12201922 이규민

5. Homework

1) [A char constant is an ascii number. A string constant is an address where it is stored in the string area.] Explain the result for following code.

#include <stdio.h>

#include <string.h> // you need this header file for string functions

void main(){

char x, y;

x='a'; y=97;

printf("%d %c %d %c\n", x, x, y, y);

char \* x1 = "hello";

printf("%s %p %s %p\n", x1, x1, "hello", "hello"); // use %p for address

}

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

캐릭터 상수는 아스키코드를 저장하기 때문에 따옴표를 사용하거나 아스키코드를 사용해 문자를 저장할 수 있다. ‘a’의 아스키 코드는 97이므로 위 코드의 x와 y는 같은 내용이 저장된 것을 볼 수 있다.

스트링 상수는 스트링이 저장되어있는 주소를 저장한다. X1의 경우 쌍따옴표를 사용하여 스트링의 주소를 저장했다. 위 결과 사진을 보면 x1의 내용과 주소는 스트링 “hello”의 내용과 주소와 동일한 것을 볼 수 있다.

Printf 에서 %d는 아스키 코드를 그대로 출력하고, %c는 문자로 전환하여 출력한다. %s는 스트링을 출력하고, %p는 스트링이 저장된 주소를 출력한다.

2) [A char constant is an ascii number] Try following code and explain the result.

char x[10]; // x is a character array with 10 rooms

x[0]=’a’; x[1]=’b’; x[2]=’c’; x[3]=’d’; x[4]=’e’;

x[5]=’f’; x[6]=’g’; x[7]=’h’; x[8]=’i’; x[9]=’j’;

int i;

for(i=0;i<10;i++){

printf(“%d %c \n“, x[i], x[i]); // print each character with its ascii number

}

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

X 배열안에 따옴표를 이용해서 문자의 아스키코드를 하나씩 저장해주었다. X를 아스키코드와 문자로 출력해보면 그 결과는 위 사진과 같다.

3) Try below. Compare the result with that of Problem 2).

char x[10]; // x is a character array with 10 rooms

int i;

for(i=0;i<10;i++){

x[i]=i+97;

}

for(i=0;i<10;i++){

printf(“%d %c \n“, x[i], x[i]); // print each character with its ascii number

}

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

2번 문제에서는 따옴표를 이용해 문자로 아스키코드를 저장했지만, 3번 문제에서는 아스키코드(숫자)로 저장을 했다. 사실상 동일하게 아스키코드로 저장했다고도 볼 수 있다.

4) Declare a character array with 128 rooms. Store 0 to 127 in this array and print the corresponding character for each ascii code in the array. Find ASCII table in the Internet and confirm the results.

char x[128];

for(i=0;i<128;i++){

x[i]=i;

}

for(i=0;i<128;i++){

printf(“%d%c%d\n”, x[i], x[i], x[i]);

}

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명 스크린샷, 텍스트, 예술, 패턴이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 평행, 번호이(가) 표시된 사진

자동 생성된 설명

0번부터 127번까지의 아스키코드를 출력해보았고, 아스키코드(10진수)와 문자를 출력해보았다. 영문자, 특수문자, 공백문자 등 아스키코드표와 동일하게 출력되는 것을 볼 수 있다.

5) [strlen] Read a string and display its length.

Enter a string

hello

The length is 5

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

scanf를 통해 입력 받은 내용을 배열 x가 가리키는 공간에 저장할 수 있도록 만들었다. Strlen 함수를 이용해서 x의 길이를 출력할 수 있다.

6) [A string is a char array ending with 0] Read a string and display each character in different lines.

Enter a string

hello

h

e

l

l

o

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

scanf를 통해 입력 받은 내용을 배열 x가 가리키는 공간에 저장해줬고, for문을 스트링의 길이만큼 돌려서 한 글자씩 출력해줬다.

