문제해결기법(13967005)

202135592 한웅재

소프트웨어

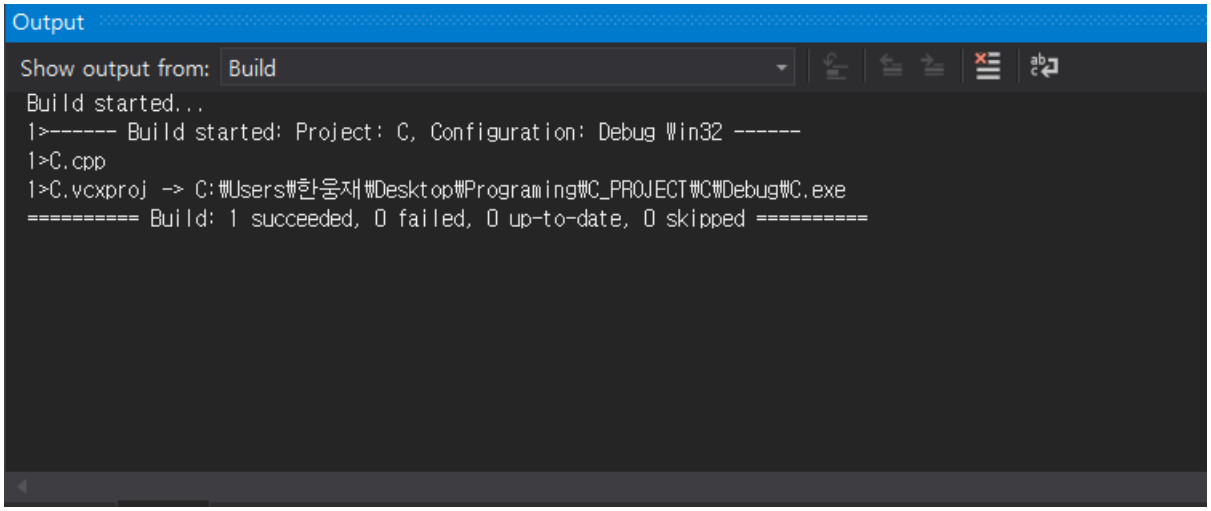
제출일: 2021. 11. 7

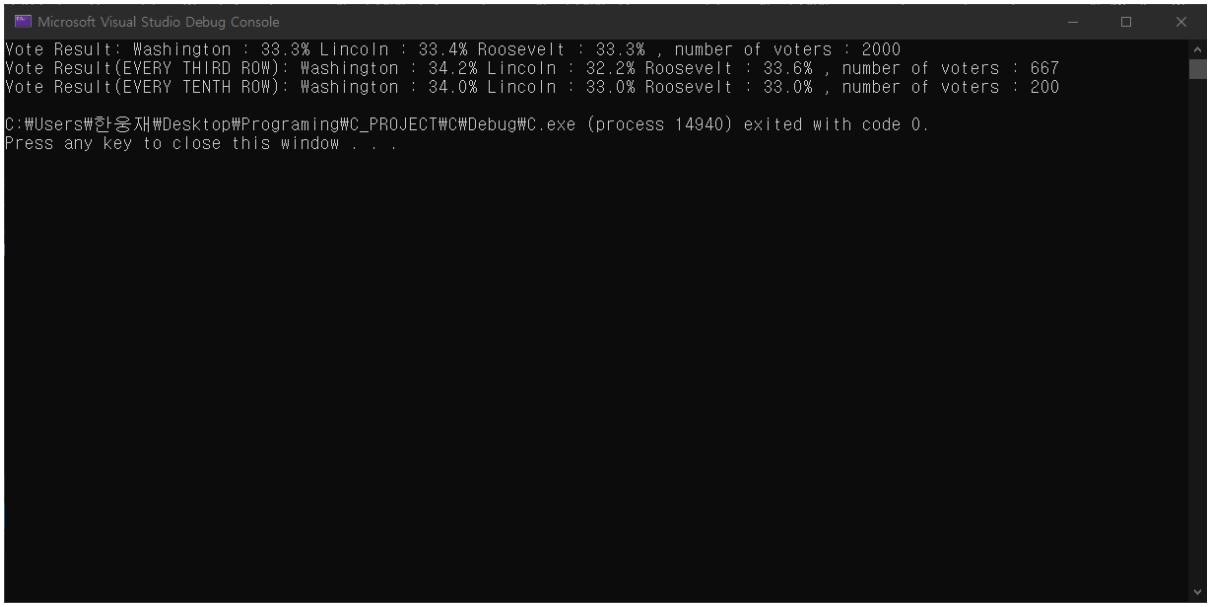
Q1. Lab-S (p.10) : terminal screenshot

```
#define _CRT_SECURE_NO_WARNINGS// or scanf_s
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <ctype.h>
#include <stdbool.h>
struct VOTE {
        char region[20];
        int age;
        char candidate_voted[20];
};

struct VOTE vote[2000];

void copy_element(struct VOTE src[], struct VOTE dest[]) {
        strcpy(dest->region, src->region);
        dest->age = src->age;
        strcpy(dest->candidate_voted, src->candidate_voted);
}

bool read_file(const char* fname) {
        FILE* pFile;
        pFile = fopen(fname, "r");
        if (pFile == NULL) {
                printf("cannot open the file!\n");
                return false;
        }
        struct VOTE person;
        int i = 0;
        while (fscanf(pFile, "%s %d %s", person.region, &person.age,
person.candidate_voted) == 3) {
                copy_element(&person, &vote[i]);
                i++;
        }



        fclose(pFile);
        return true;
}
void Compute_vote(struct VOTE* vote,
        float* Washington,
        float* Lincoln,
        float* Roosevelt,
        int* num_of_voter)
{
        float Washington_voted = 0;
        float Lincoln_voted = 0;
        float Roosevelt_voted = 0;
        for (int i = 0; i < 2000; i++) {
                if (strcmp(vote[i].candidate_voted, "Washington") == 0) {
                        Washington_voted++;
                        (*num_of_voter)++;
                }
                else if (strcmp(vote[i].candidate_voted, "Lincoln") == 0) {
                        Lincoln_voted++;
                        (*num_of_voter)++;
```

```
                }
                else if (strcmp(vote[i].candidate_voted, "Roosevelt") == 0) {
                        Roosevelt_voted++;
                        (*num_of_voter)++;
                }
        }

        *Washington = (Washington_voted / *num_of_voter) * (float)100.0;
        *Lincoln = (Lincoln_voted / *num_of_voter) * (float)100.0;
        *Roosevelt = (Roosevelt_voted / *num_of_voter) * (float)100.0;
}
void Compute_third_row_vote(struct VOTE* vote,
        float* Washington,
        float* Lincoln,
        float* Roosevelt,
        int* num_of_voter) {
        *num_of_voter = 0;
        float Washington_voted = 0;
        float Lincoln_voted = 0;
        float Roosevelt_voted = 0;
        for (int i = 0; i < 2000; i++) {
                if (i % 3 == 0) {
                        if (strcmp(vote[i].candidate_voted, "Washington") == 0) {
                                Washington_voted++;
                                (*num_of_voter)++;
                        }
                        else if (strcmp(vote[i].candidate_voted, "Lincoln") == 0) {
