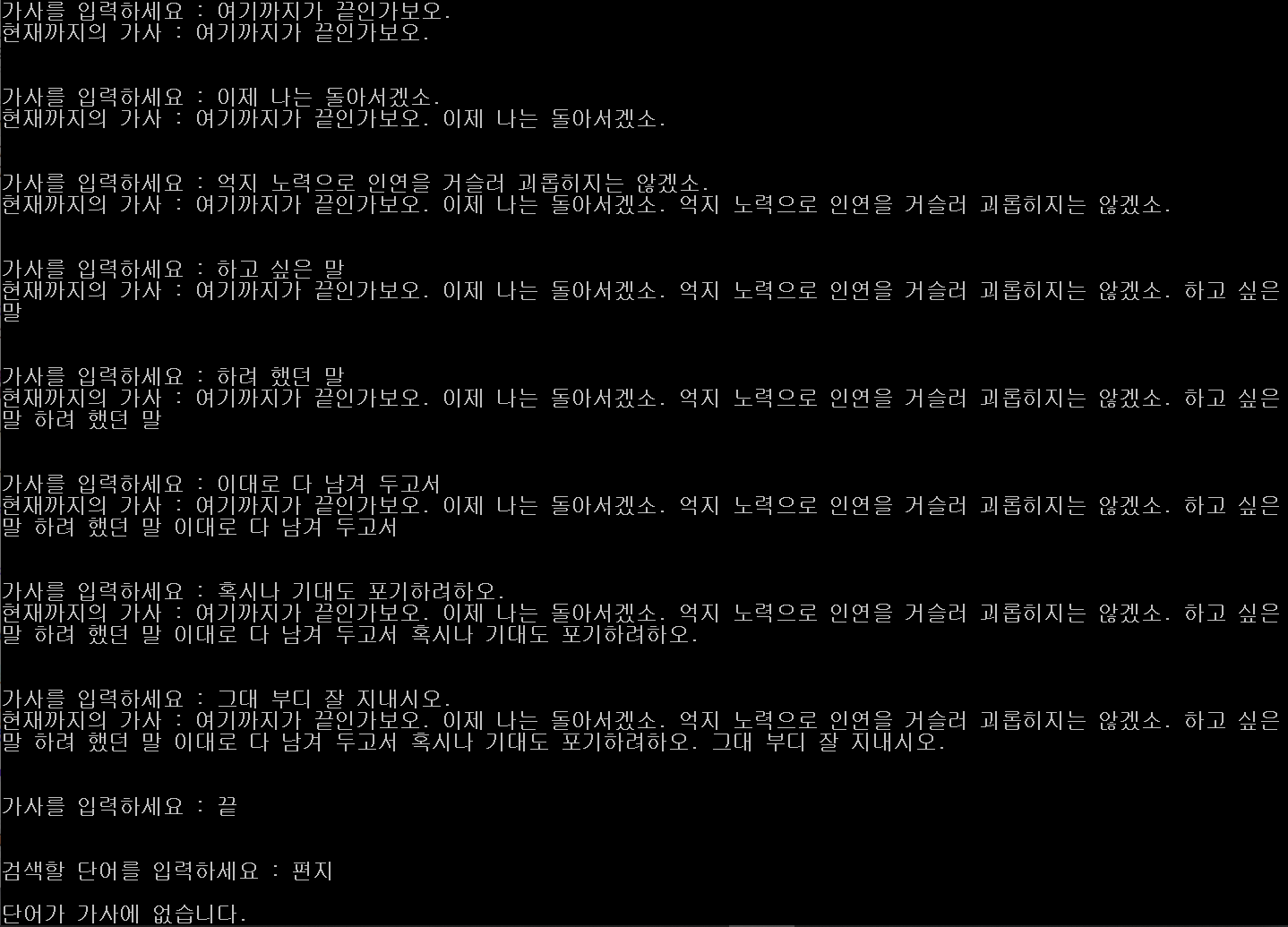
printf("%s", p); // 단어를 출력

else // 만약 단어가 가사에 없다면

printf("단어가 가사에 없습니다."); // 안내 메시지 출력

return 0;

}



* 응용 프로그램2 : 가사 쓰고 가사에서 단어 찾아서 몇 번 등장하는지 알려주는 프로그램

// 가사 쓰기고 특정 단어 등장 횟수 알려주는 프로그램

#include <stdio.h>

#include <string.h>

#define SIZE 1000

int main(void)

{

char str1[SIZE];

char str2[SIZE] = { 0 }; // 초기화하지 않았을 때 오류가 발생해 초기화해줌

char\* book;

char s[10];

char\* ptr; // 단어 저장할 포인터 변수

int i; // 단어 횟수 저장할 변수

while (1)

{

printf("가사를 입력하세요 : ");

gets(str1);

if (strcmp(str1, "끝") == 0)

break;

book = strcat(str2, str1);

strcpy(str2, book);

printf("현재까지의 가사 : %s \n\n\n", book);

}

printf("\n\n검색할 단어를 입력하세요 : ");

gets(s);

ptr = strstr(book, s); // 단어를 찾아서 저장

while (ptr != NULL) // 단어가 있는 동안 반복

{

i++; // 단어가 있는 상태로 시작하므로 i를 먼저 증가시켜줌

ptr = strstr(ptr + 1, s); // 포인터위치 1증가시켜 다시 찾기

}

printf("%d 번 등장", i); // 등장 횟수 출력하기

return 0;

}

