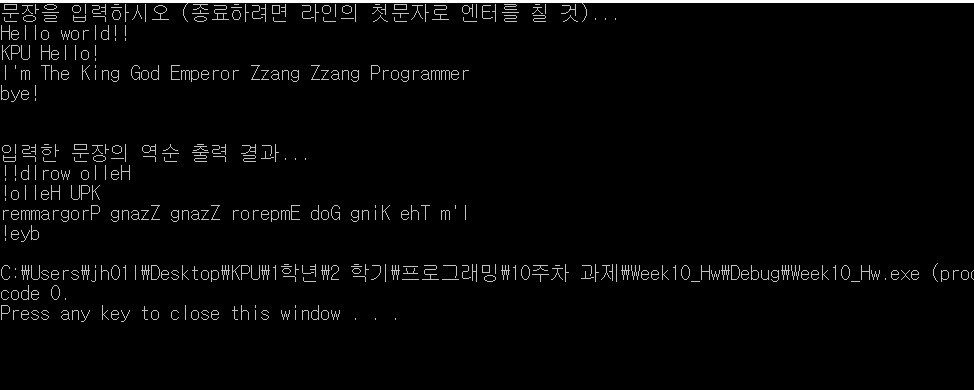
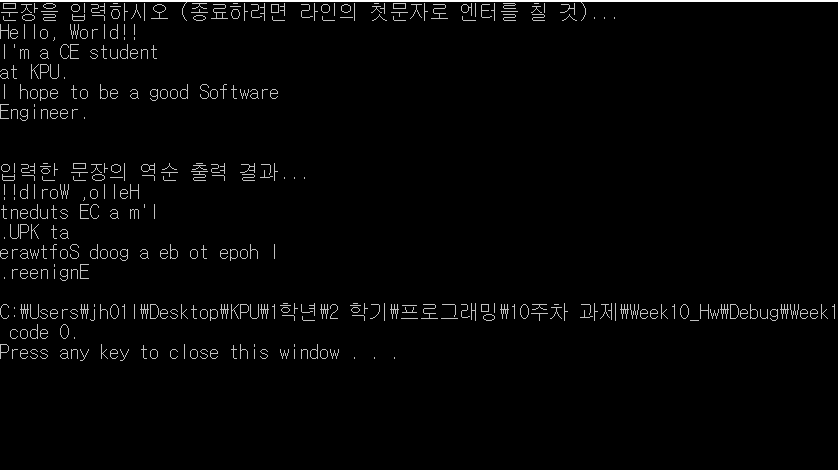
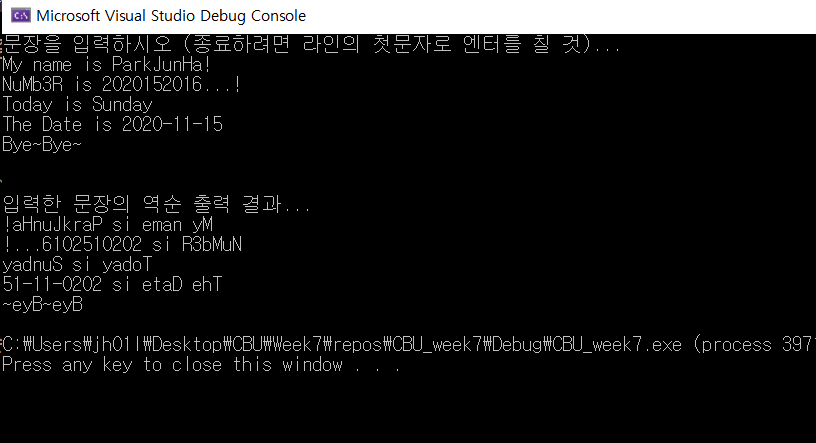
1번 문항

#define \_CRT\_SECURE\_NO\_WARNINGS   
#include <stdio.h>   
#include <stdlib.h>   
#include <string.h>   
   
