# 实验6 指针实验

6.1、实验目的

1．熟练掌握指针的说明、赋值、使用。

2．掌握用指针引用数组的元素，熟悉指向数组的指针的使用。

3．熟练掌握字符数组与字符串的使用，掌握指针数组及字符指针数组的用法。

4．掌握指针函数与函数指针的用法。

5．掌握带有参数的main函数的用法。

6.2、实验内容

**6.2.1．源程序改错题**

下面程序是否存在错误？如果存在，原因是什么？如果存在错误，要求在计算机上对这个例子程序进行调试修改，使之能够正确执行。

#include "stdio.h"

void main(void)

{

float \*p;

scanf("%f",p);

printf("%f\n",\*p);

}

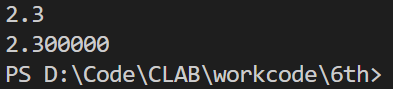
**解答：**

（1）错误修改：

第4行使用了未初始化的变量p，正确形式为：

float a,\*p=&a;

（2）错误修改后运行结果如图6-1所示：



**图6-1 改错题的测试的运行结果**

**6.2.2．源程序修改替换题**

**(1)** 下面的程序通过函数指针和菜单选择来调用字符串拷贝函数或字符串连接函数，请在下划线处填写合适的表达式、语句、或代码片段来完善该程序。

#include <stdio.h>

#include <string.h>

int main(void)

{

char \*(\*p)(char \*, const char \*);

char a[80], b[80], c[160], \*result = c;

int choice, i;

do

{

printf("\t\t1 copy string.\n");

printf("\t\t2 connect string.\n");

printf("\t\t3 exit.\n");

printf("\t\tinput a number (1-3) please!\n");

scanf("%d", &choice);

} while (choice < 1 || choice > 5);

switch (choice)

{

case 1:

p = strcpy;

break;

case 2:

p = strcat;

break;

case 3:

goto down;

}

getchar();

printf("input the first string please!\n");

i = 0;

gets(a);

printf("input the second string please!\n");

i = 0;

gets(b);

result = p(a, b);

printf("the result is %s\n", result);

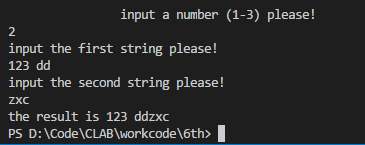
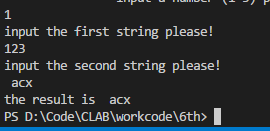
down:

;

return 0;

}

运行结果：



**图6-2 替换题（1）的测试的运行结果**

**6.2.3．跟踪调试题**

#include "stdio.h"

char \*strcpy(char \*,char \*);

void main(void)

{

char a[20],b[60]="there is a boat on the lake.";

printf("%s\n",strcpy(a,b));

}

char \*strcpy(char \*s,char \*t)

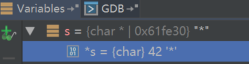
{

while(\*s++=\*t++);

return (s);

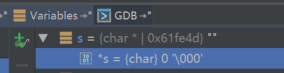
}

1. 单步执行。进入strcpy时watch窗口中s为何值？



**图6-4 跟踪调试题测试结果（1）**

返回main时, watch窗口中s为何值？



**图6-5 跟踪调试题测试结果（2）**

（2）排除错误，使程序输出结果为：

there is a boat on the lake.

程序代码：

#include "stdio.h"

char \*strcpy(char \*,char \*);

int main(void)

{

char a[20],b[60]="there is a boat on the lake.";

printf("%s\n",strcpy(a,b));

return 0;

}

char \*strcpy(char \*s,char \*t)

{

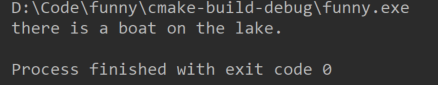
char \*a = s;

while(\*s++=\*t++);

return (a);

}

运行结果：



**图6-6 跟踪调试题测试运行结果**

**6.2.4．编程设计题**

**（1）**一个长整型变量占4个字节，其中每个字节又分成高4位和低4位。试从该长整型变量的高字节开始，依次取出每个字节的高4位和低4位并以数字字符的形式进行显示。

1）解题思路：

利用指向char型的指针取出长整型变量中的单个字节，再分别取出高低4位。

2）源程序清单

#include<stdio.h>

int main(void){

long int l = 0xabcd1234;

char \*c = (char \*)&l;

char ch[8];

for (int i = 0; i < 8; i+=2){

ch[i] = (\*c & 0xf0) >> 4;

ch[i + 1] = \*c & 0xf;