6-1) [A string is a char array ending with 0] Try below and explain the result. Use g++ to compile.

char x[10];

strcpy(x, "hello");

strcpy(x, "hi");

for(int i=0;i<10;i++){

printf("%d ", x[i]);

}

텍스트, 폰트, 스크린샷, 그래픽이(가) 표시된 사진

자동 생성된 설명

문자를 아스키코드로 출력해주는 코드다. 처음에는 “hello”라는 문자를 저장해줬기 때문에 배열 x에는 104 101 108 108 111 0으로 저장이 되어있었으나 “hi”라는 문자를 저장해줬기 때문에 104 105 0 108 111 0으로 저장되었다. 메시지 입력 끝은 0이기 때문에 104 105 뒤에도 0으로 바뀐 것을 알 수 있다.

7) [strlen, strcmp] Write a program that keeps reading a string, displaying its length, and checking whether it is "hello". If the input string is "hello", the program stops.

Enter a string

hi

You entered hi. length=2

No it is not hello

Enter a string

hello

You entered hello. length=5

Yes it is hello. Bye.

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

입력 받은 문자가 hello 라면 멈추고, 아니라면 계속 물어보는 코드를 만들어야한다.

무한으로 도는 for 문 안에서 Scanf를 통해 입력을 받고, strlen을 통해 스트링의 길이를 출력하도록 만들었다. For문 안에서 입력 받은 스트링이 “hello”인지 확인하고, 맞다면 for 문을 나가도록 만들었다. “hello”인지 확인을 위해 strcmp 함수를 사용했다. 이후 위 사진처럼 정상적인 결과가 나오는 것을 볼 수 있다.

8) [strcpy] Read a string and copy it to three other string variables and change the first letter of them to 'a', 'b', and 'c', respectively, and display them.

Enter a string

hello

After copying and changing the first letter

aello bello cello

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명 텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

배열 x, y ,z를 만든다. Scanf를 통해 입력받은 스트링을 x의 주소에 저장한다. 그리고 strcpy로 x를 y와 z에 복사한다. 그러면 x, y, z는 모두 “hello”라는 내용을 가진다. X[0] =97로 string의 첫 글자를 ‘a’의 아스키코드로 저장해줬다. Y와 z에도 동일하게 진행해줬고, 정상적으로 결과가 나온 것을 볼 수 있다.

9) [string constant] A string constant such as "hello" is an address. Explain the result of following code.

char \*x, \*y, \*z;

x="hello"; y="hi"; z="bye";

printf("%s %s %s\n", x, y, z);

printf("%p %p %p\n", x, y, z);

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

위 코드에서는 x,y,z 포인터를 선언하였다. x="hello"는 “hello”라는 스트링의 주소 값을 x에 저장하라는 의미다. 그렇게 스트링의 주소를 x,y,z에 각각 저장해줬고, printf의 %s와 %p를 통해 스트링과 스트링의 주소를 출력해줬다.

10) [string constant is an address] Try below and explain why we have an error.

char x[20];

strcpy(x, "hello"); // this is ok

x="hello"; // this is an error. "hello" is an address and we can't store address in

// x which is not a pointer variable

배열 x에 hello 라는 스트링을 저장하는 strcpy는 사용할 수 있다. 그러나 “hello”는 스트링의 주소이므로 x=”hello”를 사용할 수 없다.

11) [You need memory space for strcpy] Try below and explain why we have an error. How can you fix it?

char \*y;

y="hello1"; // this is ok

strcpy(y, "hello2"); // error because y has no space for "hello2"

텍스트, 폰트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명

포인터는 주소를 가리키는 것이므로 y 안에 hello2 라는 내용을 넣을 수 없다. 그러므로 strcpy 사용 시 에러가 발생한다.

Y=”hello2”라고 수정하면 문제없다.

12) [You need memory space for scanf] Try below and explain why you have an error. Fix it.

char \*y;

printf("enter a string\n");

scanf("%s", y); // error becuase y has no space for a string

printf("you entered %s\n", y);

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명 텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

y는 포인터이므로 주소를 입력 받아야하지만, scanf를 통해 스트링을 입력받으면 에러가 발생한다. 입력 값을 포인터에 저장하기 위해서 메모리 공간을 따로 마련해야한다. Malloc 함수로 메모리를 할당한 후에는 정상적으로 작동하는 것을 볼 수 있다.