#define MAX 100   
#define LINES 10   
   
int getline(char\* buffer, int max\_num)   
{   
 fflush(stdin); // 버퍼를 비워주는 역할   
   
 fgets(buffer, max\_num , stdin); // 표준 입력으로 문자열 저장   
 if (strlen(buffer) == 1) { //엔터인 경우 (개행문자만 포함된 상태)   
 buffer[strlen(buffer) - 1] = '\0'; //마지막 개행문자 제거   
 return 0; //0을 반환하여 Main 함수의 반복문을 탈출   
 }   
   
 else {   
 buffer[strlen(buffer) - 1] = '\0';// 그 이외의 경우   
 return 1; //그 이외의 경우 1을 반환함.   
 }   
   
}   
   
void revline(char\* line)   
{   
 int line\_length = strlen(line); //라인의 길이를 저장함.   
 for (int i = 0; i < line\_length / 2; i++) { //라인의 문자열을 바꾸는 기능을 함   
 int temp = line[i];   
 line[i] = line[line\_length - i - 1];   
 line[line\_length - i - 1] = temp;   
 }   
}   
int main(void)   
{   
 char\* line[LINES];   
 char buf[MAX];   
 int num = 0;   
   
 printf("문장을 입력하시오 (종료하려면 라인의 첫문자로 엔터를 칠 것)...\n");   
 while (getline(buf, MAX) != 0) // getline(): 한 라인을 읽어서 buf에 저장함   
 {   
 line[num] = (char \*)malloc(strlen(buf)); // 빈 칸 채우기   
 strcpy(line[num], buf); // 빈 칸 채우기   
   
 //printf("%s\n", line[num]);   
 num++;   
   
 }   
 printf("\n입력한 문장의 역순 출력 결과...\n");   
   
 for (int i = 0; i < num; i++) // 빈 칸 채우기   
 {   
 revline(line[i]); // revline(): 한 라인의 문장을 뒤집음   
 printf("%s\n", line[i]);   
 }   
 return 0;   
}

