기말고사

2016320256

김성준

1번)

#include <stdio.h>

#include <math.h>

#define N 3

char subject[N][17], grade[N];

int gpa[N];

void inputGrade();

void convertGrade();

int main()

{

int i;

double avg, sd, std=0, sum=0;

printf("%d개의 과목 성적을 입력합니다.\n\n", N);

inputGrade();

convertGrade();

printf("=============성적표===============\n");

printf("%20s %5s %5s\n", "과목", "성적", "평점");

printf("===================================\n");

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

{

printf("%20s %5c %5d\n", subject[i], grade[i], gpa[i]);

sum += gpa[i];

}

printf("===================================\n");

avg = sum / N;

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

{

std += ((gpa[i] - avg)\*(gpa[i] - avg))/N;

}

sd = sqrt(std);

printf("평균 평점 : %.2lf\n", avg);

printf("포준편차 : %.2lf\n", sd);

return 0;

}

void inputGrade()

{

int i;

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

{

printf("%d번째 과목의 이름은 무엇입니까? ", i+1);

scanf("%s", subject[i]);

printf("%s 과목의 성적은 무엇입니까? ", subject[i]);

scanf("%s", &grade[i]);

}

}

void convertGrade()

{

int i;

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

{

switch (grade[i])

{

case 'A' : gpa[i] = 4; break;

case 'B' : gpa[i] = 3; break;

case 'C' : gpa[i] = 2; break;

case 'D' : gpa[i] = 1; break;

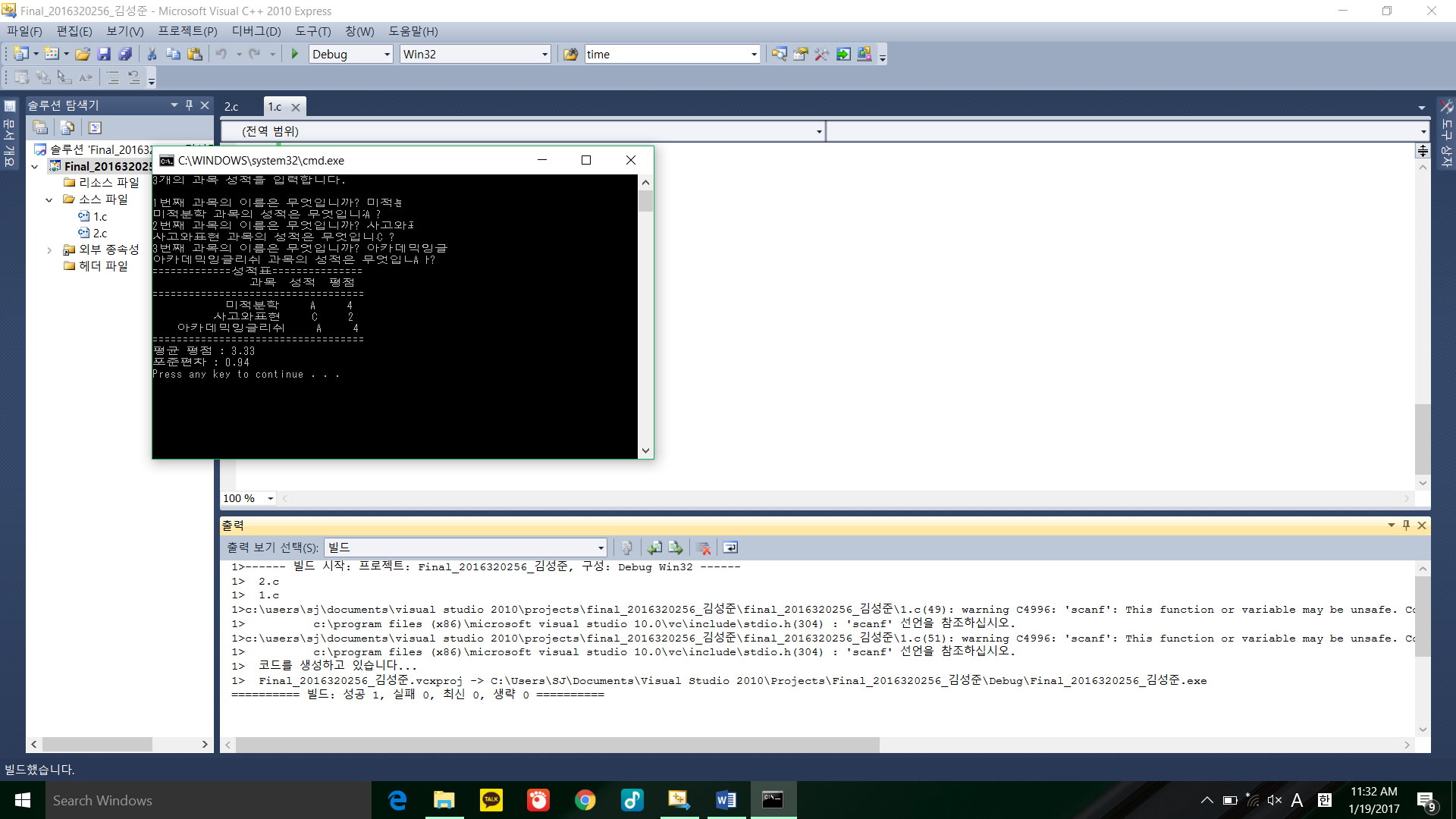
case 'F' : gpa[i] = 0; break;