                                Lincoln_voted++;
                                (*num_of_voter)++;
                        }
                        else if (strcmp(vote[i].candidate_voted, "Roosevelt") == 0) {
                                Roosevelt_voted++;
                                (*num_of_voter)++;
                        }
                }
        }

        *Washington = (Washington_voted / *num_of_voter) * (float)100.0;
        *Lincoln = (Lincoln_voted / *num_of_voter) * (float)100.0;
        *Roosevelt = (Roosevelt_voted / *num_of_voter) * (float)100.0;

}
void Compute_tenth_row_vote(struct VOTE* vote,
        float* Washington,
        float* Lincoln,
        float* Roosevelt,
        int* num_of_voter) {
        *num_of_voter = 0;
        float Washington_voted = 0;
        float Lincoln_voted = 0;
        float Roosevelt_voted = 0;
        for (int i = 0; i < 2000; i++) {
                if (i % 10 == 0) {
                        if (strcmp(vote[i].candidate_voted, "Washington") == 0) {
                                Washington_voted++;
                                (*num_of_voter)++;
                        }
                        else if (strcmp(vote[i].candidate_voted, "Lincoln") == 0) {
                                Lincoln_voted++;
                                (*num_of_voter)++;
                        }
```

```
                        else if (strcmp(vote[i].candidate_voted, "Roosevelt") == 0) {
                                Roosevelt_voted++;
                                (*num_of_voter)++;
                        }
                }
        }

        *Washington = (Washington_voted / *num_of_voter) * (float)100.0;
        *Lincoln = (Lincoln_voted / *num_of_voter) * (float)100.0;
        *Roosevelt = (Roosevelt_voted / *num_of_voter) * (float)100.0;

}
int main() {
        char fname[50] = "vote.txt";
        int num_of_voters = 0;
        float Washington;
        float Lincoln;
        float Roosevelt;
        read_file(fname);

        Compute_vote(vote, &Washington, &Lincoln, &Roosevelt, &num_of_voters);
        printf("Vote Result: Washington : %.1f%% Lincoln : %.1f%% Roosevelt : %.1f%% ,
number of voters : %d\n", Washington, Lincoln, Roosevelt, num_of_voters);

        Compute_third_row_vote(vote, &Washington, &Lincoln, &Roosevelt,
&num_of_voters);
        printf("Vote Result(EVERY THIRD ROW): Washington : %.1f%% Lincoln : %.1f%%
Roosevelt : %.1f%% , number of voters : %d\n", Washington, Lincoln, Roosevelt,
num_of_voters);

        Compute_tenth_row_vote(vote, &Washington, &Lincoln, &Roosevelt,
&num_of_voters);
        printf("Vote Result(EVERY TENTH ROW): Washington : %.1f%% Lincoln : %.1f%%
Roosevelt : %.1f%% , number of voters : %d\n", Washington, Lincoln, Roosevelt,
num_of_voters);

        return 0;
}
```