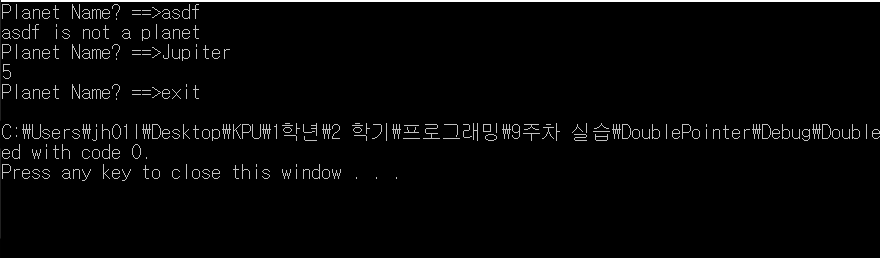


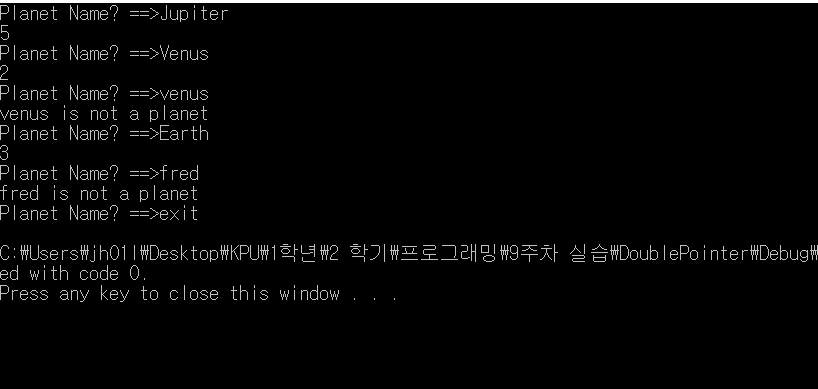


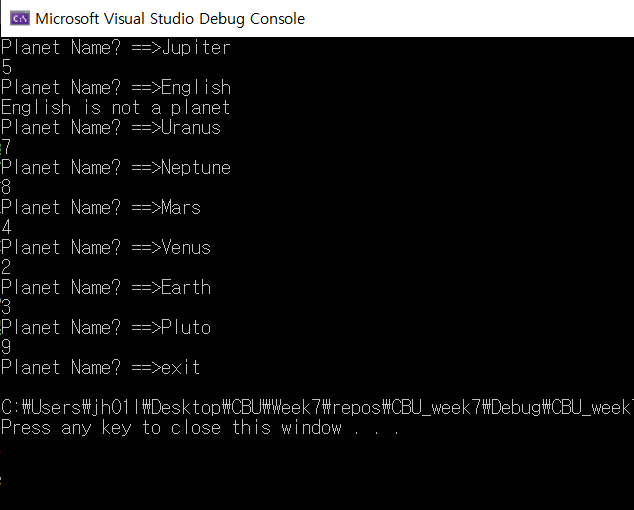


2번 문항

#define \_CRT\_SECURE\_NO\_WARNINGS   
#include <stdio.h>   
#include <string.h>   
int main(void) {   
 char planets[][20] = { //2차원 배열 생성   
 "Mercury", "Venus", "Earth",   
 "Mars", "Jupiter", "Saturn",   
 "Uranus", "Neptune", "Pluto"   
 };   
 int length = sizeof(planets) / sizeof(planets[0]); //배열의 길이   
 char planet\_name[100]; // 입력받을 행성 이름을 저장할 배열   
 int number = 0;   
   
 do {   
 printf("Planet Name? ==>");   
 scanf("%s", &planet\_name); //stdin   
   
 for (int i = 0; i < length; i++) {   
 if (!strcmp(\*(planets + i), planet\_name)) { // 포인터로 배열의 요소에 접근 & strcmp 값이 참이 되는 조건   
 printf("%d\n", i+1); // 행성의 이름을 1부터 출력 (i는 0부터 시작이므로)   
 break; // 내부 for문 탈출   
 }   
   
 if (!strcmp(planet\_name, "exit")) { //exit인 경우 프로그램 종료   
 return 0;   
 }   
   
 if (i == length - 1) //만약 모든 탐색 결과에 부합하지 않는 경우   
 printf("%s is not a planet\n", planet\_name); //아래 문자열 출력   
 }   
   