13) [char pointer array] Define a character pointer array and store/display strings as below.

char \* x[10];

x[0]="hi"; x[1]="bye"; x[2]="hello"; // store addresses of string constants

x[3]=new char[50];

strcpy(x[3], "there"); // copy a string

x[4]=new char[50];

printf("Enter a string\n");

scanf("%s", x[4]); // read a strign from the user

for(int i=0;i<5;i++)

printf("%s \n", x[i]);

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

X를 포인터 배열로 선언한 후 x[0]=”hi”로 스트링 주소를 저장할 수 있다. 포인터는 스트링을 저장할 수 없기 때문에 strcpy와 scanf 사용을 할 수 없다. 하지만 new 함수로 메모리 공간을 마련한다면 사용할 수 있다. 위 사진처럼 정상적으로 작동하는 것을 볼 수 있다.

14) [char pointer array, strcmp, new] Write a program that keeps reading strings and store them in a character pointer array. It stops when the user enters "end" and displays all strings entered so far. Use "new" to allocate memory and use g++ to compile.

Enter a string

hi

Enter a string

aaa

Enter a string

bbb

Enter a string

end

Strings entered so far are

hi aaa bbb

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

무한 for문 안에서 new char를 통해 매번 포인터 배열 x에 메모리를 할당해줬다. 그리고 scanf를 통해 매번 입력을 받고, strcmp를 통해 입력받은 메시지가 “end”인지 확인을 해준다. 만약 “end”라면 무한 for문을 멈추고, x배열에 저장되어있던 스트링을 모두 출력하게만들었다. 위 사진 처럼 정상 작동하는 것을 볼 수 있다.

15) [gets, fgets] Read the same sentence with gets() and fgets() and explain the difference. (Ignore warning for gets. It is a security warning because gets can cause security problem.)

char x[100];

printf("enter a sentence\n");

gets(x);

int slen=strlen(x);

printf("sentence length after gets:%d\n", slen);

for(i=0;i<slen;i++){

printf("%x ", x[i]);

}

printf("\nenter the same sentence\n");

fgets(x, 99, stdin); // read max 99 char’s.

slen=strlen(x);

printf("sentence length after fgets:%d\n", slen);

for(i=0;i<slen;i++){

printf("%x ", x[i]);

}

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

스트링 입력을 받으면 gets는 마지막 \n를 \0으로 바꿔주기 때문에 버퍼에는 \n이 없다. 하지만 Fgets는 \n과 \0을 모두 저장해준다.

위 코드와 같이 스트링의 길이를 확인해보면 fgets는 길이 값이 1만큼 큰 것을 볼 수 있다.

Printf로 문자를 16진수 형태로 출력해보면 fgets로 입력 받은 스트링의 경우 a인 것을 볼 수 있다. 아스키코드표를 비교해보면 a는 LF(개행문자)임을 알 수 있다.

16) [strtok] Use strtok to extract words from a sentence and store them in an array. Display the number of words as below. Note that you need to copy the sentence to another string variable before doing strtok because strtok will destroy the original sentence.

algorithm:

read a line

tokenize

display tokens

Enter a sentence

aa bcd e e ff aa bcd bcd hijk lmn al bcd

You entered aa bcd e e ff aa bcd bcd hijk lmn al bcd

There were 12 words:

aa

bcd

e

e

ff

aa

bcd

bcd

hijk

lmn

al

bcd

The original sentence was: aa bcd e e ff aa bcd bcd hijk lmn al bcd

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

strcpy로 y를 x에 복사해줬다. 그리고 strtok를 이용하여 y의 스트링에서 공백을 찾아 단어를 나눴고, for문으로 그 단어들을 출력해줬다. 그리고 마지막에는 배열 x의 내용을 출력해줬다.

여기서 처음에 y를 x에 복사해준 이유는 strtok 때문이다. Strtok 사용 시 공백인 글자는 \0(문자 종료)로 바뀌기 때문에 단어를 나누는 명령이 끝나고 y의 스트링을 출력해보면 첫 단어 aa만 출력되기 때문이다.

17) [char pointer array, new, strcmp] Write a program that keeps reading a name and stores it in a character pointer array until the user enters bye. The program should display all names after it sees "bye".

Enter a name

kim han kook

Enter a name

park dong il

Enter a name

hong gil dong

bye

There were 3 names.