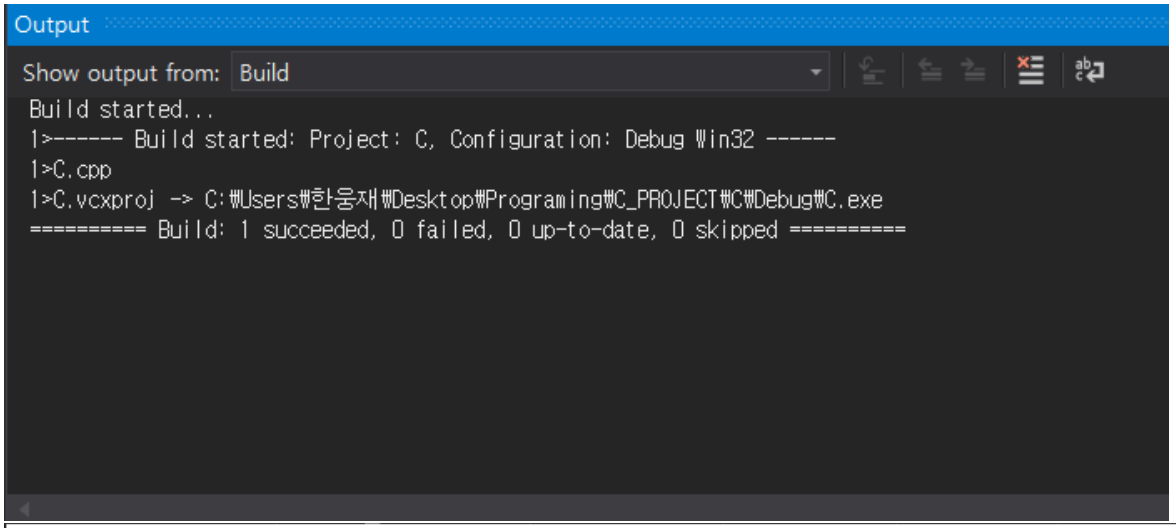




Q2. Lab-H (p.13) : output.txt screenshot

```c
#define _CRT_SECURE_NO_WARNINGS// or scanf_s
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <ctype.h>
#include <stdbool.h>
struct PERSONAL {
	char name[20];
	int age;
	char hobby[20];
};

struct PERSONAL personal[20];
struct PERSONAL decomposed_data[20];

void copy_element(struct PERSONAL src[], struct PERSONAL dest[]) {
	strcpy(dest->name, src->name);
	dest->age = src->age;
	strcpy(dest->hobby, src->hobby);
}
void group_by_age(struct PERSONAL* original, struct PERSONAL* age) {
	int index = 0;
	for (int k = 1; k < 5; k++) {
		for (int i = 0; i < 20; i++) {
			int p_age = (original[i].age / 10) * 10;
			if (p_age == k * 10) {
				copy_element(&original[i], &age[index]);
				index++;
			}

		}
	}
}
bool write_file(const char* fname, struct PERSONAL edit[]) {
	FILE* pFile;
	pFile = fopen(fname, "w");
```

```
	if (pFile == NULL) {
		printf("cannot open the file!\n");
		return false;
	}
	for (int i = 0; i < 20; i++) {
		if (i == 0) {
			fprintf(pFile, "Age from %d to %d\n--------------------------
\n", (edit[i].age / 10) * 10, (edit[i].age / 10) * 10 + 9);
		}
		if (i >= 1 && i <= 19) {
			if (edit[i].age / 10 - edit[i - 1].age / 10) {
				fprintf(pFile, "\nAge from %d to %d\n------------------
--------\n", (edit[i].age / 10) * 10, (edit[i].age / 10) * 10 + 9);
			}
		}
		fprintf(pFile, "%8s %d %8s\n",
			edit[i].name,
			edit[i].age,
			edit[i].hobby);

	}


	fclose(pFile);
	return true;
}

bool read_file(const char* fname) {
	FILE* pFile;
	pFile = fopen(fname, "r");
	if (pFile == NULL) {
		printf("cannot open the file!\n");
		return false;
	}
	struct PERSONAL person;
	int i = 0;
	while (fscanf(pFile, "%s %d %s", person.name, &person.age, person.hobby) == 3)
{
		copy_element(&person, &personal[i]);
		i++;
	}
	fclose(pFile);
	return true;
}
int main() {
	char fname[50] = "personal.txt";
	char output_name[30] = "output.txt";
	read_file(fname);
	group_by_age(personal, decomposed_data);
	write_file(output_name, decomposed_data);

	return 0;
}
```



output.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
Age from 10 to 19
--------------------------
     Ko 15   Soccer
    Lee 17   Soccer
   Park 10   Tennis
     Ra 15    Piano
   Shin 16   Tennis
   Jang 15   Soccer
   Moon 18    Tennis

Age from 20 to 29
--------------------------
   Choi 21   Tennis
   Kang 21   Guitar
    Cho 25    Piano
     Mo 24   Guitar
   Jeon 21   Soccer
    Lim 21   Tennis
     Oh 29   Guitar

Age from 30 to 39
--------------------------
    Kim 39   Tennis
    Heo 39   Tennis
  Jeong 39   Soccer
   Seol 36    Piano