 } while (1); //무한 루프 생성   
}

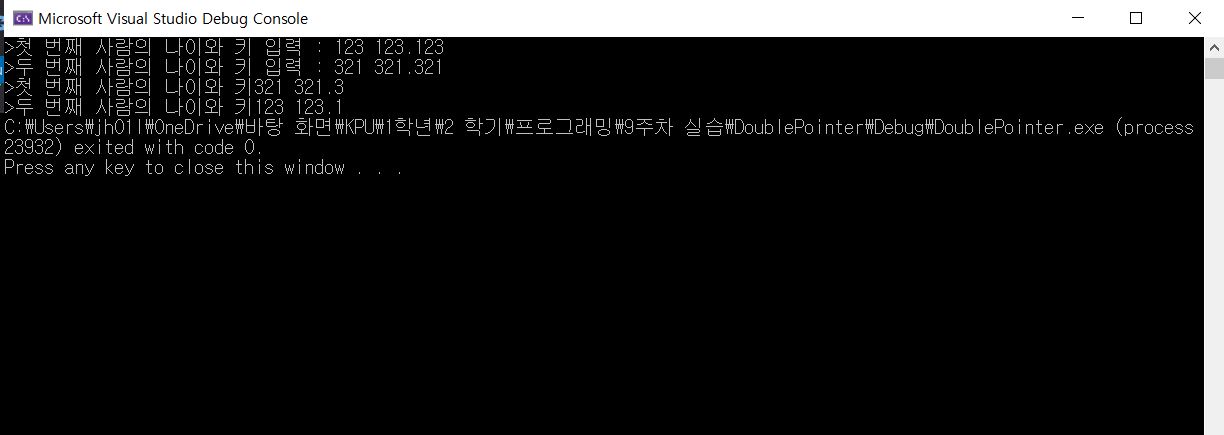


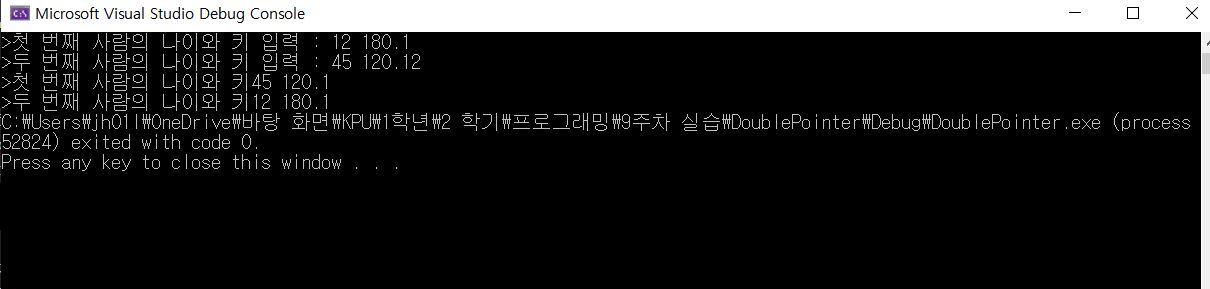


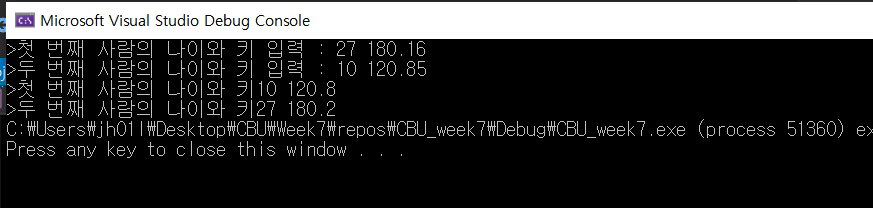


3번 문항

#define \_CRT\_SECURE\_NO\_WARNINGS   
#include <stdio.h>   
#include <string.h>   
   
void swap(char\* type, void \*a, void \*b) { // 함수 포인터 선언   
   
   
 if (strcmp(type, "int") == 0) { // 만약 타입이 int 라면   
 int temp = \*(int \*)a;   
 \*(int \*)a = \*(int\*) b;   
 \*(int \*)b = temp; //swap   
 }   
   
 else { //double 이라면   
 double temp = \*(double \*)a;   
 \*(double \*)a = \*(double \*)b;   
 \*(double \*)b = temp; //swap   
 }   
}   
   
int main(void) {   
 double first\_height, second\_height = 0;   
 int first\_yr, second\_yr = 0; //변수 선언   
   
 printf(">첫 번째 사람의 나이와 키 입력 : ");   
 scanf("%d %lf", &first\_yr, &first\_height); //첫번째 입력받기   
   
 printf(">두 번째 사람의 나이와 키 입력 : ");   
 scanf("%d %lf", &second\_yr, &second\_height); // 두번째 입력받기   
   
 swap("int", &first\_yr, &second\_yr);   
 swap("double", &first\_height, &second\_height);   
 printf(">첫 번째 사람의 나이와 키%d %.1lf\n", first\_yr, first\_height); //출력   
 printf(">두 번째 사람의 나이와 키%d %.1lf", second\_yr, second\_height); //출력   
}