\*(c++);

}

for (int i = 0; i < 8; i++){

if(ch[i] >= 0 && ch[i] <= 9)

printf("%c ", ch[i] + '0');

else if(ch[i] >= 0xa && ch[i] <= 0xf)

printf("%c ", ch[i] + 'a' - 10);

}

return 0;

}

3）测试

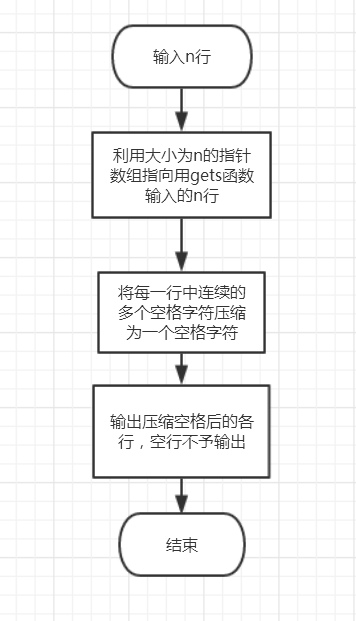


**图6-7 编程题1的测试运行结果**

1. 利用大小为n的指针数组指向用gets函数输入的n行，每行不超过80个字符。编写一个函数，它将每一行中连续的多个空格字符压缩为一个空格字符。在调用函数中输出压缩空格后的各行，空行不予输出。

**解答：**

1. 算法流程如图所示



**图6-8 编程题2的程序流程图**

1. 源程序清单

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

void fvc(char \*a);

int main(void){

char a[100];

while(gets(a))

fvc(a);

return 0;

}

void fvc(char \*a){

char \*ch;

int flag = 1;

if(strlen(a) == 0)

return;

for (ch = a; \*ch; ++ch){

if (\*ch == '\n')

return;

else if (\*ch != ' ' && \*ch != '\t' && \*ch != '\n'){

putchar(\*ch);

flag = 1;

}

else if (flag == 1){

putchar(' ');

flag = 0;

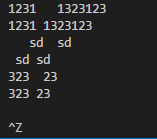
}

}

putchar('\n');

}

3)测试



**图6-9 编程题2的测试运行结果**

**（3）**设某个班有N个学生，每个学生修了M门课程（用#define定义N、M）。输入M门课程的名称，然后依次输入N个学生中每个学生所修的M门课程的成绩并且都存放到相应的数组中。编写下列函数：

a.计算每个学生各门课程平均成绩；

b.计算全班每门课程的平均成绩；

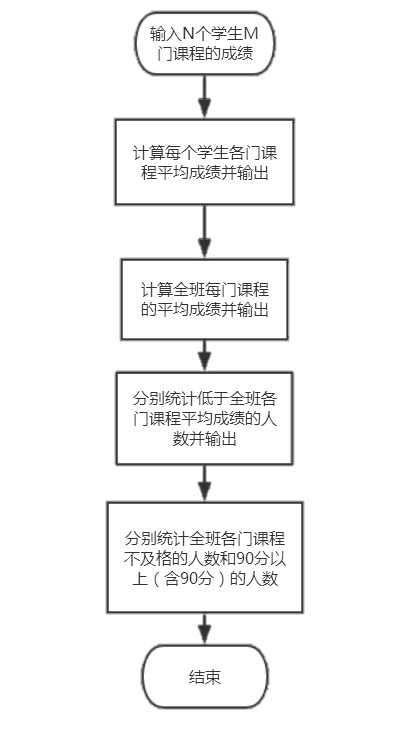
c.分别统计低于全班各门课程平均成绩的人数；

d.分别统计全班各门课程不及格的人数和90分以上（含90分）的人数。

在调用函数中输出上面各函数的计算结果。（要求都用指针操作，不得使用下标操作。）

**解答：**

1. 算法流程如图所示



**图6-10 编程题3的程序流程图**

1. 源程序清单

#include<stdio.h>

#define N 30

#define M 6

#define MAXNAME 20

double StuAverage(double (\*grade)[M], int stunum, int subsum);