Age from 40 to 49
--------------------------
    Cha 41    Piano
    Yoo 49   Tennis
```

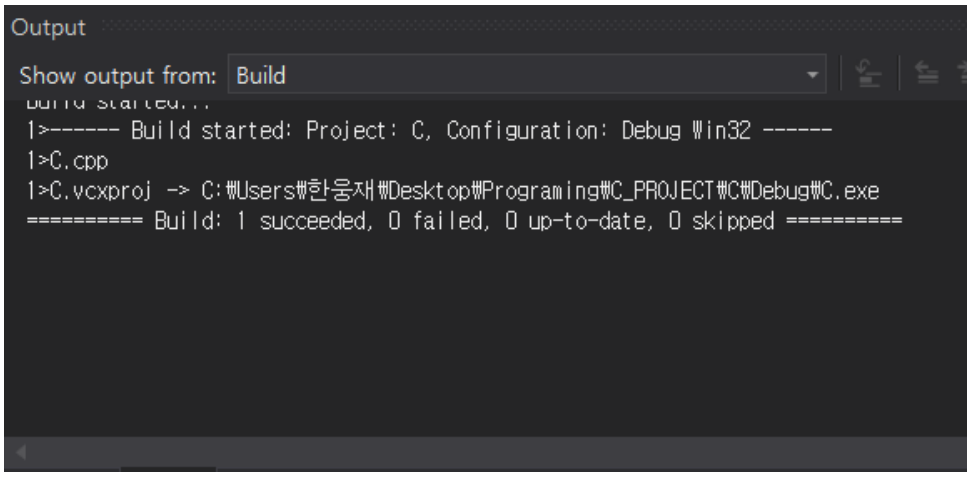
Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

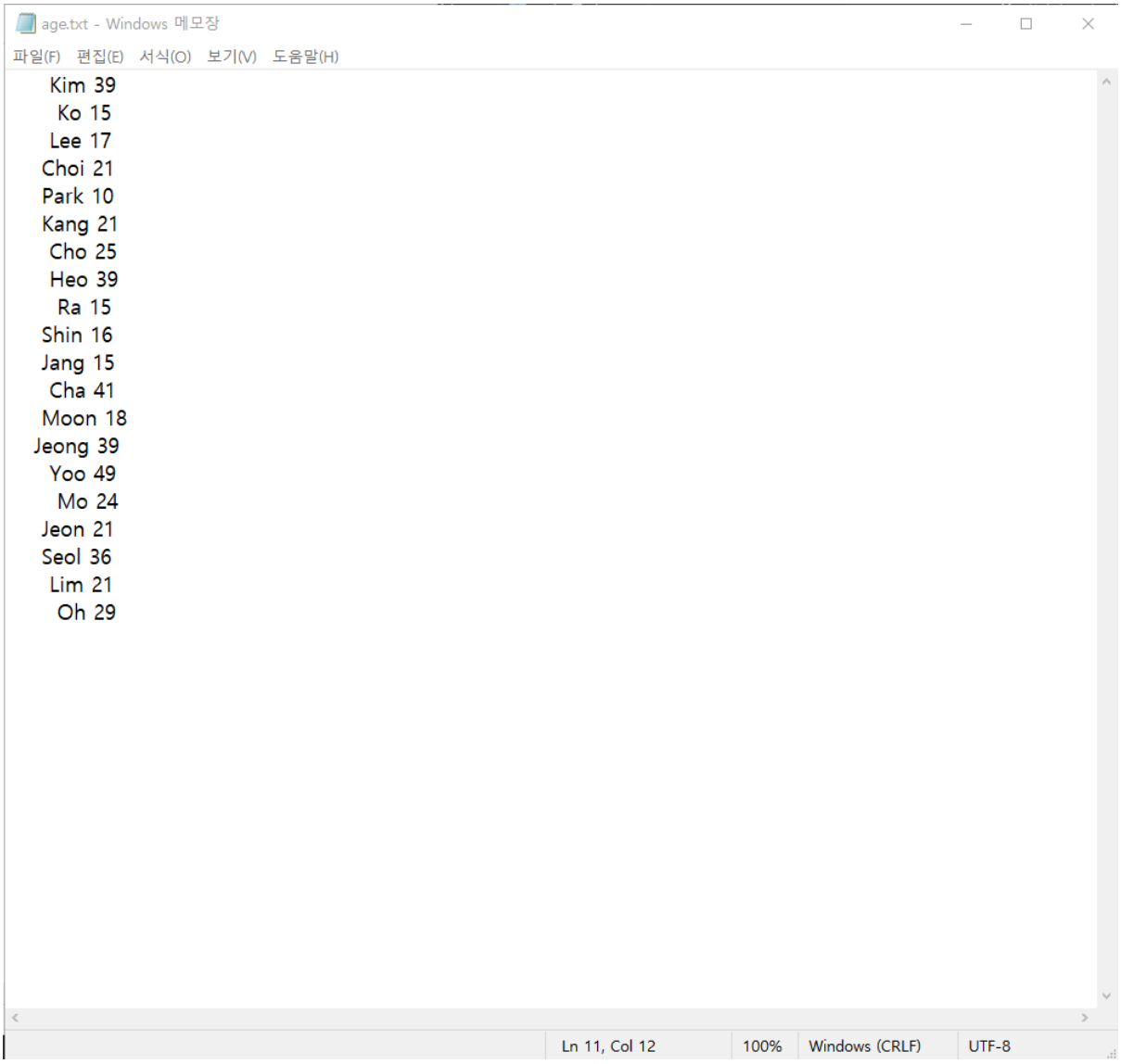
Q3. Lab-V (p.15) : age.txt & hobby.txt screenshots

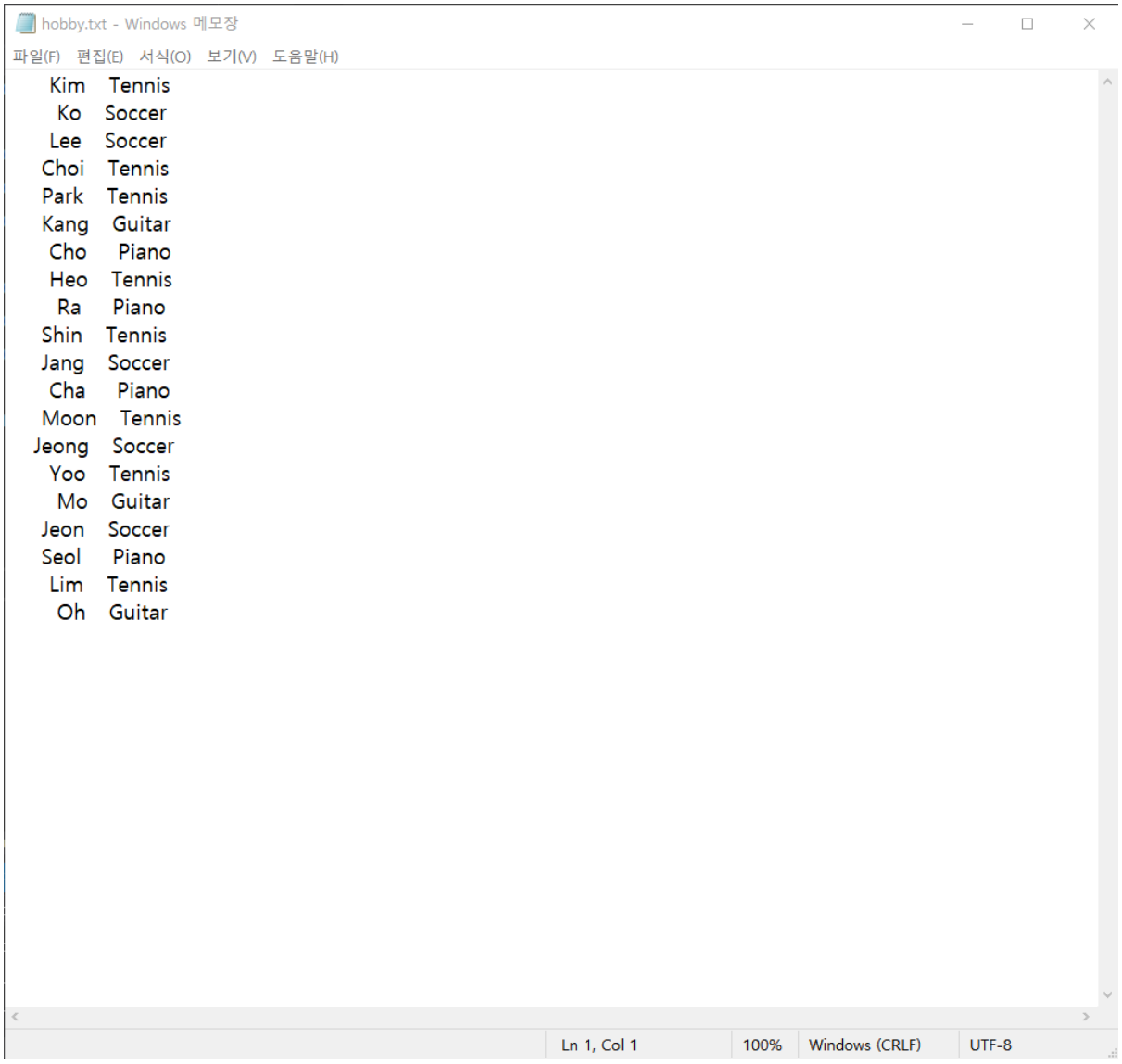
```c
#define _CRT_SECURE_NO_WARNINGS// or scanf_s
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <ctype.h>
#include <stdbool.h>
struct PERSONAL {
	char name[20];
	int age;
	char hobby[20];
};
struct NAME_AGE {
	char name[20];
	int age;
};
struct NAME_HOBBY {
	char name[20];
	char hobby[20];
};

