<lab1.c code>

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct personal
{
    char name[10];
    int age;
    char hobby[10];
} personal[20];

void printData(FILE *file, int count, struct personal personal[]);
void groupByAge(int count, struct personal personal[]);
void groupByHobby(int count, struct personal personal[]);

int main()
{
    FILE *file;
    int dataCount = 0;

    file = fopen("personal.txt", "r");
    if (file == NULL)
    {
        printf("Input File Could Not Be Opened.");
        exit(1);
    }
    else
    {
        int i = 0;
        while (fscanf(file, "%s %d %s", personal[i].name, &personal[i].age, personal[i].hobby) != EOF)
        {
            i++;
            dataCount++;
        }
        fclose(file);
    }

    file = fopen("hobby.txt", "w");
    if (file == NULL)
    {
        printf("Output File Could Not Be Opened.");
        exit(1);
    }
    else
    {
        groupByHobby(dataCount, personal);
        printData(file, dataCount, personal);
        fclose(file);
    }

    file = fopen("age.txt", "w");
    if (file == NULL)
    {
        printf("Output File Could Not Be Opened.");
        exit(1);
    }
    else
    {
        groupByAge(dataCount, personal);
        printData(file, dataCount, personal);
        fclose(file);
    }
    return 0;
}

void printData(FILE *file, int count, struct personal personal[])
{
    fprintf(file, "Name      Age     Hobby");
    for (int j = 0; j < count && personal[j].age != 0; j++)
    {
        fprintf(file, "\n%-10s%-8d%-10s", personal[j].name, personal[j].age, personal[j].hobby);
    }
}

void groupByHobby(int count, struct personal personal[])
{
    for (int i = 0; i < count - 1; i++)
    {
        for (int j = 0; j < count - 1 - i; j++)
        {
            if (strcmp(personal[j].hobby, personal[j + 1].hobby) > 0)
            {
                struct personal temp = personal[j];
                personal[j] = personal[j + 1];
                personal[j + 1] = temp;
            }
        }
    }
}

