I. Purpose of the experiment

By using and not using structure arrays, we can write separate programs for the same function. The program functions to store information about the student. The readability of the programming language is understood in depth by comparing the two programs and by combining them with reading the relevant literature.

II. Experimental content

1. Write a small program in a C-based language that uses an array of structs that stores student information including name, age, GPA as a float, and grade level as a string (e.g., "freshmen," etc.).
2. Write the same program in the same language without using structs.
3. Write a brief (at least 200 words) essay on which program better meets readability and why (or if you believe both are equal that's ok too; be sure to justify your answer). Your essay should start with a brief introduction on what you will be discussing as if the reader did not assign this work—this means, the essay (starting with the introduction) should be understandable for readers not in this class.

III. Experimental steps

1. Writing programs using arrays of structures

Step 1: Construct the student structure and build variables in the structure to store student information.

Step 2: Write the function BasedOnStruct() that stores, reads and prints student information, constructs an array of structures of type student in the function, completes the information input by reading it using C language I/O stream files, and prints the display.

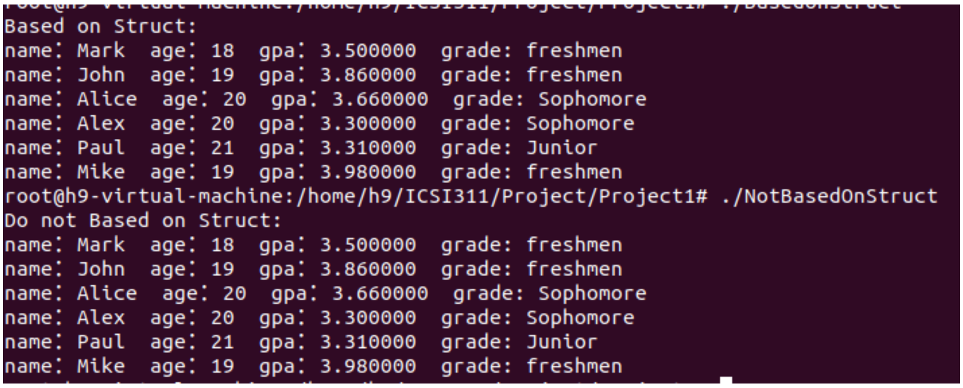
2. Write programs without using structure arrays

Step 1: Construct four separate array variables to store the four pieces of information about the student.

Step 2: Complete the information input by using I/O stream file reading in C language and print the display.

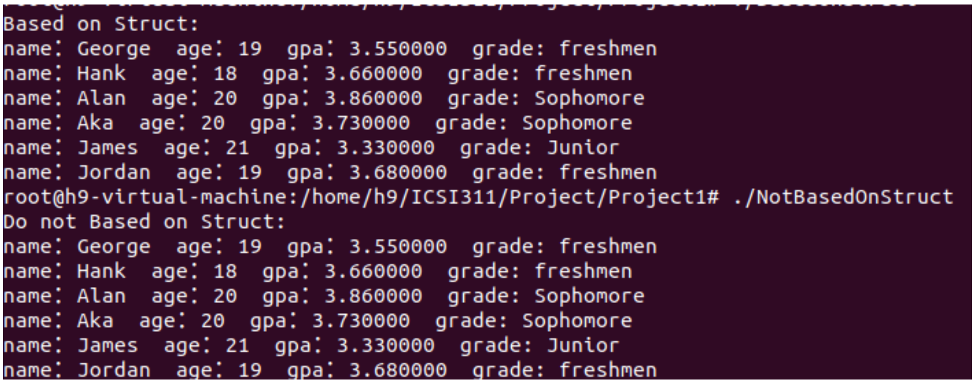
3. Analyze and compare the readability of the two.

IV. Experimental results and analysis



```
Based on Struct:
name: Mark  age: 18  gpa: 3.500000  grade: freshmen
name: John  age: 19  gpa: 3.860000  grade: freshmen
name: Alice  age: 20  gpa: 3.660000  grade: Sophomore
name: Alex  age: 20  gpa: 3.300000  grade: Sophomore
name: Paul  age: 21  gpa: 3.310000  grade: Junior
name: Mike  age: 19  gpa: 3.980000  grade: freshmen
root@h9-virtual-machine:/home/h9/ICSI311/Project/Project1# ./NotBasedOnStruct
Do not Based on Struct:
name: Mark  age: 18  gpa: 3.500000  grade: freshmen
name: John  age: 19  gpa: 3.860000  grade: freshmen
name: Alice  age: 20  gpa: 3.660000  grade: Sophomore
name: Alex  age: 20  gpa: 3.300000  grade: Sophomore
name: Paul  age: 21  gpa: 3.310000  grade: Junior
name: Mike  age: 19  gpa: 3.980000  grade: freshmen
```

Figure 1 Test 1



```
Based on Struct:
name: George  age: 19  gpa: 3.550000  grade: freshmen
name: Hank  age: 18  gpa: 3.660000  grade: freshmen
name: Alan  age: 20  gpa: 3.860000  grade: Sophomore
name: Aka  age: 20  gpa: 3.730000  grade: Sophomore
name: James  age: 21  gpa: 3.330000  grade: Junior
name: Jordan  age: 19  gpa: 3.680000  grade: freshmen
root@h9-virtual-machine:/home/h9/ICSI311/Project/Project1# ./NotBasedOnStruct
Do not Based on Struct:
name: George  age: 19  gpa: 3.550000  grade: freshmen
name: Hank  age: 18  gpa: 3.660000  grade: freshmen
name: Alan  age: 20  gpa: 3.860000  grade: Sophomore
name: Aka  age: 20  gpa: 3.730000  grade: Sophomore
name: James  age: 21  gpa: 3.330000  grade: Junior
name: Jordan  age: 19  gpa: 3.680000  grade: freshmen
```

Figure 2 Test 2

Note:The running video is attached.

