# 实验报告9

实验九名称：指针和数组名 练习

实验目的：1、理解数组名、地址和指针的概念，理解数组名为指针常量的概念；2、掌握数组名（指针常量）做函数参数、指针变量做函数参数处理数组和字符串的使用方法。3、理解动态内存的概念与使用方法。

填写日期： 2022年12月1日

本次实验成绩：

1. **实验任务：**

1、编程实现整型数组的选择排序，程序中数组的访问要求使用指针表示法。

2、请使用数组元素存取方式，设计一个函数实现字符数组中字符串常量的逆序交换存放功能，主函数调用该函数实现其功能。

3、请使用指针存取方式，实现任务2。

4、利用动态内存分配的方法，输入一个班的学生某门功课的成绩，人数由用户指定，输出全班平均分以及最高分和最低分。

1. **实验方法及知识点应用：**

1、数组名与地址和指针的概念。

2、数组地址的传递使用。

3、指针变量处理数组和字符串的使用方法。

1. **实验程序代码及结果记录：（以表述清楚为目的）**

任务1：

源代码：

#include<stdio.h>

int num[10000];

int main(){

int i;

int \*p=num;

int n=0;

int min=0;

printf("请输入n个数进行从小到大选择排序(ctrl+z结束)：\n");

while(scanf("%d",&num[n++])!=EOF);

n--;

for(i=0;i<n;i++)

{

min=i;

int j;

for(j=i+1;j<n;j++)

{

if(\*(p+j)<\*(p+min))

{

min=j;

}

}

if(i!=min)

{

int tmp=\*(p+i);

\*(p+i)=\*(p+min);

\*(p+min)=tmp;

}

}

printf("排序结果为：\n");

for(i=0;i<n;i++)

{

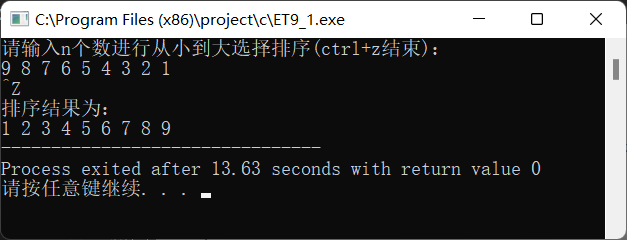
printf("%d ",\*(p+i));

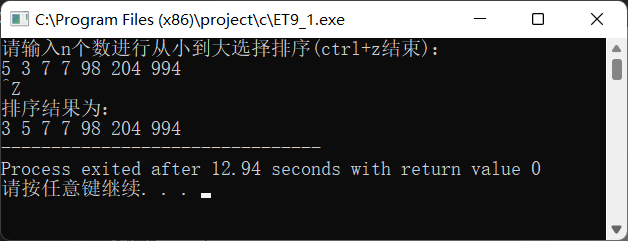
}

return 0;

}

测试：





任务2：

源代码：

#include<stdio.h>

#include<string.h>

void reverseSort(char [],int );

int main()

{

char str[1000];

printf("请输入字符串：\n");

gets(str);

int n=strlen(str);

reverseSort(str,n);

printf("逆序的结果为：\n");

puts(str);

return 0;

}

void reverseSort(char a[],int n)

{

int i;

for(i=0;i<n/2;i++)

{

int tmp=a[i];

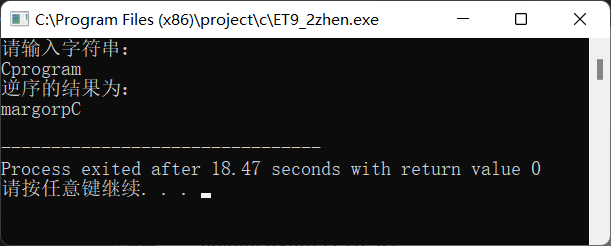
a[i]=a[n-i-1];

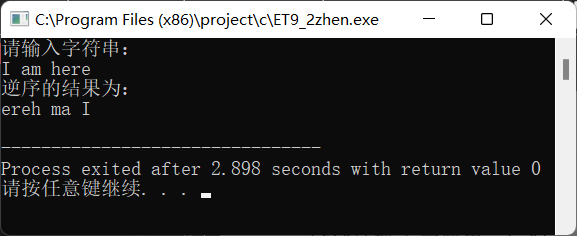
a[n-i-1]=tmp;

}

}

测试：





任务3：

源代码：

#include<stdio.h>

#include<string.h>

void reverseSort(char \*,int );

int main()