void groupByAge(int count, struct personal personal[])
{
    for (int i = 0; i < count - 1; i++)
    {
        for (int j = 0; j < count - 1 - i; j++)
        {
            if (personal[j].age > personal[j + 1].age)
            {
                struct personal temp = personal[j];
                personal[j] = personal[j + 1];
                personal[j + 1] = temp;
            }
        }
    }
}
```

lab1 : grouping

- read data from personal.txt

- group data by age and write to file age.txt

- group data by hobby and write to file hobby.txt

<결과화면>

age.txt, hobby.txt

Lab > lab1 > age.txt

	Name	Age	Hobby
1	Park	10	Tennis
2	Ra	15	Piano
3	Ko	15	Soccer
4	Jang	15	Soccer
5	Shin	16	Tennis
6	Lee	17	Soccer
7	Moon	18	Tennis
8	Kang	21	Guitar
9	Jeon	21	Soccer
10	Choi	21	Tennis
11	Lim	21	Tennis
12	Mo	24	Guitar
13	Cho	25	Piano
14	Oh	29	Guitar
15	Seol	36	Piano
16	Jeong	39	Soccer
17	Kim	39	Tennis
18	Heo	39	Tennis
19	Cha	41	Piano
20	Yoo	49	Tennis

Lab > lab1 > hobby.txt

	Name	Age	Hobby
1	Kang	21	Guitar
2	Mo	24	Guitar
3	Oh	29	Guitar
4	Cho	25	Piano
5	Ra	15	Piano
6	Cha	41	Piano
7	Seol	36	Piano
8	Ko	15	Soccer
9	Lee	17	Soccer
10	Jang	15	Soccer
11	Jeong	39	Soccer
12	Jeon	21	Soccer
13	Kim	39	Tennis
14	Choi	21	Tennis
15	Park	10	Tennis
16	Heo	39	Tennis
17	Shin	16	Tennis
18	Moon	18	Tennis
19	Yoo	49	Tennis
20	Lim	21	Tennis

<lab2.c code>

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main()
{
    FILE *file;
    char voteTo[20];

    int washington = 0, lincoln = 0, roosevelt = 0;
    int totalVote = 0;
    file = fopen("vote.txt", "r");
    while (fscanf(file, "%*s %*d %s", voteTo) != EOF)
    {
        if (strcmp(voteTo, "Washington") == 0)
        {
            washington++;
        }
        else if (strcmp(voteTo, "Lincoln") == 0)
        {
            lincoln++;
        }
        else if (strcmp(voteTo, "Roosevelt") == 0)
        {
            roosevelt++;
        }
        totalVote++;
    }
    fclose(file);

    printf("*All Data*\n");
    printf("Washington ( %d voted ) : %.2f%%\n", washington, (float)washington / totalVote *
100);printf("Lincoln ( %d voted ) : %.2f%%\n", lincoln, (float)lincoln / totalVote * 100);
    printf("Roosevelt ( %d voted ) : %.2f%%\n", roosevelt, (float)roosevelt / totalVote * 100);
    printf("-------------------------------------------------------------------------\n");
    // Sampling every third row
    washington = 0, lincoln = 0, roosevelt = 0;
    totalVote = 0;
    int row = 1;
    file = fopen("vote.txt", "r");
    while (fscanf(file, "%*s %*d %s", voteTo) != EOF)
    {
        if (row % 3 == 0)
        {
            if (strcmp(voteTo, "Washington") == 0)
            {
                washington++;
            }
            else if (strcmp(voteTo, "Lincoln") == 0)
            {
                lincoln++;
            }
            else if (strcmp(voteTo, "Roosevelt") == 0)
            {
                roosevelt++;
            }
            totalVote++;
        }
        row++;
    }
    fclose(file);
    printf("*Sampling Every 3th row*\n");
    printf("Washington ( %d voted ) : %.2f%%\n", washington, (float)washington / totalVote *
100);printf("Lincoln ( %d voted ) : %.2f%%\n", lincoln, (float)lincoln / totalVote * 100);
    printf("Roosevelt ( %d voted ) : %.2f%%\n", roosevelt, (float)roosevelt / totalVote * 100);
    printf("-------------------------------------------------------------------------\n");

    // Sampling every tenth row
    washington = 0, lincoln = 0, roosevelt = 0;
    totalVote = 0;
    row = 1;
    file = fopen("vote.txt", "r");
    while (fscanf(file, "%*s %*d %s", voteTo) != EOF)
    {
        if (row % 10 == 0)
        {
            if (strcmp(voteTo, "Washington") == 0)
            {
                washington++;
            }
            else if (strcmp(voteTo, "Lincoln") == 0)
            {
                lincoln++;
            }
            else if (strcmp(voteTo, "Roosevelt") == 0)
            {
                roosevelt++;
            }
            totalVote++;
        }
        row++;
    }
    fclose(file);
    printf("*Sampling Every 10th row*\n");
    printf("Washington ( %d voted ) : %.2f%%\n", washington, (float)washington / totalVote *
100);printf("Lincoln ( %d voted ) : %.2f%%\n", lincoln, (float)lincoln / totalVote * 100);
    printf("Roosevelt ( %d voted ) : %.2f%%\n", roosevelt, (float)roosevelt / totalVote * 100);
    return 0;
}
```