The names were

kim han kook

park dong il

hong gil dong

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

여러 개의 이름을 저장할 수 있도록 포인터 배열을 사용하였다. 무한 for 문 안에서 입력을받아야하는데, 포인터에는 문자열을 저장할 수 없으므로 new를 사용하여 메모리를 할당하였다. 이름 입력을 받기 위해 명령어를 사용해야하는데 scanf의 경우 띄어쓰기가 입력되는 경우 첫번째 단어만 입력을 받기 때문에 좋은 선택지는 아니다. 그래서 fgets를 사용하였으나 줄바꿈이 같이 들어오기 때문에 마지막 \n을 \0으로 바꿔주었다. 그리고 strcmp를 사용하여 bye라는 메시지가 입력되었다면 무한 for문을 중단하고 나가도록 만들었다. 이후 입력되었던 모든 이름들이 출력된다.

18) [There is a hidden 0 at the end of a string] Try below and explain why it behaves strange. How can you fix it?

int x3;

char x2[12];

char x1[12];

x1[0]=33;

x3=44;

strcpy(x2,"abcdefghijkl");

printf("%p %p %p %d %d %s", x1, x2, &x3, x1[0], x3, x2);

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

printf에서 5번째 즉 x3의 값이 44가 아닌 0이 출력된다는 문제가 있다. X2가 받을 수 있는 문자 수는 12개인데, strcpy를 통해 12개의 문자를 입력 받았고 문자 종료를 의미하는 0이 붙기 때문에 오버플로우가 발생한다. 그 0이 x3 주소의 값에 들어가기 때문에 이 문제가 발생한다.

고치는 방법은 여러가지지만 가장 쉬운 방법은 char x2[12]를 char x2[13]으로 바꾸는 방법이다. 그러면 오버플로우가 발생하지 않아 x3의 값은 그대로 유지된다.

19) [You need memory space for strcpy] What is wrong with the following program? How can you fix it?

int main(){

char \* strarr[10]={NULL};

strarr[0]=”hello”;

strcpy(strarr[1],”bye”);

printf("%s %s\n", strarr[0], strarr[1]);

}

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 소프트웨어, 폰트이(가) 표시된 사진

자동 생성된 설명

Char \* strarr[10]은 배열을 사용하긴했지만 여러 개의 포인터를 만든 것일 뿐이다. 포인터는 내용이 아닌 주소를 저장하는데, strcpy는 내용을 저장하는 함수이므로 오류가 발생한다. 이 문제를 해결하기 위해서는 new를 사용하여 포인터에 메모리를 할당하는 방법과 strcpy가 아닌 strarr[1]=”bye”를 사용하는 방법이 있다. 후자는 쉬운 방법이니 new를 사용하여 해결해보았다. 정상적으로 출력되는 것을 볼 수 있다.

20) [char pointer array, strtok, strcmp] Write a program that reads a long sentence and

displays the frequency of each word. It also prints the word that has the maximum

frequency. **Your main function should look like below. Implement each function.**

int main(){

// step 1. read a sentence into buf

char buf[100];

get\_sentence(buf);

// step 2. extract words and store in tokens array. return num of tokens

int ntok;

char \*tokens[50];

ntok = tokenize(buf, tokens);

// step 3. show tokens

display\_tokens(tokens, ntok);

// step 4. compute unique tokens into unique\_tokens array

char \*unique\_tokens[50];

int nuniqtok;

nuniqtok=compute\_unique\_tokens(tokens, ntok, unique\_tokens);

printf(“unique tokens are “);

display\_tokens(unique\_tokens, nuniqtok);

// step 5. compute freq of each unique token into freq array

int freq[50];

compute\_freq(tokens, ntok, unique\_tokens, nuniqtok, freq);

// step 6. show frequencies of each token

show\_freq(freq, unique\_tokens, nuniqtok);

// step 7. show max freq word

show\_max\_freq\_word(unique\_tokens, nuniqtok, freq);

