학번: 2019202052

이름: 김호성

1번

#include <stdio.h> //표준 입출력 헤더파일

char \* ReadUserName(void) //문자열 입력받는 함수

{

char name[30];

printf("What's your name? ");

gets(name);

return name; //받은 값 반환

}

int main(void) //main함수

{

char \* name1;

char \* name2;

name1=ReadUserName();

printf("name1: %s \n", name1);

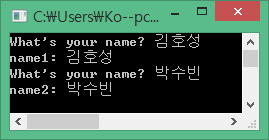
name2=ReadUserName();

printf("name2: %s \n", name2);

return 0; //메인함수 종료

}

실행결과



2번

#include <stdio.h> //표준 입출력 헤더파일

char name[30];

char \* ReadUserName(void) //문자열 입력받는 함수

{

printf("What's your name? ");

gets(name);

return name; //받은 값 반환

}

int main(void) //main함수

{

char \* name1;

char \* name2;

name1=ReadUserName();

printf("name1: %s \n", name1);

name2=ReadUserName();

printf("name2: %s \n", name2);

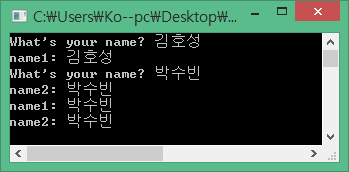
printf("name1: %s \n", name1); //예상과 다른 값 출력

printf("name2: %s \n", name2);

return 0; //main함수종료

}

실행결과



3번

#include <stdio.h> //표준 입출력 헤더파일

#include <stdlib.h> //표준 메모리라이브러리

int main(void)

{

int \* ptr1 = (int \*)malloc(sizeof(int)); //int형 \*1 = 4

int \* ptr2 = (int \*)malloc(sizeof(int)\*7); //int형 \*7 = 28

int i;

\*ptr1 = 20;

for(i=0; i<7; i++)

ptr2[i]=i+1;

printf("%d \n", \*ptr1);

for(i=0; i<7; i++)

printf("%d ", ptr2[i]);

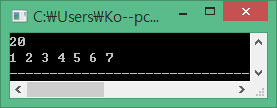
free(ptr1); //메모리 동적할당 한 것을 소멸시킴

free(ptr2);

return 0; //main함수종료

}

실행결과



4번

#include <stdio.h> //표준 입출력 헤더파일

#include <stdlib.h> //표준 메모리라이브러리

char \* ReadUserName(void) //입력받는 함수

{

char \* name = (char \*)malloc(sizeof(char)\*30); //메모리 동적할당(함수 호출 횟수만큼 새롭게 할당)

printf("What's your name? ");

gets(name);

return name;

}

int main(void) //main함수

{

char \* name1;

char \* name2;

name1=ReadUserName();

printf("name1: %s \n", name1);

name2=ReadUserName();

printf("name2: %s \n", name2);

printf("again name1: %s \n", name1);

printf("again name2: %s \n", name2);

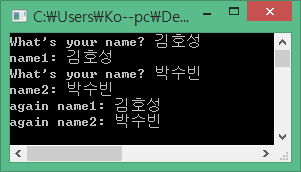
free(name1); //메모리 동적할당 한 것을 소멸시킴

free(name2); //메모리 동적할당 한 것을 소멸시킴

return 0; //main함수 종료

}

실행결과



5번

#include <stdio.h> //표준 입출력 헤더파일

#include <stdlib.h> //표준 메모리라이브러리

char \* ReadUserName() //입력받는 함수

{

char \* name = (char \*)calloc(30, sizeof(char));

printf("What's your name? ");

gets(name);

return name;

}

int main(void)

{

char \* name1;

char \* name2;

name1=ReadUserName();

printf("name1: %s \n", name1);

name2=ReadUserName();

printf("name2: %s \n", name2);

printf("again name1: %s \n", name1);

printf("again name2: %s \n", name2);

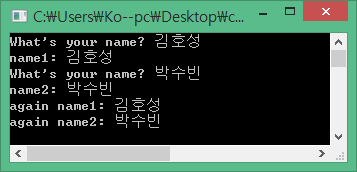
free(name1); //메모리 동적할당 한 것을 소멸시킴

free(name2); //메모리 동적할당 한 것을 소멸시킴

return 0; //main함수 종료

}

실행결과



6번

#include <stdio.h> //표준 입출력 헤더파일

#include <stdlib.h> //표준 메모리라이브러리

#include <string.h> //표준 문자열라이브러리

char \* ReadString(int maxlen) //입력받는 함수

{

char \* str = (char \*)malloc(sizeof(char)\*(maxlen + 1)); //4바이트 크기의 블록 n개를 힙 영역에 할당

printf("문자열 입력: ");

fgets(str, maxlen + 1, stdin); //fgets함수를 통해 str 입력받기

return str; //값 반환

}

int main(void) //main함수

{

int maxlen, len, i; //int형 변수 선언

char \* str; //char형 포인터 변수 str 선언

printf("문자열의 최대 길이 입력: ");

scanf("%d", &maxlen);

getchar(); //\n 문자 지우기

str = ReadString(maxlen); //ReadString함수 호출을 통해 str 입력받기

str[strlen(str) - 1] = 0; //\n 문자 지우기

len = strlen(str); //len에 strlen(str) 저장

for (i = len; i > 0; i--) //i=len부터 i>0일 때까지 i를 1씩 감소시키면서 len번 반복

if (str[i] == ' ') //str[i]가 공백이면 실행

{

printf("%s ", &str[i + 1]); //str[i+1] 출력

str[i] = 0; //str[i]에 0 저장

}

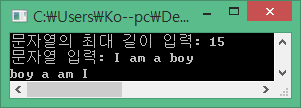
printf("%s", &str[0]); //str[0] 출력

free(str); //할당된 메모리 공간 해제

return 0; //main함수 종료

}

실행결과



7번

#include <stdio.h> //표준 입출력 헤더파일

#include <stdlib.h> //표준 메모리라이브러리

int \* ReadNum(int \* arr, int arrlen, int idx) //입력받는 함수

{

printf("정수 입력: ");

scanf("%d", &arr[idx]);

return arr; //arr의 값 반환

}

int main(void) //main함수

{

int arrlen = 5;

int idx = 0;

int i;

int \* arr = (int\*)malloc(sizeof(int)\*arrlen); //4\*arrlen바이트를 힙 영역에 할당

while (1) //무한반복문

{

arr = ReadNum(arr, arrlen, idx); //ReadNum함수 호출을 통해 arr 입력받기

if (arr[idx] == -1) //arr[idx]가 -1이면

break; //무한반복문 빠져나오기

if (arrlen == idx + 1)//arrlen이 idx+1과 같으면

{

arrlen += 3; //arrlen에 3 더하기(3바이트 늘리기)

arr = (int \*)realloc(arr, sizeof(int)\*arrlen); //다시 4\*arrlen바이트를 힙 영역에 할당

}

idx++; //idx에 1 더하기

}

for (i = 0; i < idx; i++) //i=0부터 i<idx일 때까지 idx번 반복

printf("%d ", arr[i]); //arr[i] 출력

free(arr); //할당된 메모리 공간 해제

return 0; //main함수 종료

}

실행결과