lab2

- read data from file vote.txt

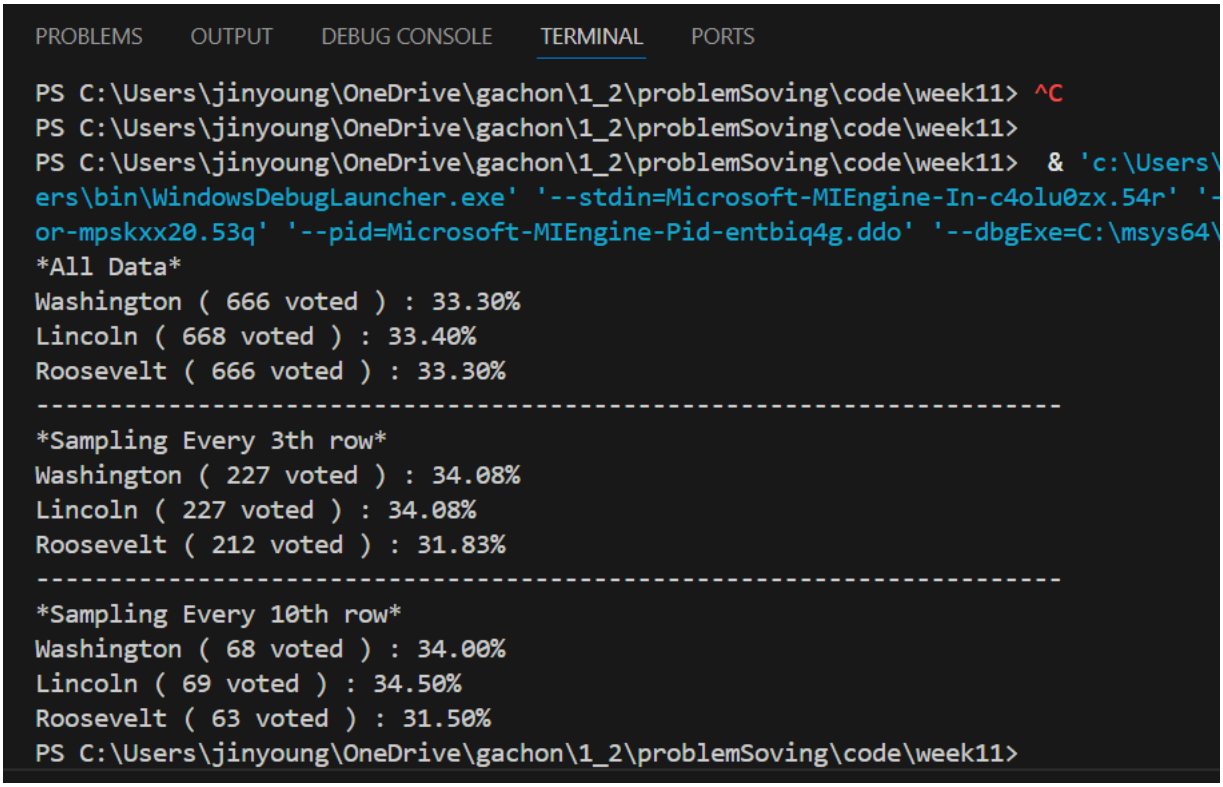
- compute percentage of votes each candidate received and print to terminal