}

Enter a sentence

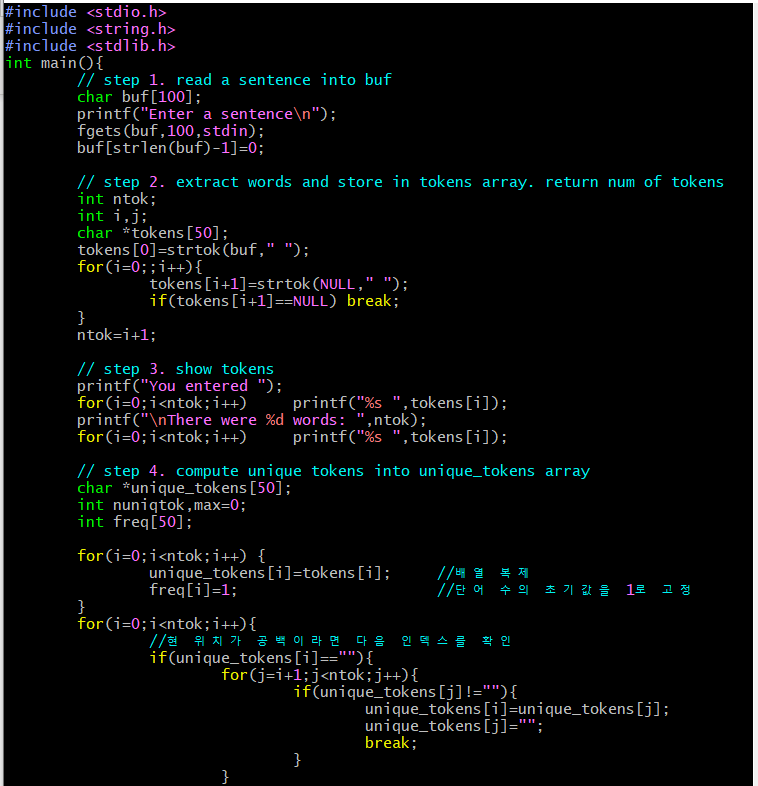
aa bcd e e ff aa bcd bcd hijk lmn al bcd

You entered aa bcd e e ff aa bcd bcd hijk lmn al bcd

There were 12 words: aa bcd e e ff aa bcd bcd hijk lmn al bcd

Frequncies: aa 2 bcd 4 e 2 ff 1 hijk 1 lmm 1 al 1

The word with the max freq: bcd



텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

Step1. 띄어쓰기를 포한한 모든 문자를 받기 위해 fgets를 사용했고, 줄바꿈을 제거하기 위해 \n을 \0으로 바꿨다.

Step2. 입력받은 문자열을 strtok을 사용하여 단어별로 나눴다. 그리고 단어의 개수를 세기 위해서 ntok=i+1이라고해줬다. 그리고 for문 안에서 tokens[i+1]=strtok(NULL," ");처럼 i+1을 사용해야 ntok=i+1 한 문장으로 단어를 셀 수 있다.

Step3. Printf를 사용하여 token 포인터들이 가리키는 내용을 출력해줬다.

Step4. Stpe5. 내가 사용한 방식은 이와 같다. 우선 토큰 배열을 유닉 토큰배열에 동일한 내용으로 복사해준다. 이후 첫번째 위치의 단어가 다른 위치에도 있는지 확인한다. 두번째 토큰부터 보는데 만약 동일한 단어가 있다면 그 토큰은 “” 공백으로 내용을 바꿔주고 그 단어의 개수인 freq[i]를 카운트해준다. 그렇게 마지막까지 동일한 작업을 해준다. 여기까지가 for문에서 i가 0일 때 작업이 완료된 것이다. 이후 두번째 토큰과 동일한 단어가 있는지 찾고 공백으로 바꾸고 이 과정을 끝까지 반복해준다. 여기서 위 작업을 반복 하다 보면 동일한 단어들은 모두 제거가 되어 공백 밖에 남지 않으면 그 때의 I값이 유닉 토큰의 개수이다. 이 작업을 통해 유닉 토큰과, 그 단어의 개수를 알 수 있다.

Step6. Printf를 통해 유닉 토큰을 출력한다. 출력함과 동시에 freq값이 최대일 때를 찾고, 그게 몇 번째 배열인지 확인하고 j 변수에 저장해준다.

Step7. 위 단계에서 찾은 j를 사용하면 가장 많이 입력된 단어를 출력할 수 있다.