double SubAverage(double (\*grade)[M], int stusum, int subnum);

int Lower(double (\*grade)[M], int stusum, int subnum);

int Failed(double (\*grade)[M], int stusum, int subnum);

int Excel(double (\*grade)[M], int stusum, int subnum);

int main(void){

double grade[N][M] = {};//N为最大学生数，M为最大科目数

char subname[M][MAXNAME] = {};

int subsum = 1, stusum = 1;

int subnum, stunum;

do{

printf("\nPlease input the amount of subject and students:(0 to quit)");

scanf("%d%d", &subsum, &stusum);

if(subsum==0)

break;

printf("Input the names of those subs\n");

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

scanf("%s", subname[i]);

}

printf("Input the grade of every one\n");

for (int i = 0; i < stusum; i++)

for (int j = 0; j < subsum; j++)

scanf("%lf", &grade[i][j]);

printf("Whose Average?\n");

scanf("%d", &stunum);

printf("this classmate's average is%lf\n", StuAverage(grade, stunum, subsum));

printf("Which Subject\n");

scanf("%d", &subnum);

printf("the SubAverage is %lf\n", SubAverage(grade, stusum, subnum));

printf("there is %d student(s) lower than the subaverage.\n", Lower(grade, stusum, subnum));

printf("And %d student(s) failed, %d student(s) gets a A.\n", Failed(grade, stusum, subnum), Excel(grade, stusum, subnum));

} while (subsum);

return 0;

}

double StuAverage(double (\*grade)[M], int stunum, int subsum){

double result, sum = 0;

for (int i = 0; i < subsum; i++)

sum += \*(\*(grade+stunum-1)+i);

result = sum / (double)subsum;

return result;

}

double SubAverage(double (\*grade)[M], int stusum, int subnum){

double result, sum = 0;

for (int i = 0; i < stusum; i++)

sum += \*(\*(grade+i)+subnum-1);

result = sum / (double)stusum;

return result;

}

int Lower(double (\*grade)[M], int stusum, int subnum){

double Average = SubAverage(grade, stusum, subnum);

int result = 0;

for (int i = 0; i < stusum; i++)

if(\*(\*(grade+i)+subnum-1) <= Average)

result++;

return result;

}

int Failed(double (\*grade)[M], int stusum, int subnum){

int result = 0;

for (int i = 0; i < stusum; i++)

if(\*(\*(grade+i)+subnum-1) < 60)

result++;

return result;

}

int Excel(double (\*grade)[M], int stusum, int subnum){

int result = 0;

for (int i = 0; i < stusum; i++)

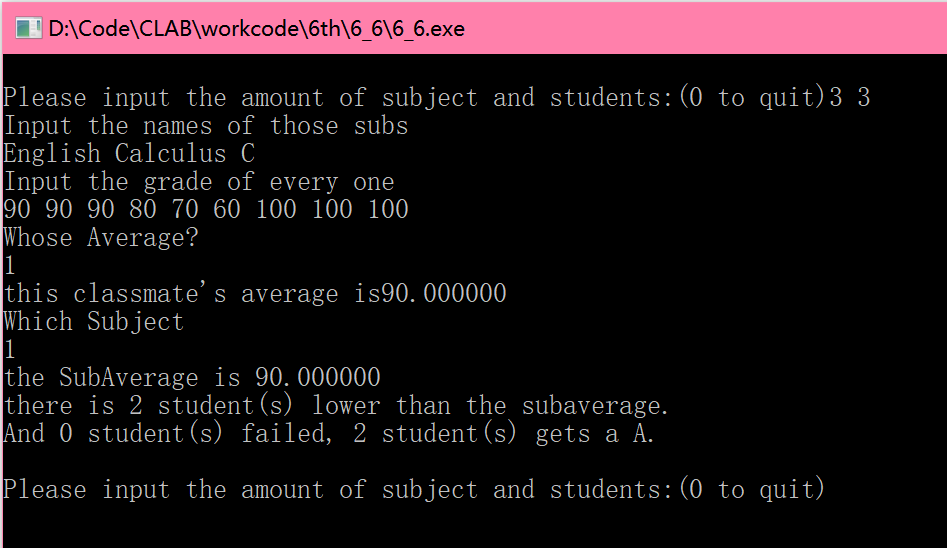
if(\*(\*(grade+i)+subnum-1) >= 90)

result++;

return result;