Essay:

The evaluation criteria for programming languages include readability, writability, reliability, and cost. Readability is one of the important criteria, which determines whether a program is easy to understand and determines how easy it is to maintain. Next, I will go through two programs to compare their readability. The programs are written in C and store information about students by using both structure arrays and not. Student information including name, grade level , age, GPA.

By comparing the two programs, I believe that using an array of structures is more consistent with the readability of the program for the following reasons:

When not using structures, you need to store information by constructing 4 array variables, and arrays of string types are more complicated to construct in C, involving the use of two-dimensional arrays or pointers. When using structures, you only need to construct a structure array variable and store the corresponding information in the structure.

The latter is easier to read and understand, more in line with overall simplicity, and more in line with the human mind.

When not using structure arrays, you need to always be careful to store the student’s information at the corresponding index of each of the four arrays. In contrast, when using structure arrays, to deposit student information, it is sufficient to deposit it in the corresponding structure. The latter has fewer data types, fewer variables, and is therefore more readable.

In programs that use arrays of structures, we store information about each student through structures, each representing a student. This approach is somewhat similar to object-oriented programming, where each structure in the array is equivalent to an object, and the student information it stores is equivalent to the object’s attributes. And this object-oriented programming thinking is more readable.

V. Experimental experience and gains

1. In-depth understanding of the readability of programming languages, and the evaluation criteria of programming languages.
2. Mastered the different ways of constructing string arrays in C.
3. Become familiar with and further master the use of pointers in the C language.
4. Be familiar with and understand the advantages and disadvantages of C language and other programming languages.

VI. Core Code

Environment：ubuntu20.04

BasedOnStruct.c:

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

/**
 * @File name: BasedOnStruct
 * @Author: H9
 * @Version: 1.1
 * @Date: 2022-10-10
 * @Description: Store and get information of students by using struct
 */
```

```c
/*Struct to store information about student*/
struct student
{
    char name[20];
    int age;
    float gpa;
    char grade[20];
};

/**
 *  Function name ：BasedStruct
 *  Description   : Store  information of students by using struct and print theinformation
 */
void BasedStruct()
{
   struct student stu[10];
   int length = 0;
   // file pointer
   FILE *cfPtr;

   // fopen opens file; exits program if file cannot be opened
   if ((cfPtr = fopen("data1.txt", "r")) == NULL)
   {
      puts("File could not be opened");
      exit(1);
   }
   else
   {
      while (!feof(cfPtr))
      {
    //input information from file
    fscanf(cfPtr, "%s%d%f%s", stu[length].name, &(stu[length].age), &(stu[length].gpa),
stu[length].grade);
    length++;
      }
      // fclose closes the file
      fclose(cfPtr);
   }

   // print the information of student
   for (int i = 0; i < length - 1; i++)
   {
    printf("name：%s  age：%d  gpa：%f  grade：%s\n", stu[i].name, stu[i].age, stu[i].gpa,
stu[i].grade);
```

```c
    }
}

int main()
{
    printf("Based on Struct:\n");
    BasedStruct();
}
```

NotBasedOnStruct.c:

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

/**
 * @File name: NotBasedOnStruct.c
 * @Author: H9
 * @Version: 1.1
 * @Date: 2022-10-10
 * @Description: Store and get information by using others
 */

/**
 *  Function name : NotBasedStruct
 *  Description   : Store information of students by anothe way which not bansedon struct and print the information
 */
void NotBasedStruct()
{
    //Constructing string arrays using two-dimensional arrays
    char name[10][20];
    int age[10];
    float gpa[10];
    char grade[10][20];
    int length = 0;
    // file pointer
    FILE *cfPtr;

    // fopen opens file; exits program if file cannot be opened
    if ((cfPtr = fopen("data1.txt", "r")) == NULL)
    {
            puts("File could not be opened");
            exit(1);
    }
    else
    {
             while (!feof(cfPtr))
             {
             fscanf(cfPtr, "%s%d%f%s", name[length], &(age[length]),
&(gpa[length]),grade[length]);
             length++;
             }

             // fclose closes the file
             fclose(cfPtr);
         // print the information of student
         for (int i = 0; i < length - 1; i++)
         {
             printf("name：%s  age：%d  gpa：%f  grade: %s\n", name[i], age[i], gpa[i],
grade[i]);
         }

    }

}


int main()
{
    printf("Do not Based on Struct:\n");
    NotBasedStruct();
}
```

VII. Reference

1. https://blog.csdn.net/cnds123321/article/details/122973636?spm=1001.2014.3001.5506

2. [12th] Robert W. Sebesta - Concepts of Programming Languages - libgen.lc

3. https://www.sciencedirect.com/topics/engineering/readability

4. https://www.dlsweb.rmit.edu.au/set/Courses/Content/CSIT/oua/cpt220/chapter/01/ProgramReadabilityandUsability.html