struct PERSONAL personal[20];
struct NAME_AGE Age[20];
struct NAME_HOBBY Hobby[20];

void copy_element(struct PERSONAL src[], struct PERSONAL dest[]) {
	strcpy(dest->name, src->name);
	dest->age = src->age;
	strcpy(dest->hobby, src->hobby);
}
void copy_age_element(struct PERSONAL src[], struct NAME_AGE dest[]) {
	strcpy(dest->name, src->name);
	dest->age = src->age;
}
void copy_hobby_element(struct PERSONAL src[], struct NAME_HOBBY dest[]) {
	strcpy(dest->name, src->name);
	strcpy(dest->hobby, src->hobby);
}

bool write_file(const char* fname, struct PERSONAL edit[]) {
	FILE* pFile;
	pFile = fopen(fname, "w");
	if (pFile == NULL) {
		printf("cannot open the file!\n");
		return false;
	}
	for (int i = 0; i < 20; i++) {
		fprintf(pFile, "%8s %d %8s\n",
			edit[i].name,
			edit[i].age,
			edit[i].hobby);
	}


	fclose(pFile);
	return true;
}
bool write_file_for_age(const char* fname, struct NAME_AGE edit[]) {
```

```
	FILE* pFile;
	pFile = fopen(fname, "w");
	if (pFile == NULL) {
		printf("cannot open the file!\n");
		return false;
	}
	for (int i = 0; i < 20; i++) {
		fprintf(pFile, "%8s %d\n",
			edit[i].name,
			edit[i].age);
	}


	fclose(pFile);
	return true;
}
bool write_file_for_hobby(const char* fname, struct NAME_HOBBY edit[]) {
	FILE* pFile;
	pFile = fopen(fname, "w");
	if (pFile == NULL) {
		printf("cannot open the file!\n");
		return false;
	}
	for (int i = 0; i < 20; i++) {
		fprintf(pFile, "%8s %8s\n",
			edit[i].name,
			edit[i].hobby);
	}


	fclose(pFile);
	return true;
}

bool read_file(const char* fname) {
	FILE* pFile;
	pFile = fopen(fname, "r");
	if (pFile == NULL) {
		printf("cannot open the file!\n");
		return false;
	}
	struct PERSONAL person;
	int i = 0;
	while (fscanf(pFile, "%s %d %s", person.name, &person.age, person.hobby) == 3)
{
		copy_element(&person, &personal[i]);
		i++;
	}
	fclose(pFile);
	return true;
}
bool read_file_for_age(const char* fname, struct NAME_AGE* Age) {
	FILE* pFile;
	pFile = fopen(fname, "r");
	if (pFile == NULL) {
		printf("cannot open the file!\n");
		return false;
	}
	struct PERSONAL person;
	int i = 0;
	while (fscanf(pFile, "%s %d %s", person.name, &person.age, person.hobby) == 3)
```

```
{
                copy_age_element(&person, &Age[i]);
                i++;
        }
        fclose(pFile);
        return true;


}
bool read_file_for_hobby(const char* fname, struct NAME_HOBBY* Hobby) {
        FILE* pFile;
        pFile = fopen(fname, "r");
        if (pFile == NULL) {
                printf("cannot open the file!\n");
                return false;
        }
        struct PERSONAL person;
        int i = 0;
        while (fscanf(pFile, "%s %d %s", person.name, &person.age, person.hobby) == 3)
{
                copy_hobby_element(&person, &Hobby[i]);
                i++;
        }
        fclose(pFile);
        return true;


}

int main() {
        char fname[50] = "personal.txt";
        char fage[30] = "age.txt";
        char fhobby[30] = "hobby.txt";

        read_file(fname);
        read_file_for_age(fname,Age);
        read_file_for_hobby(fname, Hobby);

