模擬卷總結

1819

真的特別簡單，但是時間還是不夠。原因在於想盡一切辦法用自己不熟悉的知識。這份試題的重點在於復習而不是考試。但是現在需要考試模式!！！

采取了開好所有Project並另存爲workspace的方法。但是workspace路徑直接在Desktop/Exam/Shiyao Li中，不太好。建議直接存到桌面。有空的話排版在右手邊，方便點開。

不熟悉的知識點

1. size_t

   時常和`sizeof()`一起使用。

   ```
   int i;
   size_t size = sizeof(i);
   ```

   如果真的要定義一個數組大小，不如直接`#define size 10`

2. rand (Project 8)

   ```
   #include <stdio.h>
   #include <time.h>
   #include <stdlib.h>
   #include <math.h>
   #include <string.h>

   int main()
   {
       int num1, num2;//num1為上限+1，num2為下限
       int num3;//num3為結果
       srand((unsigned)time(NULL));//插入種子
       num3 = rand()%num1 + num2;//num3的取值範圍：[num2, num1]
       return 0;
   }
   ```

3. 天天幻想著從external function return數組/多個數字 (Project 5)

   答：別想了，少想點。老師的指針示例代碼都是通過複製值來傳遞數據的。

問題

1. code::Blocks的問題：Project 3

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

```

```c
#include <string.h>

int main()
{
    int n;
    int i, j;
    //Prompt the user to enter the number of * on the longest line N
    puts("Please enter the number N of '*' on the longest line N");
    scanf("%d", &n);
    //printf("%d\n", n);//這一行出大問題
    for(i = n; i > 0; i--){
        for(j = i; j > 0; j--){
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```

當沒有`printf("%d\n", n);`這一行的時候，代碼直接終止運行。當有這一行時，代碼正常運行。

答： code::Blocks中所有的`scanf()`後面最好都要加上一個`getchar();`，防止誤輸入。

2. 缺失.txt文件

   Project 7和Project 9都缺失題目中對應的.txt文件。

3. `^Z`有誤解

   爲什麼以下這段代碼要退出兩次才算退出？如何修改它？爲什麼會將最後一位數字儲存兩遍？

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

   int main()
   {
       FILE *fPtr;
       int nums[50];
       fPtr = fopen("test.txt", "w");
       for(int i = 0; i < 50 && !feof(stdin); i++){
           while(!feof(stdin)){
               scanf("%d", &nums[i]);
               fprintf(fPtr, "%d\t", nums[j]);
           }
       }

       return 0;
   }
   ```

答:

1. 可以通過fprint少一位來解決（是i++的問題， 換成++i后最後一問變成了0）

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>
#include <math.h>

int main()
{
    FILE *fPtr;
    int nums[50];
    int i = 0;
    fPtr = fopen("test.txt", "w");
        while(!feof(stdin)){
            scanf("%d", &nums[i]);
            getchar();
            i += 1;
        }
    for(int j = 0; j < i-1; ++j){
            fprintf(fPtr, "%d\t", nums[j]);
    }

    return 0;
}
```

2. gpt更改

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

int main() {
    FILE *fPtr;
    int *nums = NULL;
    int i = 0;
    int num;

    // 開啟檔案
    fPtr = fopen("test.txt", "w");
    if (fPtr == NULL) {
        printf("無法打開檔案\n");
        return 1;
    }

    // 輸入數字，動態分配內存
    while (scanf("%d", &num) == 1) {
        nums = realloc(nums, (i + 1) * sizeof(int));
        if (nums == NULL) {
            printf("內存分配失敗\n");
            fclose(fPtr);
            return 1;
        }
        nums[i] = num;
        i++;
    }

    // 將數字寫入檔案
    for (int j = 0; j < i; ++j) {
        fprintf(fPtr, "%d\t", nums[j]);
    }

    // 釋放動態分配的內存
    free(nums);