}

3)测试



**图6-11 编程题3的测试运行结果**

**6.2.5选做题**

（1）设有N位整数和M位小数（N=20，M=10）的数据a,b。编程计算a+b并输出结果。

如：12345678912345678912.1234567891 + 98765432109876543210.0123456789

1. 解题思路：

利用字符串存储输入的大数以及运算结果，按位对齐之后相加，并记录是否进位。

2）程序清单：

#include<stdio.h>

#include<string.h>

void reverse(char \*string, int length);

int main(void){

char num1[50] = {}, num2[50] = {}, sum[55] = {};

int length1, length2, length3;

int acc = 0;

int i, bit;

printf("please input a addition expression(^z to quit)");

while(scanf("%s %s", num1, num2) != EOF){

for (int j = 0; j < 55; j++)

sum[j] = 0;

length1 = strlen(num1);

length2 = strlen(num2);

reverse(num1, length1-1);

reverse(num2, length2-1);

//printf("%s %s", num1, num2);

for (i = 0; i < length1 && i < length2; i++){

if(num1[i] == '.'){

sum[i] = '.';

}

else{

bit = num1[i] + num2[i] - 2 \* '0' + acc;

sum[i] = bit % 10 + '0';

if(bit<10)

acc = 0;

else

acc = 1;

}

}

if(i < length1)

for (; i < length1; i++){

bit = num1[i] - '0' + acc;

sum[i] = bit % 10 + '0';

if(bit<10)

acc = 0;

else

acc = 1;

}

if(i < length2)

for (; i < length2; i++){

bit = num2[i] - '0' + acc;

sum[i] = bit % 10 + '0';

if(bit<10)

acc = 0;

else

acc = 1;

}

if(acc == 1)

sum[i++] = '1';

sum[i] = '\0';

length3 = strlen(sum);

reverse(sum, length3 - 1);

printf("%s\n", sum);

}

return 0;

}

void reverse(char \*string, int length){

for (int i = 0; i <= length; i++, length--){

char tempc = string[i];

string[i] = string[length];

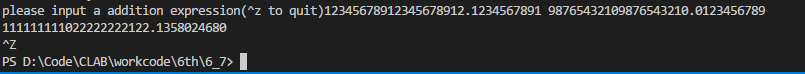
string[length] = tempc;

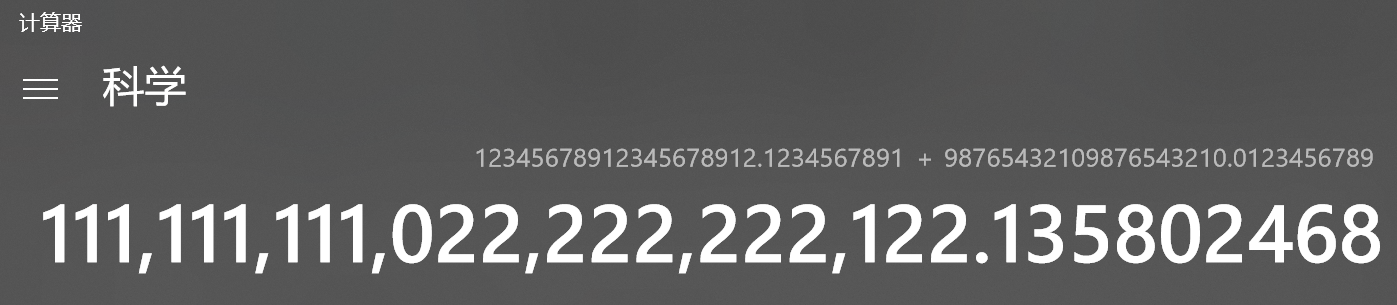
}

}

3）运行结果：

经Windows自带计算器验算结果正确





**图6-12 选做题1的测试运行结果**

（2）编写使用复杂声明char \*(\*p[2])(const char \*,const char \*);的程序。

提示：p中元素可为strcmp、strstr等函数名。

1. 程序思路：

利用复杂声明，定义一个函数指针数组，分别指向strcpy与strstr两个字符处理函数并通过函数指针调用函数。

1. 程序清单

#include <stdio.h>

#include <string.h>

int main(int argc, char \*argv[]) {

char a[6] = “1love2”;

char b[5] = “love”;

char \*(\*p[2])(const char \*, const char \*) = {strcpy, strstr};

printf("%s %s", p[1](a, b) ,p[0](a, b));

return 0;