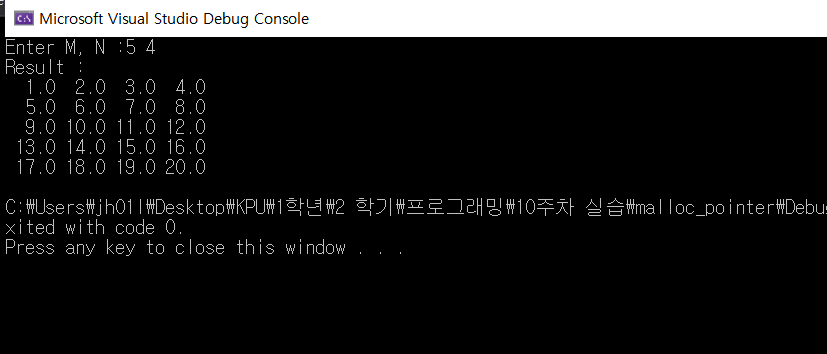


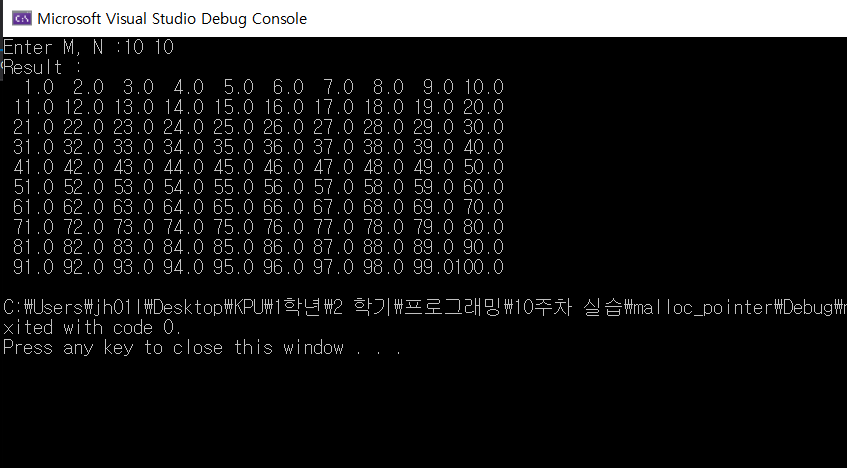
4번 문항

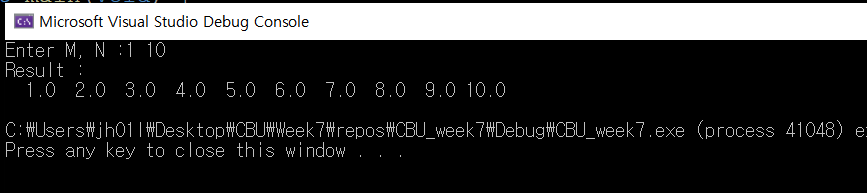
#define \_CRT\_SECURE\_NO\_WARNINGS   
#include <stdio.h>   
#include <stdlib.h>   
void fill\_array(double\*\* M\_array, int M, int N) {   
  
 for (int p = 0; p < M; p++) {

double num = 1;

for (int q = 0; q < N; q++) {   
 M\_array[p][q] = num \* (p + 1);   
 num++; //숫자를 순서대로 배열에 대입   
 }   
 }   
}   
void print\_array(double\*\* M\_array, int M, int N) { // 배열을 순서대로 접근하여 출력하는 함수   
 printf("Result :\n");   
 for (int p = 0; p < M; p++) {   
 for (int q = 0; q < N; q++) {   
 printf("%5.1lf", M\_array[p][q]);   
   
 }   
 printf("\n");   
 }   
}   
   
int main(void) {   
 int M, N = 0;   
 printf("Enter M, N :");   
   
 scanf("%d %d", &M, &N); //stdin   
   
 double\*\* M\_array;   
 M\_array = (double\*\*) malloc(sizeof(double) \* M); //행의 수만큼 배열을 동적할당받음   
 for (int i = 0; i < M; i++) { // 더블포인터의 각 행을 열만큼 동적할당 받음   
 M\_array[i] = (double \*)malloc(sizeof(double) \* N);   
 }   
 fill\_array(M\_array, M, N); //함수 실행   
 print\_array(M\_array, M, N); //함수 실행   
   
 for (int i = 0; i < M; i++) {   
 free(M\_array[i]);   
 }   
 free(M\_array);   
}







5번 문항

#define \_CRT\_SECURE\_NO\_WARNINGS   
#include <stdio.h>   
#include <string.h>   
#include <ctype.h>   
   
   
int main(int argc, char\*\* argv) {   
   
 if (argc < 3) {   
 printf("인수를 제대로 전달해 주세요.\n"); //전달된 값이 3보다 작은 경우   
 return 0;   
 }   
   
 if (strcmp(argv[1], "-u") == 0) { //toUpper   
 printf("-u :");   
 for (int i = 0; i < strlen(argv[2]); i++) { //toUpper의 경우 '문자'를 변경하므로   
 printf("%c", toupper(\*(\*(argv + 2) + i)));//문자열의 길이만큼 반복문 실행   
 }   
 }   
   
 else if (strcmp(argv[1], "-l") == 0) {   
 printf("-l :");   
 for (int i = 0; i < strlen(argv[2]); i++) { //toLower의 경우 '문자'를 변경하므로   
 printf("%c", tolower(\*(\*(argv + 2) + i)));//문자열의 길이만큼 반복문 실행   
 }   
 }   
   
 else if (strcmp(argv[1], "-p") == 0) {//그대로 출력하는 옵션   
 printf("-p : %s",argv[2]);   
 }   
   
 else {   
 printf("지원되지 않는 옵션입니다\n"); //옵션 값이 오류가 났을 떄   
 }   
}