    // 關閉檔案
    fclose(fPtr);

    return 0;
}
```

總結：

1. 所有`scanf()`后都要加`getchar()`以防輸入報錯
2. 開闢9個project，并且將所有workspace儲存在桌面右手位。**考試時候一定要這麽做**，模擬的時候還是建立在文件夾裏，不然好麻煩我説真的。
3. 先`int *nums;`再`nums = realloc(dest, (count) * sizeof(src));`最後后接上`free(dest);`

1920

時間是絕對夠用的。主要問題還是出在了file上。

Q7

```
#include <stdio.h>

int main() {
    FILE *file;
    file = fopen("Q7_mark.txt", "w");

    if (file == NULL) {
        printf("Error opening file!\n");
        return 1;
    }
    printf("Enter the first name and mark of a student. Press Ctrl+Z (Windows) or Ctrl+D (Unix) to finish:\n");

    char firstName[100];
    double mark;

    // Read input until EOF (Ctrl+Z or Ctrl+D) is encountered
    while (scanf("%s %lf", firstName, &mark) != EOF) {
        fprintf(file, "%s %.2lf\n", firstName, mark);
    }

    fclose(file);

    printf("Data written to Q7_mark.txt.\n");

    return 0;
}
```

Q8

```
#include <stdio.h>

// Function to allow the user to enter a series of numbers and write them to a file
void enterNumbersToFile() {
    FILE *file;
    file = fopen("UserNumber.txt", "w");

    if (file == NULL) {
        printf("Error opening file!\n");
        return;
    }

    printf("Enter a series of numbers, one per line. Press Ctrl+Z (Windows) or Ctrl+D (Unix) to finish:\n");

    double number;
    while (scanf("%lf", &number) != EOF) {
        fprintf(file, "%lf\n", number);
    }

    fclose(file);
}

// Function to read data from the file, calculate sum, and find max and min values
void readDataFromFile(double *sum, double *max, double *min) {
    FILE *file;
    file = fopen("UserNumber.txt", "r");

    if (file == NULL) {
        printf("Error opening file!\n");
        return;
    }

    double number;
    if (fscanf(file, "%lf", &number) == EOF) {
        printf("No numbers found in the file.\n");
        fclose(file);
        return;
    }

    *sum = *max = *min = number;

    while (fscanf(file, "%lf", &number) != EOF) {
        *sum += number;

        if (number > *max) {
            *max = number;
        }

        if (number < *min) {
            *min = number;
        }
    }

    fclose(file);
}

int main() {
    double sum, max, min;

    // Step (a)
    enterNumbersToFile();

    // Step (b)
    readDataFromFile(&sum, &max, &min);

    // Step (c)
    printf("Sum: %.2lf\n", sum);
    printf("Maximum Value: %.2lf\n", max);
    printf("Minimum Value: %.2lf\n", min);

    return 0;
}
```

`typedef`

```
// 使用 typedef 为结构体命名
typedef struct {
    int id;
    char name[80];
    float percentage;
} Student; // 使用 Student 替代 struct students

int main() {
    // 使用 typedef 后的结构体声明数组
    Student toomany[80]; // 一个有80个学生的struct数组
}
```

總結： The input from the user will stop until the user enters EOF (ctrl+z) 要用scanf(format, add) != EOF來滿足，否則就用realloc

2020

總結： 所有的FILE讀寫都可以一次性輸入，不用數組。要讀完一部分題目再整體思考代碼。

實例：2020Q2 a)-d)是一起編寫的。

```
    FILE *classListFile, *updatedListFile;
    Student student;
    char newStudent;

    // Open the existing class list file
    classListFile = fopen("ClassList.txt", "r");
    if (classListFile == NULL) {
        perror("Error opening ClassList.txt");
        return 1;
    }

