

**캣서트 보고서**

202221649 인공지능학과 김민영

4.27 C언어 스터디4

3.

N찍기

|  |
| --- |
| #define \_CRT\_SECURE\_NO\_WARNINGS  #include<stdio.h>  int main()  {  int i, j, st,a ;  scanf("%d", &st);    for (i = 0; i < st; i++) {  printf("\*");  if (i > 1) {  for (a = 0; a < i - 1; a++) {  printf(" ");  }  }  if (i > 0 && i < st) {  printf("\*");  }  for (j = 1; j < (st - i) - 1; j++) {  printf(" ");  }  if (i+1==st)  continue;  printf("\*\n");  }  } |



Wave

|  |
| --- |
| #define \_CRT\_SECURE\_NO\_WARNINGS  #include <stdio.h>  #include <string.h>  int main() {  char arr[100];  int i, length;  scanf("%[^\n]s", arr);  length = strlen(arr);  for (i = 0; i < length; i++) {  if (arr[i] == ' ') {  printf("%s", arr);  }  arr[i] -= 32;  printf("%s ", arr);  arr[i] += 32;  }  } |

텍스트이(가) 표시된 사진

자동 생성된 설명

재귀함수 n!

|  |
| --- |
| #define \_CRT\_SECURE\_NO\_WARNINGS  #include<stdio.h>  int num = 1;  void number(int a) {  num \*= a;  if (a == 1) {  printf("%d", num);  return;  }  number(a - 1);  }  int main() {  int a;  printf("INPUT NUMBER: ");  scanf("%d", &a);  number(a);  } |

텍스트이(가) 표시된 사진

자동 생성된 설명

2진수

|  |
| --- |
| #define \_CRT\_SECURE\_NO\_WARNINGS  #include<stdio.h>  int num[100] = { 0, };  void number(int a, int b) {  num[b] = a % 2;  a = a / 2;  if (a < 2) {  printf("%d", num);  return;  }  number(a, b + 1);  }  int main() {  int a;  int b = 0;  scanf("%d", &a);  number(a,b);  } |

피보나치 수열

|  |
| --- |
| #define \_CRT\_SECURE\_NO\_WARNINGS  #include <stdio.h>  int a = 0;  int sum(int n, int num) {  n += n;  a += 1;  if (a == num)  printf("%d", n);  return;  sum(n,num);    }  int main() {  int num;  scanf("%d", &num);  sum(num);  } |

**#1. memset, strlen, strcmp, strcat, strcpy, strrev, strtok, strstr, atoi, itoa 함수 조사 및 실습**

**1) memset:** 메모리의 내용(값)을 원하는 크기만큼 특정 값으로 세팅할 수 있는 함수

**함수 원형 : void\* memset(void\* ptr, int value, size\_t num);**

Void\*ptr – 세팅하고자 하는 메모리의 시작 주소

Int value – 메모리에 세팅하고자 하는 값

Size\_t num – 길이 (바이트 단위로써 메모리의 크기 한조각 단위의 길이)

헤더 파일 : memory.h / string.h

|  |
| --- |
| #include<string.h>  #include<stdio.h>  int main(void)  {  char arr1[] = "blockdmask blog";  memset(arr1, 'c', 5 \* sizeof(char));  printf(arr1);  return 0;  } |

**2) strlen:** const char\* 타입의 문자열을 받아서 해당 문자열의 길이를 반환하는 함수

**함수원형 : size\_t strlen(const char\* str);**  
size\_t – 객체나 값이 포함할 수 있는 최대 크기의 데이터를 표현하는 데이터 타입

헤더파일 - C언어 : **<string.h>** / C++ : **<cstring>**

|  |
| --- |
| #include <stdio.h>  #include <string.h>  int main()  {  char s1[10] = "cat";  char s2[10] = "cert";    printf("s1 문자열 길이 : %d\n", strlen(s1));  printf("s2 문자열 길이 : %d\n", strlen(s2));  return 0;  } |