- sample every third row

- sample every tenth row

<결과화면>

lab2 Terminal output



<lab3.c code>

```c
#include <stdio.h>
#include <stdlib.h>

struct personal
{
    char name[10];
    int age;
    char hobby[10];
} personal[20];

int main()
{
    FILE *file;
    int dataCount = 0;
    file = fopen("personal.txt", "r");
    if (file == NULL)
    {
        printf("Input File Could Not Be Opened.");
        exit(1);
    }
    else
    {
        int i = 0;
        while (fscanf(file, "%s %d %s", personal[i].name, &personal[i].age, personal[i].hobby) != EOF)
        {
            i++;
            dataCount++;
        }
        fclose(file);
    }

    // Decompose data by age.
    struct personal age1019[dataCount], age2029[dataCount], age3039[dataCount], age4049[dataCount];
    int cursor1019 = 0, cursor2029 = 0, cursor3039 = 0, cursor4049 = 0;

    for (int i = 0; i < dataCount; i++)
    {
        if (personal[i].age >= 10 && personal[i].age < 20)
        {
            age1019[cursor1019] = personal[i];
            cursor1019++;
        }
        else if (personal[i].age >= 20 && personal[i].age < 30)
        {
            age2029[cursor2029] = personal[i];
            cursor2029++;
        }
        else if (personal[i].age >= 30 && personal[i].age < 40)
        {
            age3039[cursor3039] = personal[i];
            cursor3039++;
        }
        else if (personal[i].age >= 40 && personal[i].age < 50)
        {
            age4049[cursor4049] = personal[i];
            cursor4049++;
        }
    }

    // Generate a struct arrary for each age range.
    file = fopen("output.txt", "w");
    if (file == NULL)
    {
        printf("Output File Could Not Be Opened.");
        exit(1);
    }
    else
    {
        fprintf(file, "Age from 10 to 19\n");
        fprintf(file, "--------------------\n");
        for (int i = 0; i < cursor1019; i++)
        {
            fprintf(file, "%-5s %-5d %s\n", age1019[i].name, age1019[i].age, age1019[i].hobby);
        }

        fprintf(file, "\nAge from 20 to 29\n");
        fprintf(file, "--------------------\n");
        for (int i = 0; i < cursor2029; i++)
        {
            fprintf(file, "%-5s %-5d %s\n", age2029[i].name, age2029[i].age, age2029[i].hobby);
        }

        fprintf(file, "\nAge from 30 to 39\n");
        fprintf(file, "--------------------\n");
        for (int i = 0; i < cursor3039; i++)
        {
            fprintf(file, "%-5s %-5d %s\n", age3039[i].name, age3039[i].age, age3039[i].hobby);
        }

        fprintf(file, "\nAge from 40 to 49\n");
        fprintf(file, "--------------------\n");
        for (int i = 0; i < cursor4049; i++)
        {
            fprintf(file, "%-5s %-5d %s\n", age4049[i].name, age4049[i].age, age4049[i].hobby);
        }

        fclose(file);
    }
    return 0;
}
```

lab3

- read data from file personal.txt

- decompose data by age as follows

- generate a struct array for each age range and write to file output.txt in the following format

<결과화면>

output.txt

Lab > lab3 > output.txt

```
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

```

<lab4.c code>

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct personal
{
    char name[10];
    int age;
    char hobby[10];
} personal[100];

struct nameAge
{
    char name[10];
    int age;
} nameAge[100];

struct nameHobby
{
    char name[10];
    char hobby[10];
} nameHobby[100];

int main()
{
    FILE *file;
    int dataCount = 0;
    file = fopen("personal.txt", "r");
    if (file == NULL)
    {
        printf("Input File Could Not Be Opened.");
        exit(1);
    }
    else
    {
        int i = 0;
        while (fscanf(file, "%s %d %s", personal[i].name, &personal[i].age, personal[i].hobby) != EOF)
        {
            i++;
            dataCount++;
        }
        fclose(file);
    }

    for (int i = 0; i < dataCount; i++)
    {
        strcpy(nameAge[i].name, personal[i].name);
        nameAge[i].age = personal[i].age;
        strcpy(nameHobby[i].name, personal[i].name);
        strcpy(nameHobby[i].hobby, personal[i].hobby);
    }

    FILE *ageFile = fopen("age.txt", "w");
    FILE *hobbyFile = fopen("hobby.txt", "w");
    if (ageFile == NULL || hobbyFile == NULL)
    {
        printf("Input File Could Not Be Opened.");
        exit(1);
    }
    else
    {
        fprintf(ageFile, "Name   Age");
        fprintf(hobbyFile, "Name   Hobby");
        for (int i = 0; i < dataCount; i++)
        {
            fprintf(ageFile, "%-6s %d\n", nameAge[i].name, nameAge[i].age);
            fprintf(hobbyFile, "%-6s %s\n", nameHobby[i].name, nameHobby[i].hobby);
        }
        fclose(ageFile);
        fclose(hobbyFile);
    }
    return 0;
}
```

lab4

- read data from file personal.txt

- generate a struct array that has only name and age, and write to file age.txt

- generate a struct array that has only name and hobby, and write to file hobby.txt

<결과화면>

age.txt, hobby.txt

Lab > lab4 > age.txt

	Name	Age
1	Kim	39
2	Ko	15
3	Lee	17
4	Choi	21
5	Park	10
6	Kang	21
7	Cho	25
8	Heo	39
9	Ra	15
10	Shin	16
11	Jang	15
12	Cha	41
13	Moon	18
14	Jeong	39
15	Yoo	49
16	Mo	24
17	Jeon	21
18	Seol	36
19	Lim	21
20	Oh	29

Lab > lab4 > hobby.txt

	Name	Hobby
1	Kim	Tennis
2	Ko	Soccer
3	Lee	Soccer
4	Choi	Tennis
5	Park	Tennis
6	Kang	Guitar
7	Cho	Piano
8	Heo	Tennis
9	Ra	Piano
10	Shin	Tennis
11	Jang	Soccer
12	Cha	Piano
13	Moon	Tennis
14	Jeong	Soccer
15	Yoo	Tennis
16	Mo	Guitar
17	Jeon	Soccer
18	Seol	Piano
19	Lim	Tennis
20	Oh	Guitar

<lab5-1.c code>