    // Open the updated class list file
    updatedListFile = fopen("ClassListUpdated.txt", "w");
    if (updatedListFile == NULL) {
        perror("Error opening ClassListUpdated.txt");
        fclose(classListFile);
        return 1;
    }

    while (fscanf(classListFile, "%s", student.name) != EOF) {
        printf("Enter gender for %s (M/F): ", student.name);
        scanf(" %c", &student.gender);
        validateGender(&student.gender);

        printf("Enter height for %s (100-220): ", student.name);
        scanf("%d", &student.height);
        validateHeight(&student.height);
        // Write the updated information to ClassListUpdated.txt
        fprintf(updatedListFile, "%s, %c, %d\n", student.name, student.gender,
student.height);
    }
```

第二題是非常典型的題目。

所有指針struct都有考到。用append也可以直接write也可以。建議考前多看看這份題目和代碼

Q2 Assume you are one academic staff in UNNC and will deliver the module of EEEE1044. A class list is given to you in the beginning of autumn semester, now develop a program which meets the following requirements.

a) Read from the class list, which is given on the desktop named as “ClassList.txt”. The names are written one student per line.

b) For each student, request the keyboard input from the user for the information below: o Student’s gender o Student’s height (in centimetres)

c) Your program should check the inputs from the user following the requirements below:

1. Student’s gender can only be “M” “m” or “F” “f”, any other inputs should be considered as invalid input and should not be recorded. A valid input should be obtained for each student.
2. Student’s height in centimetres should be within the range of 100 – 220, any input out of this range should be considered as invalid input and should not be recorded. A valid input should be obtained for each student.

d) Open a new file named “ClassListUpdated.txt”, and write the student’s name, gender (valid) and height (valid) into ClassListUpdated.txt, one student per line following the format of the example below. MeimeiHan, F, 165

e) After completing reading ClassList.txt, prompt the following message to user in order to check if there are any students newly enrolled to EEEE1044. Any new student? (Y/N)

f) If there is no new student, go to step i).

g) If there is new student(s), do the following:

1. Prompt the user to enter the name (the name should follow the format in ClassList.txt, assuming no space is entered from user, and no need to check if there is any space entered), gender and height of each new student, and continue to write the information to ClassListUpdated.txt file, following the requirements in step c) and d).
2. Write the name(s) of new student(s) to ClassList.txt, appending to the original names in that file.

h) The input from the user will stop until the user enters EOF (ctrl+z).

i) For all the students both in ClassList.txt and those entered by user if any, do the following counting and calculation, then display all the results properly to the screen.

1. count the overall number of students
2. count the number of male students
3. count the number of female students

4. calculate the average height of all male students

5. calculate the average height of all female students

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

#define MAX_NAME_LENGTH 50
#define FILENAME "ClassList.txt"
#define UPDATED_FILENAME "ClassListUpdated.txt"

typedef struct {
    char name[MAX_NAME_LENGTH];
    char gender;
    int height;
} Student;

void writeStudentToFile(FILE *file, const Student *student) {
    fprintf(file, "%s, %c, %d\n", student->name, student->gender, student->height);
}

void validateGender(char *gender) {
    while (*gender != 'M' && *gender != 'm' && *gender != 'F' && *gender != 'f') {
        printf("Invalid gender. Please enter 'M' or 'F': ");
        scanf(" %c", gender);
    }
}

void validateHeight(int *height) {
    while (*height < 100 || *height > 220) {
        printf("Invalid height. Please enter a height between 100 and 220: ");
        scanf("%d", height);
    }
}

int main() {
    FILE *classListFile, *updatedListFile;
    Student student;
    char newStudent;
    int overallCount = 0, maleCount = 0, femaleCount = 0;
    int totalMaleHeight = 0, totalFemaleHeight = 0;

    // Open the existing class list file
    classListFile = fopen("ClassList.txt", "r");
    if (classListFile == NULL) {
        perror("Error opening ClassList.txt");
        return 1;
    }

    // Open the updated class list file
    updatedListFile = fopen("ClassListUpdated.txt", "w");
    if (updatedListFile == NULL) {
        perror("Error opening ClassListUpdated.txt");
        fclose(classListFile);
        return 1;
    }

    // Read from ClassList.txt and update the information
    while (fscanf(classListFile, "%s", student.name) != EOF) {
        printf("Enter gender for %s (M/F): ", student.name);
        scanf(" %c", &student.gender);
        validateGender(&student.gender);

        printf("Enter height for %s (100-220): ", student.name);
        scanf("%d", &student.height);
        validateHeight(&student.height);

        // Write the updated information to ClassListUpdated.txt
        writeStudentToFile(updatedListFile, &student);

        overallCount++;

        if (student.gender == 'M' || student.gender == 'm') {
            maleCount++;
            totalMaleHeight += student.height;
        } else {
            femaleCount++;
            totalFemaleHeight += student.height;
        }
    }

    // Prompt for new students
    printf("Any new student? (Y/N): ");
    scanf(" %c", &newStudent);
    getchar();
    while(newStudent != 'Y' && newStudent != 'y' && newStudent != 'n' && newStudent
!= 'N'){
        puts("Invalid input. Please try again");
        scanf("%c", &newStudent);
        getchar();
    }

    // Process new students
    while ((newStudent == 'Y' || newStudent == 'y') && scanf("%s %c %d",
&student.name, &student.gender, &student.height) != EOF) {
        // Write the new student information to ClassListUpdated.txt
        writeStudentToFile(updatedListFile, &student);
        overallCount++;

        if (student.gender == 'M' || student.gender == 'm') {
            maleCount++;
            totalMaleHeight += student.height;
        } else {
            femaleCount++;
            totalFemaleHeight += student.height;
        }
    }

    // Close files
    fclose(classListFile);
    fclose(updatedListFile);

    printf("\nOverall Number of Students: %d\n", overallCount);
    printf("Number of Male Students: %d\n", maleCount);
    printf("Number of Female Students: %d\n", femaleCount);

    printf("Average Height of Male Students: %.2f\n", (float)totalMaleHeight / maleCount);
    printf("Average Height of Female Students: %.2f\n", (float)totalFemaleHeight / femaleCount);


    return 0;
}
```

