实验五、六 函数和指针

1. 实验目的
2. 掌握函数定义的方法。
3. 掌握函数实参和形参的对应关系，以及“值传递”的方式。
4. 掌握函数的嵌套调用和递归调用的方法。
5. 掌握全局变量和局部变量、动态变量和静态变量的概念和使用方法。
6. 理解和掌握多模块的程序设计与调试的方法。
7. 熟练掌握指针、地址、指针类型等概念
8. 熟练掌握指针变量的定义、初始化、指针间接访问方式、指针的加减运算和指针表达式。
9. 学会使用数组的指针。
10. 学会使用字符串指针。
11. 实验准备
12. 复习函数调用的基本知识
13. 复习函数的嵌套调用和递归调用的方法。
14. 复习全局变量和局部变量、动态变量和静态变量等概念的使用方法。
15. 复习变量、变量的地址、指针变量的概念并且明确区分它们的不同。
16. 复习指针和数组的结合运用。
17. 实验步骤及内容

第8章习题1，3，6，12，15，所有题目均要求模块化设计，函数调用实现，下面给出了题目的函数接口定义，编程时请按照这个格式来定义自己的函数。

1，int swap(int \*, int \*)，函数返回形参中的较大值。

3. 写3个函数，第一个参数传入数组地址，第二个参数传入数组长度

①输人10个数; void input(int \*, int )

②进行处理; void print(int \*, int )

* 1. 输出10个数。void handle(int \*, int)

6. int mystrlen(char \*str)

12. void sort(char \*[]，int)，主函数定义第一个实参为指针数组，第二个参数为数组长度。

15．float avg(int (\*)[5], int n)，第一个形参为指向元素的地址，第二个形参传入学生人数

void fail(int (\*)[5], int n)，第一个形参为指向行（指向一维数组）的地址，第二个形参传入学生人数

void excellent(int (\*)[5], int n)，第一个形参为指向行（指向一维数组）的地址，第二个形参传入学生人数

四、实验报告

1. 源程序。

(1)

#include <stdio.h>

int swap(int\*, int\*);

int main() {

// 输入3个整数，要求按由小到大的顺序输出

// 201833050025 成凯

int a, b, c;

printf("Enter three numbers:");

scanf("%d %d %d", &a, &b, &c);

if (a > b) {

swap(&a, &b);

}

if (a > c) {

swap(&a, &c);

}

if (b > c) {

swap(&b, &c);

}

printf("%d %d %d\n", a, b, c);

return 0;

}

int swap(int\* a, int\* b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

// 返回较大值

return \*a > \*b ? \*a : \*b;

}

(3)

#include "stdio.h"

#define N 10

void input(int\*, int);

void print(const int\*, int);

void handle(int\*, int);

int main() {

// 输入10个整数,将其中最小的数与第一个数对换, 把最大的数与最后一个数对换

// 201833050025 成凯

int arr[N];

printf("Enter ten nums: ");

input(arr, N);

handle(arr, N);

print(arr, N);

return 0;

}

/\*\*

\* 输入10个数

\*/

void input(int\* arr, int len) {

for (int i = 0; i < len; ++i) {

scanf("%d", arr + i);

}

}

/\*\*

\* 输出10个数

\*/

void print(const int\* arr, int len) {

for (int i = 0; i < len; ++i) {

printf("%d ", \*(arr + i));

}

printf("\n");

}

/\*\*

\* 进行处理

\*/

void handle(int\* arr, int len) {

int\* max = arr;

int\* min = arr;

for (int i = 0; i < len; ++i) {

int num = \*(arr + i);

if (num > \*max) {

max = arr + i;

}

if (num < \*min) {

min = arr + i;

}

}

printf("min=%d, max=%d\n", \*min, \*max);

int temp;

temp = \*arr;

\*arr = \*min;

\*min = temp;

temp = \*(arr + len - 1);

\*(arr + len - 1) = \*max;