}

1. 运行结果：



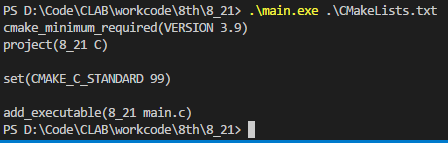
**图6-13 选做题2的测试运行结果**

**(三) 指定main函数的参数**

1) 此题未使用原题中介绍方法，在命令行中直接传入参数

2）程序可显示以参数为文件名的文本文件的内容

3）结果：



**图6-14 选做题3的测试运行结果**

## 6.3 实验小结

第六次C语言上机实验，主要学习并熟练了指针的说明、赋值与使用。掌握了用指针引用数组的元素与指向数组的指针、字符数组字符串、指针数组及字符指针数组、指针函数与函数指针以及带有参数的main函数的用法。

1. **改错实验**

改错实验还原了我们这种初学者在使用指针时最常见的错误，故意设置了一个悬挂指针并进行了引用和解引用，访问到了可能非法的内存地址，在将其修改之后，我了解到了悬挂指针的问题关键所在。如果随意地使用指针，轻则程序崩溃，重则可能引发一下潜在的问题，导致出现极难发现的BUG。

1. **程序修改替换实验**

此题考察了函数指针的运用，运用gets函数进行输入。此时我对原题中设置的变量i的存在产生了疑问，经思考后得知题目本意可能是利用循环以及scanf函数配合，避免gets函数带来的内存安全问题。同时我对指针函数以及函数指针的概念有了正确的认知，一个本质上是返回指针的函数，另一个是指向一个函数的指针变量，二者根本不同，在实际使用之后，明白了这一概念。

1. **跟踪调试题**

此题解题过程可谓一波三折。起初编译运行时，此题并无错误，虽有warning，但是可以正常输出，便并没有处理此题。在助教检查作业时，我突然发现，此题在单步运行时，并不能跳转到文件中定义的strcpy函数。这时我怀疑，编译器的自动优化功能忽视了与标准库函数同名的函数，调用了标准库函数。在给题中函数重新命名后，验证了猜想。随后发现了在while循环中，对指针s与t进行的自增操作会使两个指针最终都指向字符串的末尾，因此无法正常输出。在适当扩大字符数组a的大小使其可以容纳字符串，并在定义的函数中新加入指针保存字符串首地址并返回他之后，此题得解。

1. **程序设计实验**

第一题，主要让我明白了为何指针也有类型之分。通过这题，我认识到对指针进行操作时，实际上考虑其指向的变量的数据类型是十分重要的。指针本身只存储内存地址，但是需正确处理所指向变量，就需要知道应该从所指向地址向后几位以获得完整的数据，指针与整数加减法，也依赖此进行，数组的实现，也需要得知数据类型以及其占据的内存大小。

第二题，熟悉了指针数组的使用以及处理，指针数组与普通数组实际上并无差别，熟悉了对指针数组的操作。

第三题，计算学生成绩的各项数据，此题利用二维数组，计算学生平均成绩，可用行来存储每个学生的各项成绩，列自然成为各科目的每人成绩，再在嵌套循环中选择下标处理即可。此题主要让我意识到了模块化的重要，编写的四个函数可以很方便的得到再利用。同时，对于这类略有实用性的程序，在助教提醒下，还应该做好UI的设计，以方便使用与测试。

1. **选做题**

此次选做题中，第一题利用了我之前曾编写过的超大整数相加的程序修改而来，加上了小数点判断后便可正常工作，利用字符数组逐位相加并进位。第二个选做题主要还是在熟悉函数指针数组的使用，但是原题中略有错误，strcmp函数并不能作为题目要求中指针指向的对象。Strcmp函数返回整型数值，以表示参数中两个字符串的排序关系。第三题对main函数的参数的使用，使用了命令行进行完成，没有使用IDE的功能。

总之，此次C语言上机实验，让我领会到了指针的各种用途，对指针不再惧怕，在之后的学习与工作中，应该有勇气面对指针的使用。