突然意識到之前沒給文檔的題目其實大概率特別有價值。明天去看電影，後天自創文檔重新做。

2122

2223

時間還是不夠用。最大問題在於這個b黑軸鍵盤真tnd難用。

建議：每次的函數庫+scanf()後面需要加getchar()這樣的操作感覺可以複製到其它軟件裏。

超大份總結

1. 所有`scanf()`后都要加`getchar()`以防輸入報錯
2. 開闢9個project，并且將所有workspace儲存在桌面右手位。**考試時候一定要這麽做**，模擬的時候還是建立在文件夾裏，不然好麻煩我説真的。
3. 對於所有不知道大小的數組，先`int *nums;`再`nums = realloc(dest, (count) * sizeof(src));`最後后接上`free(dest);`
4. `switch(a){ case 'b': Action; break;}`和`if...else if`的分數會比多個`if`要高。
5. `while`分比`if`高。
6. 當有空時一定要改輸入限制檢查！！！（2223Q3)
7. ~~don't save invalid input (2223Q5)~~

   把我害慘了的一條定義。scanf輸入後重新覆蓋就行了，沒那麽多講究。
8. 最好再打開一個別的編輯軟件，把所有庫和scanf()後面需要加getchar()這樣的定式操作給儲存一下。

9. The input from the user will stop until the user enters EOF (ctrl+z) 要用scanf(format, add) != EOF 來滿足
10. 要讀完一模塊題目再進行代碼編寫。

Chat救我狗命

1. terminate cmd指令快速速打開codeB::locks：有，但是要找到具體下載路徑（明天下午兩點到四點學校機房去看看）
2. win上運行.log文件的方法：Arduino的log只是儲存編譯信息的，不能像innovus和ROS一樣直接執行。