\*max = temp;

}

(6)

#include <stdio.h>

int mystrlen(char\*);

int main() {

// 写一函数,求一个字符串的长度。在main函数中输入字符串,并输出其长度

// 201833050025 成凯

char string[20];

gets(string);

printf("len: %d", mystrlen(string));

return 0;

}

int mystrlen(char\* str) {

int len = 0;

while (\*str != '\0') {

str++;

len++;

}

return len;

}

(12)

#include <stdlib.h>

#include <string.h>

#include "stdio.h"

#define N 10

void sort(char\* [], int);

int main() {

// 在主函数中输入10个等长的字符串。用另一函数对它们排序。然后在主函数输出这10个已排好序的字符串

// 用指针数组处理上一题目,字符串不等长

// 201833050025 成凯

char\* string[N];

printf("Enter ten strings:\n");

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

string[i] = (char\*) malloc(20);

gets(string[i]);

}

sort(string, N);

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

printf("%s\n", string[i]);

}

return 0;

}

void sort(char\* string[], int len) {

char\* temp;

for (int i = 0; i < len; ++i) {

for (int j = i; j < len; ++j) {

if (strcmp(string[i], string[j]) > 0) {

temp = string[i];

string[i] = string[j];

string[j] = temp;

}

}

}

}

(15)

#include <stdlib.h>

#include <stdio.h>

// 课程数

#define COURSE\_COUNT 5

// 学生数

#define STU\_COUNT 4

float avg(int (\*)[COURSE\_COUNT], int);

void fail(int (\*)[COURSE\_COUNT], int);

void excellent(int (\*)[COURSE\_COUNT], int);

int main() {

// 201833050025 成凯

int (\* arr)[COURSE\_COUNT];

arr = (int (\*)[COURSE\_COUNT]) malloc(sizeof(int\*) \* COURSE\_COUNT);

// 成绩信息

printf("输入学生成绩：\n");

for (int i = 0; i < STU\_COUNT; i++) {

for (int j = 0; j < COURSE\_COUNT; j++) {

scanf("%d", \*(arr + i) + j);

}

}

printf("第一门课平均分：%.2f\n", avg(arr, STU\_COUNT));

fail(arr, STU\_COUNT);

excellent(arr, STU\_COUNT);

return 0;

}

/\*\*

\* 求第一门课的平均分

\*

\* @param arr 指向元素的地址

\* @param stuCount 学生人数

\*/

float avg(int (\* arr)[COURSE\_COUNT], int stuCount) {

float sum = 0;

for (int i = 0; i < stuCount; ++i) {

sum += (float) \*(arr[i]);

}

return sum / (float) stuCount;

}

/\*\*

\* 找出有两门以上课程不及格的学生,输出他们的学号和全部课程成绩及平均成绩

\*

\* @param stuCount 学生人数

\*/

void fail(int (\* arr)[COURSE\_COUNT], int stuCount) {

printf("不及格学生：\n");

for (int i = 0; i < stuCount; i++) {

int failC = 0;

// 找不及格课程的数量

for (int j = 0; j < COURSE\_COUNT; j++) {

if (arr[i][j] < 60) {

failC++;

}

}

if (failC <= 2) {

continue;

}

printf("学号：%d ", i + 1);

printf("全部课程成绩: ");

int sum = 0;

for (int j = 0; j < COURSE\_COUNT; j++) {

sum += arr[i][j];

printf("%d ", arr[i][j]);

}

printf("平均分：%d ", sum / COURSE\_COUNT);

printf("\n");

}

}

/\*\*

\* 找出平均成绩在90分以上或全部课程成绩在85分以上的学生

\*

\* @param stuCount 学生人数

\*/

void excellent(int (\* arr)[COURSE\_COUNT], int stuCount) {

printf("优秀学生：\n");

for (int i = 0; i < stuCount; i++) {

int sum = 0, count = 0;

for (int j = 0; j < COURSE\_COUNT; j++) {

sum += arr[i][j];

if (arr[i][j] > 85) {

count++;