```c
#include <stdio.h>
#include <stdlib.h>

struct salary
{
    char name[20];
    int age;
    double salary;
} salary[100];

int main()
{
    FILE *file;
    int dataCount = 0;

    file = fopen("salary_v1.txt", "r");
    if (file == NULL)
    {
        printf("Input File Could Not Be Opened.\n");
        exit(1);
    }
    else
    {
        while (fscanf(file, "%s %d %lf", salary[dataCount].name, &salary[dataCount].age, &salary[dataCount].salary) !=
EOF)
        {
            dataCount++;
        }
        fclose(file);
    }

    file = fopen("salary_v2.txt", "w");
    if (file == NULL)
    {
        printf("Output File Could Not Be Opened.\n");
        exit(1);
    }
    else
    {
        for (int i = 0; i < dataCount; i++)
        {
            if (salary[i].age >= 40 && salary[i].age < 50)
            {
                salary[i].salary *= 1.1;
            }
            fprintf(file, "%12s %5d %12.1lf\n", salary[i].name, salary[i].age, salary[i].salary);
        }
        fclose(file);
    }

    file = fopen("salary_v2.txt", "r");
    if (file == NULL)
    {
        printf("Input File Could Not Be Opened.\n");
        exit(1);
    }
    else
    {
        int j = 0;
        while (fscanf(file, "%s %d %lf", salary[j].name, &salary[j].age, &salary[j].salary) != EOF)
        {
            j++;
        }
        fclose(file);
    }

    file = fopen("salary_v3.txt", "w");
    if (file == NULL)
    {
        printf("Output File Could Not Be Opened.\n");
        exit(1);
    }
    else
    {

        for (int i = 0; i < dataCount; i++)
        {
            if (salary[i].age >= 30 && salary[i].age < 40)
            {
                salary[i].salary *= 1.2;
            }
            fprintf(file, "%12s %5d %12.1lf\n", salary[i].name, salary[i].age, salary[i].salary);
        }
        fclose(file);
    }

    return 0;
}
```

<lab5-2 code>

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct salary
{
    char name[20];
    int age;
    double salary;
} salary_v1[100], salary_v3[100];

int main()
{
    FILE *file_v1, *file_v3;
    int dataCount_v1 = 0, dataCount_v3 = 0;

    file_v1 = fopen("salary_v1.txt", "r");
    if (file_v1 == NULL)
    {
        printf("Input File Could Not Be Opened.\n");
        exit(1);
    }
    else
    {

        while (fscanf(file_v1, "%s %d %lf", salary_v1[dataCount_v1].name, &salary_v1[dataCount_v1].age,
&salary_v1[dataCount_v1].salary) != EOF)
        {
            dataCount_v1++;
        }
        fclose(file_v1);
    }

    file_v3 = fopen("salary_v3.txt", "r");
    if (file_v3 == NULL)
    {
        printf("Input File Could Not Be Opened.\n");
        exit(1);
    }
    else
    {

        while (fscanf(file_v3, "%s %d %lf", salary_v3[dataCount_v3].name, &salary_v3[dataCount_v3].age,
&salary_v3[dataCount_v3].salary) != EOF)
        {
            dataCount_v3++;
        }
        fclose(file_v3);
    }

    for (int i = 0; i < dataCount_v1; i++)
    {
        for (int j = 0; j < dataCount_v3; j++)
        {
            if (strcmp(salary_v1[i].name, salary_v3[j].name) == 0)
            {
                if (salary_v1[i].salary != salary_v3[j].salary)
                {
                    printf("%-6s %-5d %8.1lf  →  %8.1lf\n", salary_v1[i].name, salary_v1[i].age, salary_v1[i].salary,
salary_v3[j].salary);
                }
            }
        }
    }

    return 0;
}
```