**3) strcat:** 문자열A에 원하는 문자열B를 문자열A 뒤쪽에 이어 붙이는 함수

**함수 원형 :** **char\* strcat(char\* dest, const char\* origin);**

**dest :** 복사를 받을 대상의 시작 주소  
**src :** 복사를 할 원본의 시작 주소  
**num :** 복사를 할 문자의 개수  
  
헤더파일: <string.h>    //C++ : <cstring>

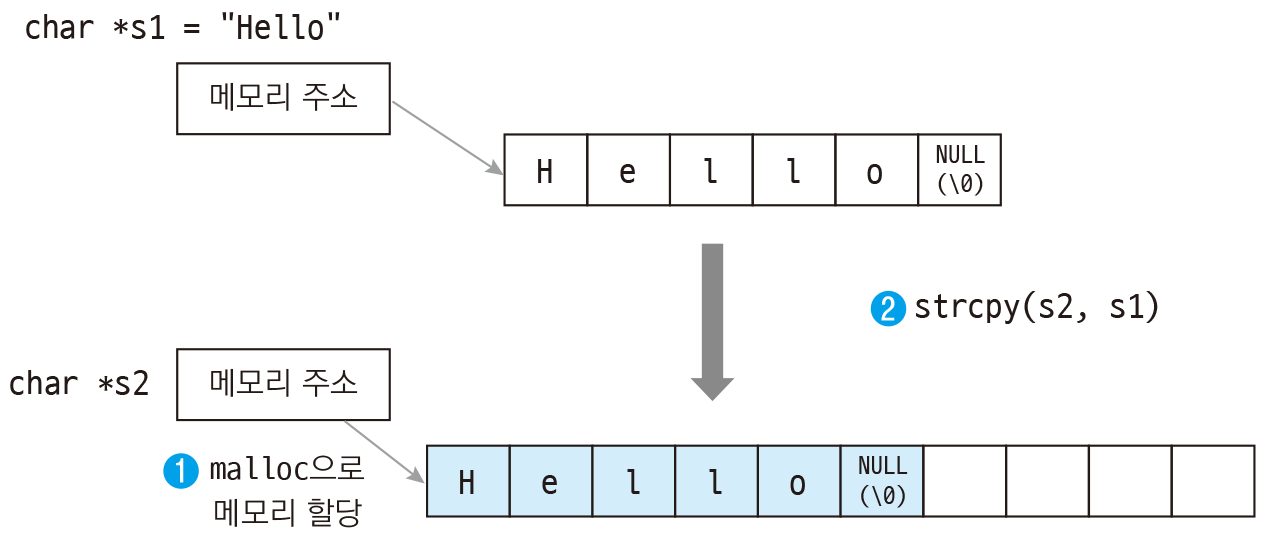
|  |
| --- |
| #include <stdio.h>  int main()  {    char str1[10] = "abc";  char str2[] = "def";  char str3[] = "ghi";    strcat(str1, str2);  printf("strcat : %s\n", str1);  return 0;  } |

**4) strcpy: 대상 문자열 전체를 복사한다**

+ strncpy : 대상 문자열을 일정 길이만큼 복사한다

**함수원형 : char\* strcpy(char\* dest, const char\* src);**

dest : 복사를 받을 대상의 시작 주소  
src : 복사를 할 원본의 시작 주소  
num : 복사를 할 문자의 개수



|  |
| --- |
| #define \_CRT\_SECURE\_NO\_WARNINGS  #include <stdio.h>  #include <string.h>  int main()  {  char\* s1 = "hello";  char\* s2 = malloc(sizeof(char) \* 10);  strcpy(s2, s1);  printf("%s\n", s2);  return 0;  } |

**5) strrev:** 문자열을 거꾸로 뒤집는 함수

매개변수로 받은 문자열을 그대로 뒤집어서 반환하기 때문에 수정할 수 있는 형태의 문자열이여야함.

(문자열리터럴 수정 X)

**함수원형 :** char\*strrev(char\*string);

|  |
| --- |
| #define \_CRT\_SECURE\_NO\_WARNINGS  #include <stdio.h>  int main()  {  char string[] = "Cat cert jjang";  char\* pstr;  pstr = strrev(string);  printf("%s\n", pstr);  printf("%s\n", string);  return 0;  } |

6) strtok: 문자열을 토큰처럼 조각내는 함수

**함수원형 : char\* strtok(char\* str, char\* delimiters);’**

Char\* 타입의 문자열 str을 첫번째 매개변수로 받음

→ 두번째 매개변수로 들어온 char\* 타입의 구분자를 기준으로 문자열을 잘라 문자열의 포인터를 하나씩 반환

|  |
| --- |
| #include<stdio.h>  int main(void)  {  char str[] = "Cat,cert,Kim,min,yeong";  char\* ptr = strtok(str, ",");  while (ptr != NULL)  {  printf("%s\n", ptr);  ptr = strtok(NULL, ",");  }  return 0;  } |

**7) strstr**: 특정 문자열에 원하는 문자열이 포함되어 있는지 확인하는 함수

**함수원형 :** char \*strstr(char \* const \_String, char const \* const \_SubString);

Str1: search 대상 문자열

Str2: 찾고자 하는 문자열

리턴값: 해당문자열로 시작하는 문자열의 위치 포인터

|  |
| --- |
| #include <stdio.h>  #include <string.h>  int main()  {  char s1[30] = "cat cert";  char\* ptr = strstr(s1, "cat");  printf("%s\n", ptr);  return 0;  } |

**8) atoi** = char to int = 문자열을 정수 타입으로

**함수원형 :** int atoi (const char\* cStr);

매개변수로 들어온 문자열을 앞에서부터 읽어서 “공백” or “숫자가 아닌 문자”가 올 떄까지 숫자로 변환해주는 원리

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  int main()  {  char\* s1 = "20220502";  int num1;  num1 = atoi(s1);  printf("%d\n", num1);  return 0;  } |

**9) itoa:** 정수형을 문자열로

integer 값을 2진, 8진, 10진, 16진 문자열

**함수원형 : char** \* **itoa**(**int** val, **char** \* buf, **int** radix)

변환할 정수 / 변환받을 문자열을 저장할 문자열 변수 / 변환할 진수 (10: 10진수)

|  |
| --- |
| #include <stdio.h>  int main()  {  int a = 502;  char str[10] = "2020";  char tmp[10] = { 0, };    strcat(str, itoa(a, tmp, 10));  printf("%s\n", str);  return 0;  } |

**10) strcmp :** **두개의 문자열을 비교 / string1과 string2 두 문자열을 비교함**

**원형 함수 : int strcmp(const char\* str1, const char\* str2)**

첫번째 매개변수 str1 : 비교할 문자열1

두번째 매개변수 str2 : 비교할 문자열2

|  |
| --- |
| #include <stdio.h>  #include <string.h>  int main()  {  char s1[10] = "cat";  char\* s2 = "cat";  int ret = strcmp(s1, s2);  printf("%d\n", ret);  return 0;  } |

**#2 위 문자열 함수 중 리눅스 환경에서 실행되지 않는 함수 조사**

**Strrev 함수 (우분투에도 없음)**

**Itoa 함수 (비 표준함수)**