}

}

if ((sum / COURSE\_COUNT) > 90 || count == COURSE\_COUNT) {

printf("学号：%d\n", i + 1);

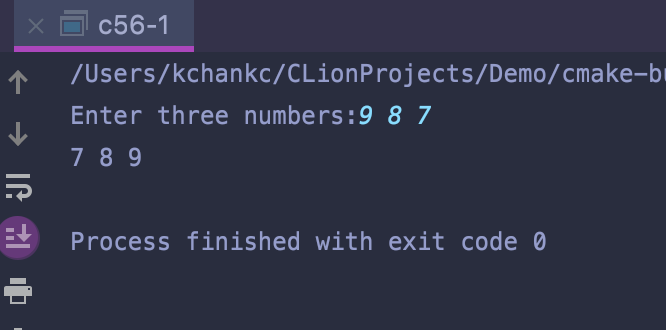
}

}

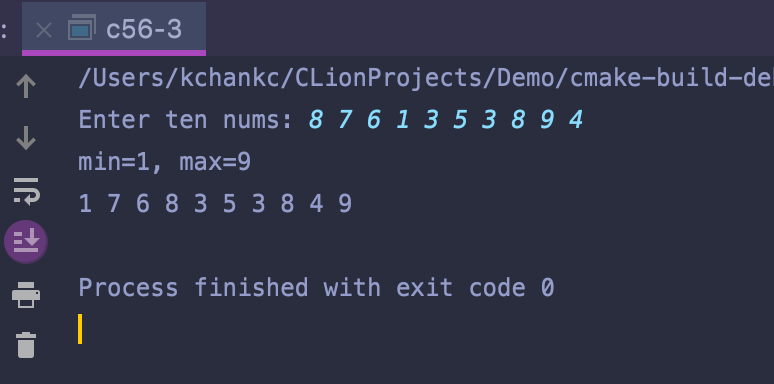
}

1. 编译调试过程中出现的错误及修改情况，分析错误原因。
2. 实验结果截屏。

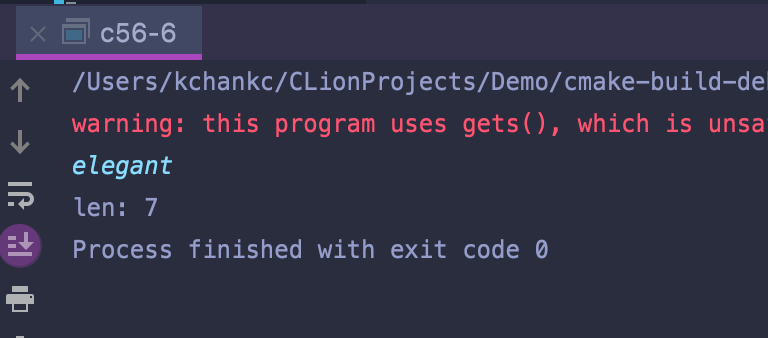
(1)



