1 使得分开后的字符串前一部分是数字后一部分是字母

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define N 8

void main() {

char c[N] = "h1ell2o3";

char a[N] = { 0 };

char b[N] = { 0 };

int j = 0, k = 0;

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

if (c[i] >= 'a') {

a[j++] = c[i];

}

else{

b[k++] = c[i];

}

}

printf("%s%s",b,a);

}



2 将 字 符 串 中 的 空 格 替 换 成 “%020”

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

//指针变量是存地址，指针变量，工作时，地址就是指针

//指针变量本质为了实现间接访问

#define N 30

void main() {

char\* p = "hello world how ";

int len = strlen(p);

while (len--) {

\*p++ >= 'a' ? printf("%c", \*(p-1)) : printf("%s","%020");

}

}



3 删除字符串中指定的字符

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define N 30

void main() {

char\* p = "abcdaefaghiagkl";

int len = strlen(p);

char res[N] = { 0 };

int k = 0;

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

if (\*p != 'a') {

res[k++] = \*p;

}

p++;

}

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

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

}

}



4 删除一个数组中重复的元素

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define N 30

void main() {

int p[15] = {1,2,2,2,3,3,3,4,4,5,5,5,6,6,6 };

int a[15] = { 0 };

for (int i = 0; i < 15; i++) {

a[p[i]] ++;

}

for (int j = 0; j < 15; j++) {

if (a[j] > 0) {

p[j] = j;

printf("%d", j);

}

}

}



5 将 字 符 串 中 的 相 邻 的 多 余 空 格 去 掉

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

//指针变量是存地址，指针变量，工作时，地址就是指针

//指针变量本质为了实现间接访问

#define N 30

void main() {

char str[] = "\_\_\_hello\_\_\_\_world\_\_\_how\_are\_you\_\_";

char delims[] = "\_";

char\* result = NULL;

result = strtok(str, delims);

while (result != NULL) {

printf("%s", result);

result = strtok(NULL, delims);

if (result != '\0') printf("\_");

}

}



6 大数相加

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

//指针变量是存地址，指针变量，工作时，地址就是指针

//指针变量本质为了实现间接访问

#define N 50

void main() {

char longs[N+1] = { 0 };

char shorts[N+1] = { 0 };

scanf("%s", &shorts);

scanf("%s", &longs);

int lenA = strlen(shorts);

int lenB = strlen(longs);

if (lenA == lenB) {

for (int i = N - 1; i >= 0; i--) {

char temp = 1;

char add = longs[i] - '0' + shorts[i] - '0';

if (add <= '9' - '0') {

longs[i] = add;

}

else {

longs[i] = add - 10;

if (i - 1 == -1) longs[N] = 1;

else longs[i - 1] ++;

}

}

if(longs[N] == 1)

printf("%c", longs[N] + '0');

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

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

}

}

else {

char temp[N + 1];

if (lenA > lenB) {

strcpy(temp, longs);

strcpy(longs, shorts);

strcpy(shorts, temp);

}

int longlen = strlen(longs);

int shortlen = strlen(shorts);

char res[N + 1] = { 0 };

int jinwei = 0;

int yizhijinwei = 0;

int lencha = longlen - shortlen;

for (int i = longlen - 1; i >= lencha; i--) {

char add = longs[i] - '0' + shorts[i - lencha] - '0';

if (add <= '9' - '0') {

res[i] = add;

if (i == lencha) {

res[i - 1] = 1 - '0';

}

}

else {

res[i] = add - 10;

res[i - 1] = 0;

yizhijinwei = 1;

//if (res[i - 1] == 10) {

// res[i - 1] = 0 - '0';

// res[i - 2] += zaijinyiwei - '0';

//}

}

}

for (int j = 0; j < lencha - 1; j++) {

res[j] += longs[j] - '0';

if (yizhijinwei) {

res[j]++;

}

}

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

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

if (res[i] == '0') break;

}

}

}



