实验九

基础练习

1.

#include<stdio.h>

void maxmin(int\* date, int\* max, int\* min, int len){//查找并返回数组的最小值

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

//判断是否大于最大值或小于最小值

if(\*max < \*date){

\*max = \*date;

}else if(\*min > \*date){

\*min = \*date;

}

//数组指向后移一个单位

date++;

}

}

int main(){

int date[10] = {1,2,3,4,5,6,7,8,9,10};

int max = date[0];

int min = date[0];

maxmin(date, &max, &min, 10);

printf("%d, %d", max, min);

}

2.

#include<stdio.h>

#include <ctype.h>

#include<string.h>

int count(char\* string) {

int count = 0;

int str = strlen(string);

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

if (!isdigit(\*string)) {

count++;

}

string++;

}

return count;

}

int main() {

char str[] = "heel123abcd";

int sum = count(str);

printf("%d", sum);

}

3．

#include<stdio.h>

int del(int\* data, int n, int len) {

int\* index = data;

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

if (\*data == n) {

index++;

i++;

len--;

}

\*data = \*index;

data++;

index++;

}

return len;

}

int main() {

int arr[1024];

int i = 0;

int len = 0;

while (scanf("%d", &arr[i]) != -1)

{

i++;

len++;

}

len = del(arr, 4, len);

printf("%d", len);

}

4.

#include<stdio.h>

int search(int\* data, int n, int len) {

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

if (\*data == n) {

return i;

}

}

return -1;

}

int main() {

int data[1024];

int len = 0;

int i = 0;

while (scanf("%d", &data[i]) != -1) {

i++;

len++;

}

int n;

scanf("%d", &n);

int index = search(data, n, len);

if (len != -1) {

printf("%d\n", index);

}

else {

printf("没有找到\n");

}

}

5.

#include<stdio.h>

void delch(char\* string, int\* len, char n) {

char\* index = string;

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

if (\*string == n) {

index++;

i++;

}

\*string = \*index;

string++;

index++;

}

--\* len;

}

int main() {

char string[] = "abcdrf";

char n;

int len = sizeof(string) / sizeof(string[0]) - 1;

scanf(" %c", &n);

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

delch(string, &len, n);

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

printf("%c", string[i]);

}

printf("\n");

}

6.

#include<stdio.h>

void wpos(char\* string, int\* begin, int\* end) {

int len = 0;

while(1){

int\* start = string + \*begin;

char\* temp = strchr(start, ' ');

if (temp == NULL) {

break;

}

\*end = temp - string;

if (\*end - \*begin > len && \*end >= 0) {

len = \*end - \*begin;

}

else if (\*end < 0) {

break;

}

\*begin = \*end + 1;

}

printf("%d", len);

}

int main() {

char ch[1024] = "The code is likely to give a read access violation error because";

int begin = 0;

int end = 0;

wpos(ch, &begin, &end);

return 0;

}

7.

#include<stdio.h>

void strsta(char\* string, int\* num) {

int len = strlen(string);

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

if (\*string >= '0' && \*string <= '9') {

(\*(num + 1))++;

}

else if (isalpha(\*string)) {

(\*num)++;

}

else if (\*string == ' ') {

(\*(num + 2))++;

}

else {

(\*(num + 3))++;

}

string++;

}

}

int main(void) {

char string[1024] = "abcd12345 #$%";

int num[4] = {0,0,0,0};

strsta(string, num);

printf("%d %d %d %d", num[0], num[1], num[2], num[3]);

}

8.

#include<stdio.h>

void svowel(char\* s1, char\* s2) {

int len = strlen(s1);

int count = 0;

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

if (\*s1 == 'a' || \*s1 == 'e' || \*s1 == 'i' || \*s1 == 'o' || \*s1 == 'u'

|| \*s1 == 'A' || \*s1 == 'E' || \*s1 == 'I' || \*s1 == 'O' || \*s1 == 'U' ) {

\*s2 = \*s1;

s2++;

count++;

}

s1++;

}

}

int main(void) {

char s1[1024] = "abcdefghhhhhh";

char s2[1024] = "0";

svowel(s1, s2);

for (int i = 0; i < strlen(s1); i++) {

printf("%c", s1[i]);

}

printf("\n");

for (int i = 0; i < strlen(s2); i++) {

printf("%c", s2[i]);

}

}

9.

#include<stdio.h>

#define STUFENT 10

int searcho(char(\*stuno)[11], char\* no, int n) {

//储存数组的长度

int index = -1;

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

//no储存的值要循环变化，所以创建一个临时变量来储存

char\* temp = no;

for (int j = 0; j < n; j++) {

//判断数组中的每一个元素是否相等

if ((\*stuno)[j] != \*temp) {//11703990101

//不相等就直接退出开始下一个数组的判断

break;

}

else {

//相等的话就继续判断下一个元素

temp++;

}

}

//如果temp储存的地址和no储存的地址+11相等，说明每一个元素都相等

if (temp == no + 11) {//调试看一下具体情况

index = i;

break;

}

else {

stuno++;

}

}

return index;

}

int main() {

char student[10][11] = { "11703990101","11703990102", "11703990103", "11703990104", "11703990105",

"11703990106", "11703990107", "11703990108", "11703990109", "11703990110" };

char target[11] = { "11703990103" };

int index = searcho(student, target, 11);

if (index != -1) {

printf("%d", index);

}

else {

printf("没有找到");

}

return 0;

}

10.

#include<stdio.h>

void stringvert(char\* str) {

int len = strlen(str);

char temp = 0;

for (int i = 0, j = len - 1; i < j; i++, j--) {

temp = \*(str + i);

\*(str + i) = \*(str + j);

\*(str + j) = temp;

}

}

int main(void) {

char str[1024] = "abcdef";

stringvert(str);

for (int i = 0; i < strlen(str); i++) {

printf("%c", str[i]);

}

}