lab5-1

- read data from file salary_v1.txt

- update data so that salaries of employees aged from 40 to 49 are raised 10%

- write to file salary_v2.txt

- read data from file salary_v2.txt

- update data so that salaries of employees aged from 30 to 39 are raised 20%

- write to file salary_v3.txt

lab5-2

- compare salary_v1.txt and salary_v3.txt

- write to screen the difference between two versions as follows

<결과화면>

salary_v1.txt, salary_v2.txt, salary_v3.txt, lab5-2 terminal output

Lab > lab5 > salary_v1.txt

```
 1        Kim    31      34000.0
 2         Ko    48      59000.0
 3       Seol    17      39000.0
 4       Moon    12      86000.0
 5       Choi    10      38000.0
 6       Park    21      37000.0
 7        Lee    25      48000.0
 8        Lim    30      54000.0
 9        Cho    15      64000.0
10        Heo    16      66000.0
11       Kang    15      59000.0
12         Mo    41      88000.0
13       Yoon    18      34000.0
14        Yoo    39      31000.0
15       Jeon    49      78000.0
16       Kwak    24      53000.0
17         Oh    21      43000.0
18       Shin    36      88000.0
19       Jang    20      66000.0
20        Han    29      81000.0
```

Lab > lab5 > salary_v2.txt

```
 1        Kim    31      34000.0
 2         Ko    48      64900.0
 3       Seol    17      39000.0
 4       Moon    12      86000.0
 5       Choi    10      38000.0
 6       Park    21      37000.0
 7        Lee    25      48000.0
 8        Lim    30      54000.0
 9        Cho    15      64000.0
10        Heo    16      66000.0
11       Kang    15      59000.0
12         Mo    41      96800.0
13       Yoon    18      34000.0
14        Yoo    39      31000.0
15       Jeon    49      85800.0
16       Kwak    24      53000.0
17         Oh    21      43000.0
18       Shin    36      88000.0
19       Jang    20      66000.0
20        Han    29      81000.0
```

Lab > lab5 > salary_v3.txt

```
 1        Kim    31      40800.0
 2         Ko    48      64900.0
 3       Seol    17      39000.0
 4       Moon    12      86000.0
 5       Choi    10      38000.0
 6       Park    21      37000.0
 7        Lee    25      48000.0
 8        Lim    30      64800.0
 9        Cho    15      64000.0
10        Heo    16      66000.0
11       Kang    15      59000.0
12         Mo    41      96800.0
13       Yoon    18      34000.0
14        Yoo    39      37200.0
15       Jeon    49      85800.0
16       Kwak    24      53000.0
17         Oh    21      43000.0
18       Shin    36     105600.0
19       Jang    20      66000.0
20        Han    29      81000.0
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\jinyoung\OneDrive\gachon\1_2\problemSoving\code\week11>  & 'c:
ers\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-fvkv2crl.
or-0dyo5t4j.40f' '--pid=Microsoft-MIEngine-Pid-fdwpjzon.ppg' '--dbgExe=C:\
Kim    31     34000.0  →   40800.0
Ko     48     59000.0  →   64900.0
Lim    30     54000.0  →   64800.0
Mo     41     88000.0  →   96800.0
Yoo    39     31000.0  →   37200.0
Jeon   49     78000.0  →   85800.0
Shin   36     88000.0  →  105600.0
PS C:\Users\jinyoung\OneDrive\gachon\1_2\problemSoving\code\week11>
```

<- lab 5-2 terminal output