{

char str[1000];

printf("请输入字符串：\n");

gets(str);

int n=strlen(str);

reverseSort(str,n);

printf("逆序的结果为：\n");

puts(str);

return 0;

}

void reverseSort(char \*p,int n)

{

int i;

for(i=0;i<n/2;i++)

{

int tmp=\*(p+i);

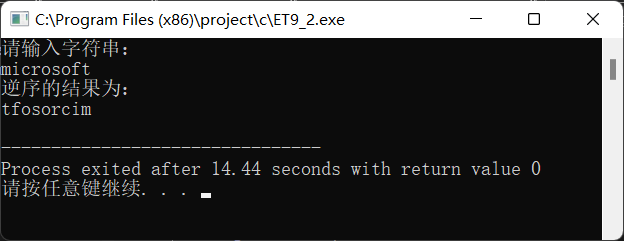
\*(p+i)=\*(p+n-i-1);

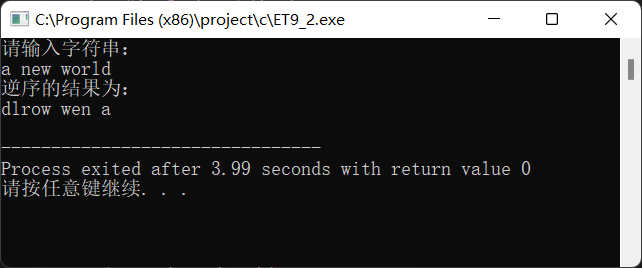
\*(p+n-i-1)=tmp;

}

}

测试：





任务4：

源代码：

#include<stdio.h>

#include<stdlib.h>

int main()

{

float min=0x3f3f3f3f;

float max=-0x3f3f3f3f;

printf("请输入人数：\n");

int n;

float sum=0;

scanf("%d",&n);

float \*p=(float\*)malloc(sizeof(float\*));

int i;

for(i=0;i<n;i++)

{

printf("请输入第%d个人的成绩:\n",i+1);

scanf("%f",p);

sum+=\*p;

if(\*p<min)

{

min=\*p;

}

if(\*p>max)

{

max=\*p;

}

}

printf("平均分为：%.2f\n",sum/n);

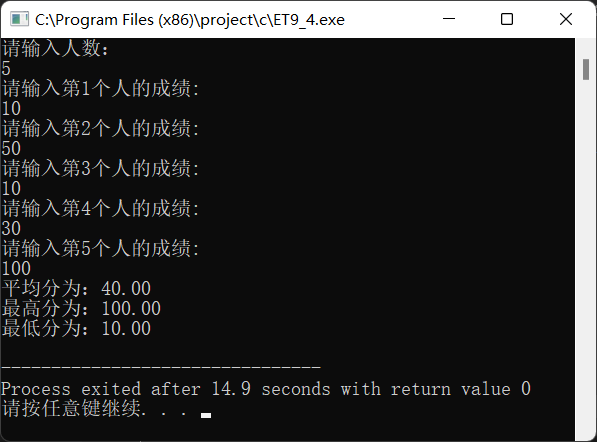
printf("最高分为：%.2f\n",max);

printf("最低分为：%.2f\n",min);

return 0;

}

测试：



1. **实验小结：**

先总结自己应用的知识点，再总结自己对知识点的理解和分析，根据实验结果和记录，最后总结实验中常出现的错误和问题。

应用知识点：

使用指针操作数组，选择排序算法，gets、puts函数的使用，动态分配内存malloc的使用，stdlib.h头文件的使用，strlen计算字符串长度函数的使用。

理解：

1. 数组名本质上便是一个指向数组首地址的指针，所以新设指针变量指向数组名时不用加&取址符。
2. 因为指针一开始指向数组的首地址，所以需要使用‘+’来对指针的定位进行改变，如\*(p+1)便是使指针向后移动指向第二个元素，并解引用得到该内存中的值。
3. 因为strlen得到的是字符串的长度，而字符串是从字符数组的下标0开始存放，所以在进行排序或者交换值的时候要-1，否则会越界。
4. 指针如果不是在定义的时候进行赋值的话，之后不会指向任何内存地址，所以不能在之后直接为其赋值，所以需要动态内存分配使其指向一块内存地址，这样才能够赋值给指针。

注意：

1. 当输入带空格的字符串时要使用gets函数，使用scanf函数会使输入在空格时截止。
2. malloc函数包含在stdlib.h头文件中，使用时注意要强制类型转换成相应指针类型。
3. 在指针操作数组时向后移位不要忘记使用括号，应为\*(p+1)，不能写成\*p+1，否则得到的是当前数组下标中值加1。