default : printf("잘못된 입력입니다.\n");

}

}

}



2번)

#include <stdio.h>

#include <conio.h>

#include <string.h>

#include <stdlib.h>

#define MAX 10

struct person\_info

{

char name[9], year[3], month[3], day[3];

char \*address;

};

typedef struct person\_info PERSON;

void display();

void regist(PERSON person[]);

void print(PERSON person[]);

void sort(PERSON \*arr);

void swap(PERSON \*x, PERSON \*y);

void find(PERSON person[]);

void save(PERSON person[]);

int static current=0;

int main()

{

PERSON person[MAX];

int answer;

do

{

display();

scanf("%d", &answer);

switch (answer)

{

case 1: regist(person); break;

case 2: print(person); break;

case 3: sort(person); break;

case 4: find(person); break;

case 5: save(person); break;

case 6: exit(0);

}

} while (answer !=6);

}

void display()

{

printf("<회원 관리>\n");

printf("1. 등록\n");

printf("2. 보기\n");

printf("3. 정렬\n");

printf("4. 탐색\n");

printf("5. 결과저장\n");

printf("6. 종료\n");

printf("원하는 서비스를 선택하세요: ");

}

void regist(PERSON person[])

{

int total;

int static i;

char temp[12];

printf("\n등록할 회원의 수를 입력하세요: ");

scanf("%d", &total);

for (i; i<total; i++)

{

printf("\n<%d번째 회원 등록>\n", i+1);

printf("이름: "); scanf("%s", person[i].name);

printf("생년: "); scanf("%s", person[i].year);

printf("월: "); scanf("%s", person[i].month);

printf("일: "); scanf("%s", person[i].day);

printf("주소: ");

fflush(stdin);

gets(temp);

person[i].address = (char\*)malloc((strlen(temp)+1)\*sizeof(char));

if (person[i].address == NULL)

{

puts("동적 기억장소 할당에 실패하였습니다. \n");

exit(1);

}

strcpy(person[i].address, temp);

printf("%s 등록 완료\n", person[i].name);

}

current = i;

printf("\n계속 서비스를 원하시면 아무키나 누르세요.");

getch();

system("cls");

}

void print(PERSON person[])

{

int i;

printf("\n<생년월일 목록>\n");

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

{

printf("%s: %s-%s-%s, 주소: %s\n", person[i].name, person[i].year, person[i].month, person[i].day, person[i].address);

}

printf("\n계속 서비스를 원하시면 아무키나 누르세요.");

getch();

system("cls");

}

void sort(PERSON \*arr)

{

int i, j;

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

{

for(j=0; j<=current; j++)

{

if(strcmp((arr+j)->name,(arr+j+1)->name))

swap(&arr[j], &arr[j+1]);

}

}

printf("정렬 완료!\n");

printf("\n계속 서비스를 원하시면 아무키나 누르세요.");

getch();

system("cls");

}

void swap(PERSON \*x, PERSON \*y)

{

PERSON temp;

temp=\*x;

\*x = \*y;

\*y = temp;

}

void find(PERSON person[])

{

char findn[9];

int i, findi;

printf("\n찾고자 하는 회원의 이름을 입력하세요: ");

scanf("%s", findn);

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

{

if (strcmp(findn, person[i].name))

{

findi = i;

}

}

printf("%s: %s-%s-%s, 주소: %s\n", person[i].name, person[i].year, person[i].month, person[i].day, person[i].address);

}

void save(PERSON person[])

{

int i;

FILE \*fp = fopen("result.txt", "w");

if (fp == NULL)

{

printf("파일 열기 에러! "); exit(1);

}

for (i=1; i<=current; i++)

fprintf(fp, "%s: %s-%s-%s, 주소: %s\n", person[i].name, person[i].year, person[i].month, person[i].day, person[i].address);

fclose(fp);

}