        write_file_for_age(fage,Age);
        write_file_for_hobby(fhobby,Hobby);
        return 0;
}
```



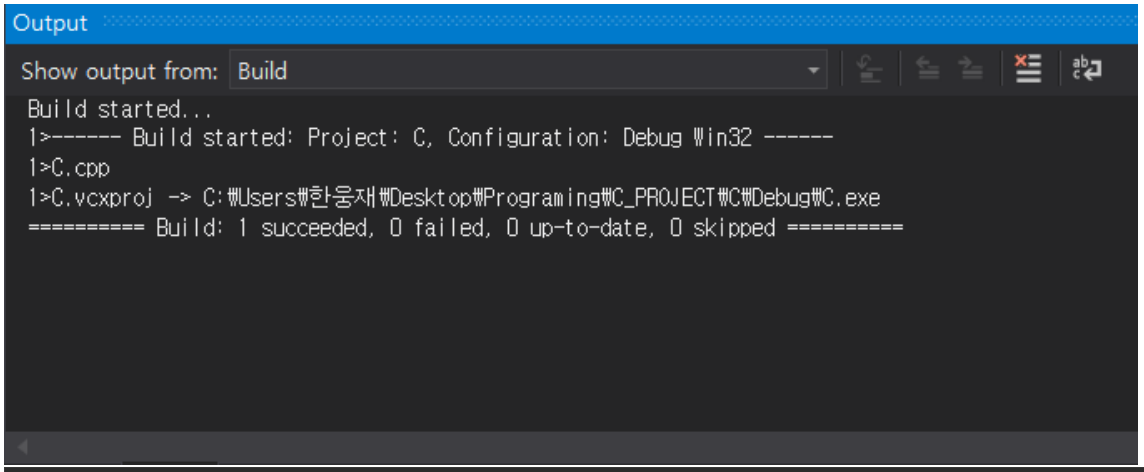

age.txt - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
Kim 39
Ko 15
Lee 17
Choi 21
Park 10
Kang 21
Cho 25
Heo 39
Ra 15
Shin 16
Jang 15
Cha 41
Moon 18
Jeong 39
Yoo 49
Mo 24
Jeon 21
Seol 36
Lim 21
Oh 29
Ln 11, Col 12
100%
Windows (CRLF)
UTF-8

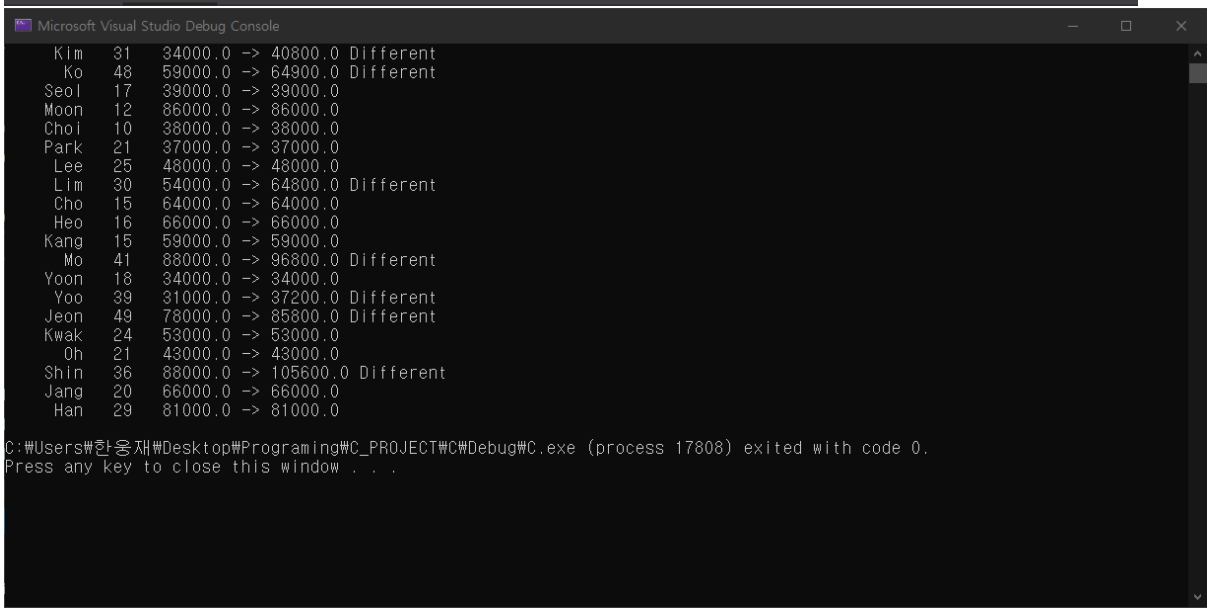

hobby.txt - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
Kim Tennis
Ko Soccer
Lee Soccer
Choi Tennis
Park Tennis
Kang Guitar
Cho Piano