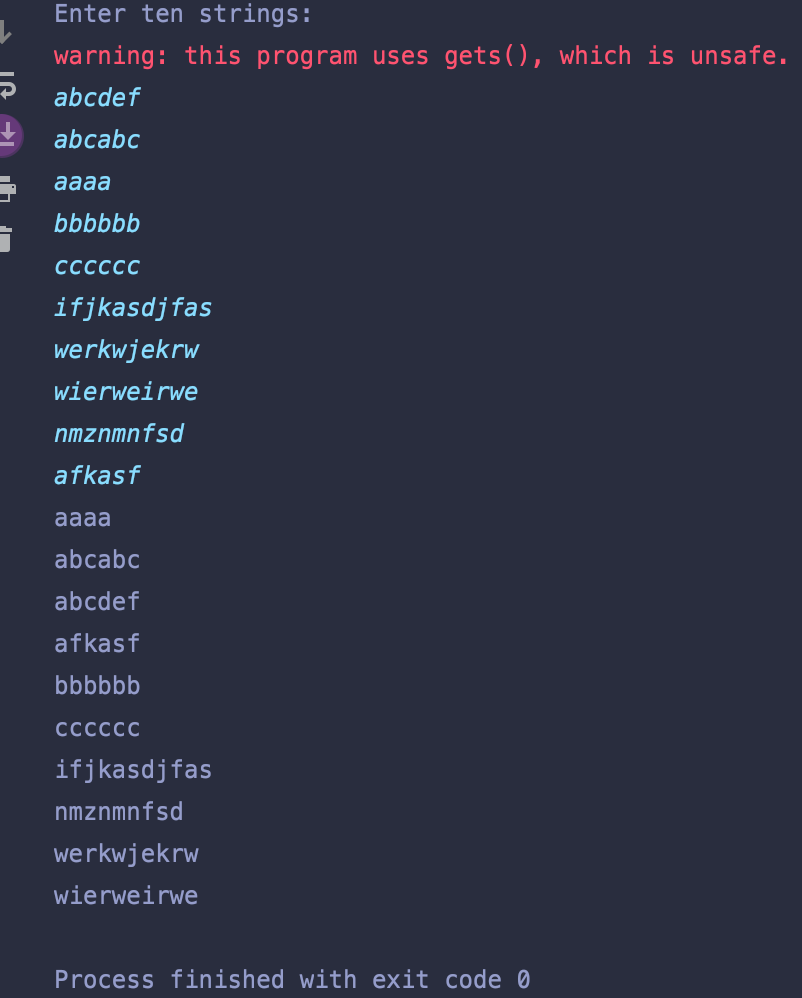
(3)



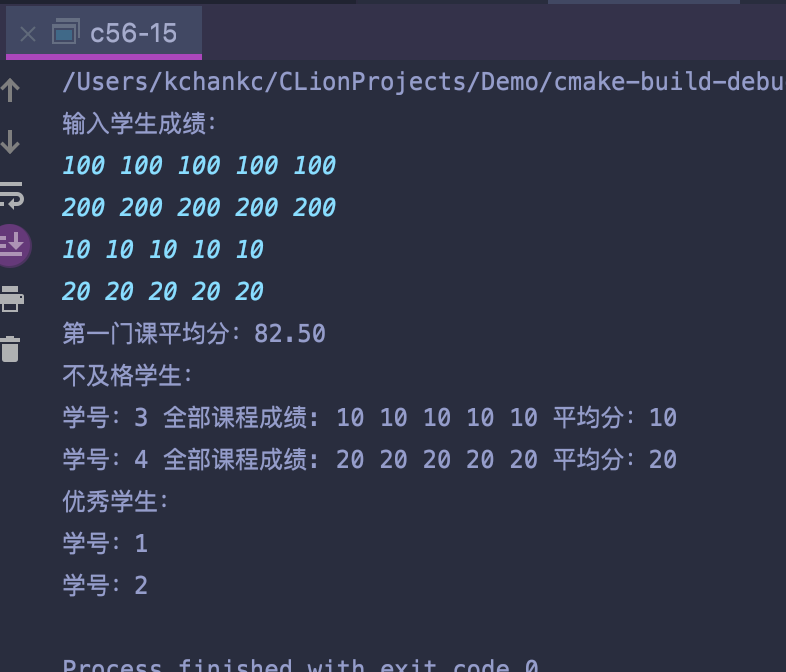
(6)



(12)



(15)



1. 实验感受及体会。

通过本次实验，掌握了如何定义函数，以及函数实参和形参之间的关系。理解了函数的嵌套调用和递归调用的方法。掌握指针、地址、指针类型之间的关系。熟练掌握指针变量的定义、初始化、指针间接访问方式、指针的加减运算和指针表达式。掌握了数组指针和指针数组以及字符串指针。