Heo Tennis
Ra Piano
Shin Tennis
Jang Soccer
Cha Piano
Moon Tennis
Jeong Soccer
Yoo Tennis
Mo Guitar
Jeon Soccer
Seol Piano
Lim Tennis
Oh Guitar
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8

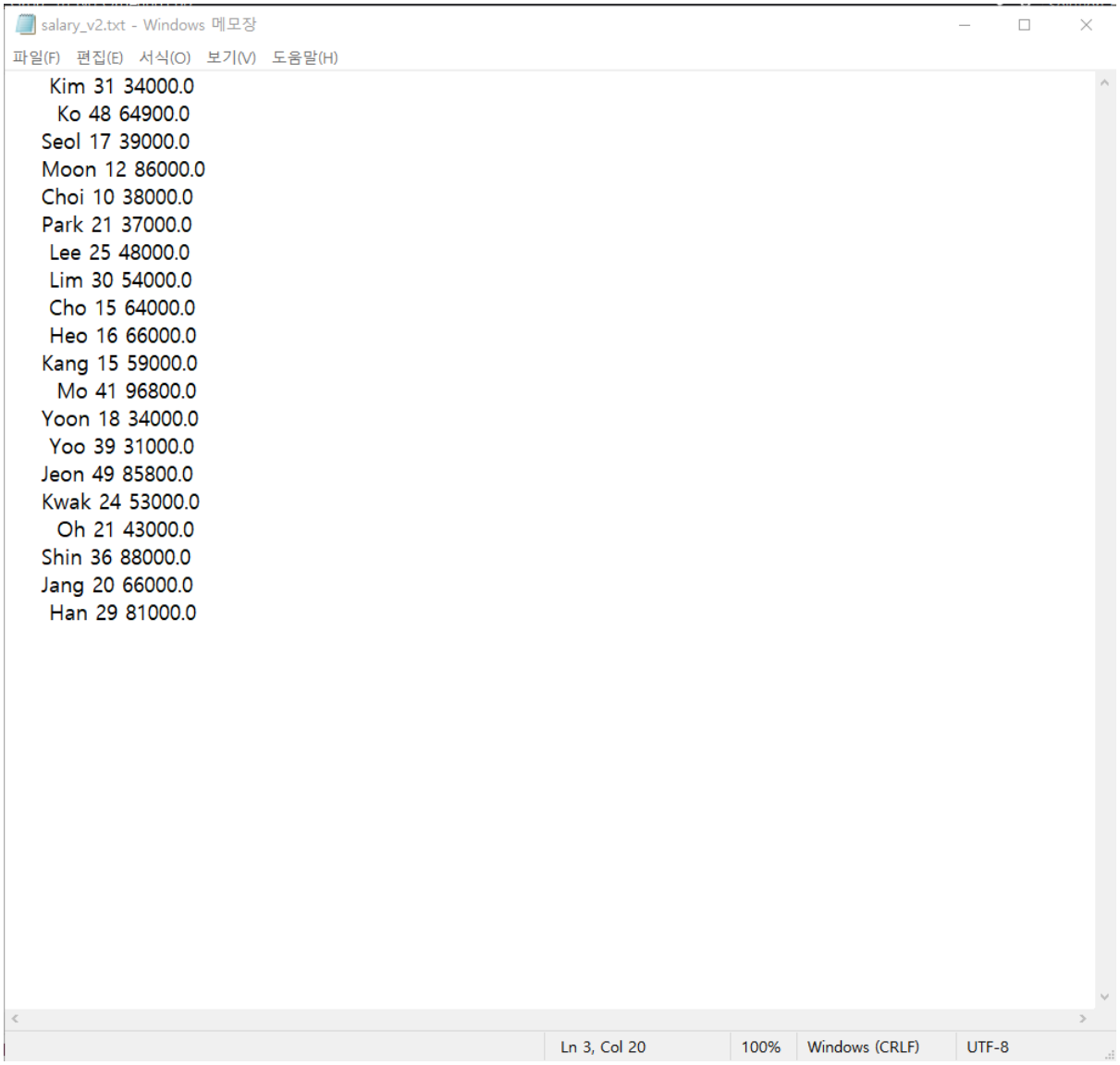
Q4. Lab-VS (p.19,20) : salary_v2.txt, salary_v3.txt screenshots + terminal screenshot

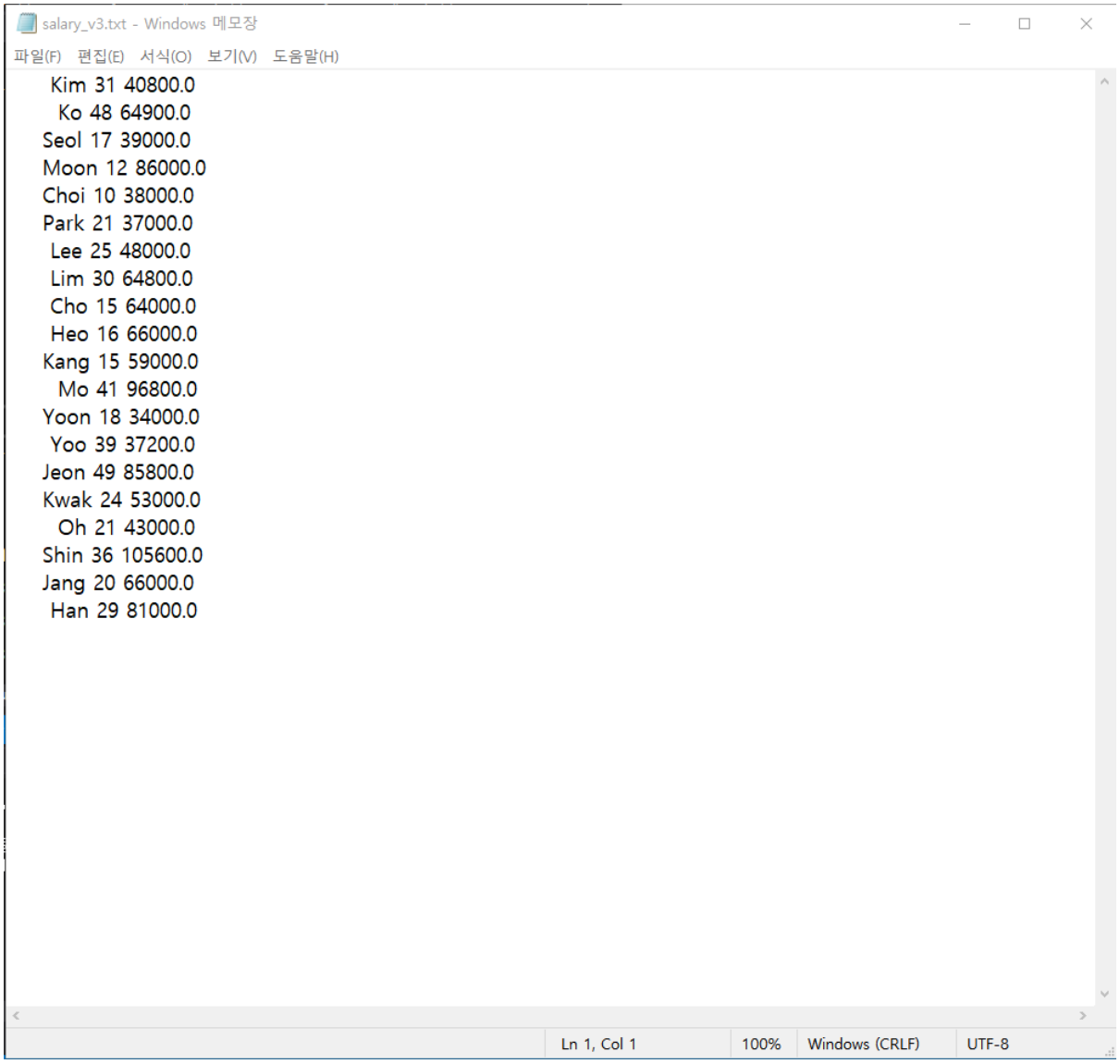
```
#define _CRT_SECURE_NO_WARNINGS// or scanf_s
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <ctype.h>
#include <stdbool.h>
struct PERSONAL {
        char name[20];
        int age;
        double salary;
};

struct PERSONAL personal_1[20];
struct PERSONAL personal_2[20];
struct PERSONAL personal_3[20];
struct PERSONAL temp[20];

void copy_element(struct PERSONAL src[], struct PERSONAL dest[]) {
        strcpy(dest->name, src->name);
        dest->age = src->age;
        dest->salary = src->salary;
}

bool write_file(const char* fname, struct PERSONAL edit[]) {
        FILE* pFile;
        pFile = fopen(fname, "w");
        if (pFile == NULL) {
                printf("cannot open the file!\n");
                return false;
        }
        for (int i = 0; i < 20; i++) {
                fprintf(pFile, "%8s %d %.1lf\n",
                        edit[i].name,
                        edit[i].age,
                        edit[i].salary);
        }
        fclose(pFile);
        return true;
}

bool read_file(const char* fname,struct PERSONAL* data) {
        FILE* pFile;
        pFile = fopen(fname, "r");
        if (pFile == NULL) {
                printf("cannot open the file!\n");
                return false;
        }
        struct PERSONAL person;
        int i = 0;
        while (fscanf(pFile, "%s %d %lf", person.name, &person.age, &person.salary) ==
3) {
                copy_element(&person, &data[i]);
                i++;
        }
        fclose(pFile);
        return true;
}
```

```c
void Update_1(struct PERSONAL *personal1, struct PERSONAL* personal2) {
	for (int i = 0; i < 20; i++) {
		copy_element(&personal1[i], &personal2[i]);
		if ((personal1[i].age / 10) * 10 == 40) {
			personal2[i].salary = personal1[i].salary * 1.1;
		}
	}

}
void Update_2(struct PERSONAL* personal2, struct PERSONAL* personal3) {
	for (int i = 0; i < 20; i++) {
		copy_element(&personal2[i], &personal3[i]);
		if ((personal2[i].age / 10) * 10 == 30) {
			personal3[i].salary = personal2[i].salary * 1.2;
		}
	}

}
void Compare_element(struct PERSONAL* personal1, struct PERSONAL* personal3) {
	for (int i = 0; i < 20; i++) {
		printf("%8s   %d   %.1lf -> %.1lf", personal1[i].name,
personal1[i].age, personal1[i].salary, personal3[i].salary);
		if (personal1[i].salary != personal3[i].salary)
			printf(" Different\n");
		else
			printf("\n");

	}

}
int main() {
	char fname[50] = "salary_v1.txt";
	char fname_2[30] = "salary_v2.txt";
	char fname_3[30] = "salary_v3.txt";

	read_file(fname,personal_1);
	Update_1(personal_1, temp);
	write_file(fname_2, temp);



	read_file(fname_2,personal_2);
	Update_2(personal_2, personal_3);
	write_file(fname_3, personal_3);


	Compare_element(personal_1,personal_3);

	return 0;
}
```


Output
Show output from: Build
Build started...
1>------ Build started: Project: C, Configuration: Debug Win32 ------
1>C.cpp
1>C.vcxproj -> C:₩Users₩한웅재₩Desktop₩Programing₩C_PROJECT₩C₩Debug₩C.exe
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========


Microsoft Visual Studio Debug Console
Kim 31 34000.0 -> 40800.0 Different
Ko 48 59000.0 -> 64900.0 Different
Seol 17 39000.0 -> 39000.0
Moon 12 86000.0 -> 86000.0
Choi 10 38000.0 -> 38000.0
Park 21 37000.0 -> 37000.0
Lee 25 48000.0 -> 48000.0
Lim 30 54000.0 -> 64800.0 Different
Cho 15 64000.0 -> 64000.0
Heo 16 66000.0 -> 66000.0
Kang 15 59000.0 -> 59000.0
Mo 41 88000.0 -> 96800.0 Different
Yoon 18 34000.0 -> 34000.0
Yoo 39 31000.0 -> 37200.0 Different
Jeon 49 78000.0 -> 85800.0 Different
Kwak 24 53000.0 -> 53000.0
Oh 21 43000.0 -> 43000.0
Shin 36 88000.0 -> 105600.0 Different
Jang 20 66000.0 -> 66000.0
Han 29 81000.0 -> 81000.0
C:₩Users₩한웅재₩Desktop₩Programing₩C_PROJECT₩C₩Debug₩C.exe (process 17808) exited with code 0.
Press any key to close this window . . .


salary_v2.txt - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
Kim 31 34000.0
Ko 48 64900.0
Seol 17 39000.0
Moon 12 86000.0
Choi 10 38000.0
Park 21 37000.0
Lee 25 48000.0
Lim 30 54000.0
Cho 15 64000.0
Heo 16 66000.0
Kang 15 59000.0
Mo 41 96800.0
Yoon 18 34000.0
Yoo 39 31000.0
Jeon 49 85800.0
Kwak 24 53000.0
Oh 21 43000.0
Shin 36 88000.0
Jang 20 66000.0
Han 29 81000.0
Ln 3, Col 20
100%
Windows (CRLF)
UTF-8


salary_v3.txt - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
Kim 31 40800.0
Ko 48 64900.0
Seol 17 39000.0
Moon 12 86000.0
Choi 10 38000.0
Park 21 37000.0
Lee 25 48000.0
Lim 30 64800.0
Cho 15 64000.0
Heo 16 66000.0
Kang 15 59000.0
Mo 41 96800.0
Yoon 18 34000.0
Yoo 39 37200.0
Jeon 49 85800.0
Kwak 24 53000.0
Oh 21 43000.0
Shin 36 105600.0
Jang 20 66000.0
Han 29 81000.0
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8