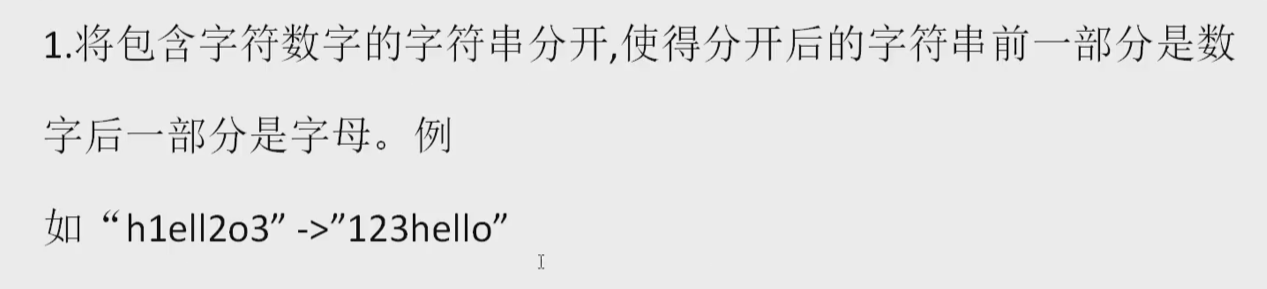
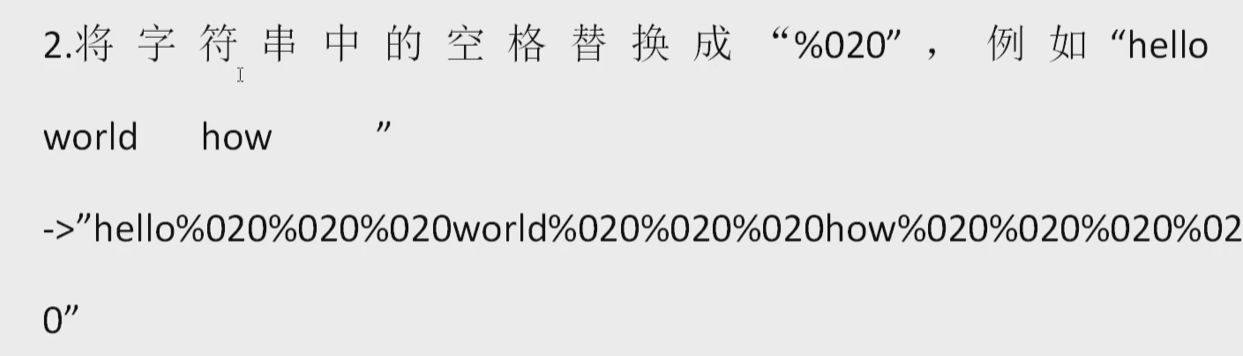
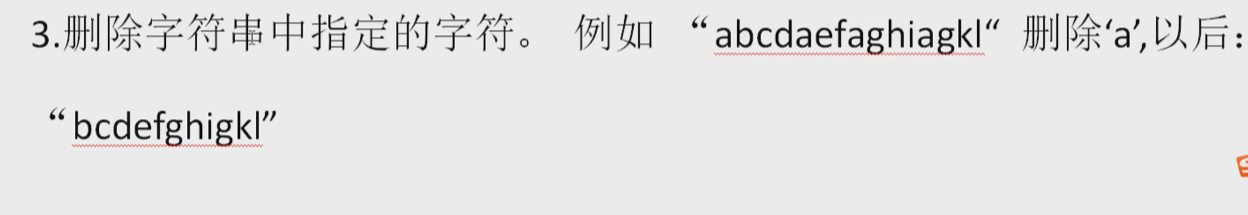
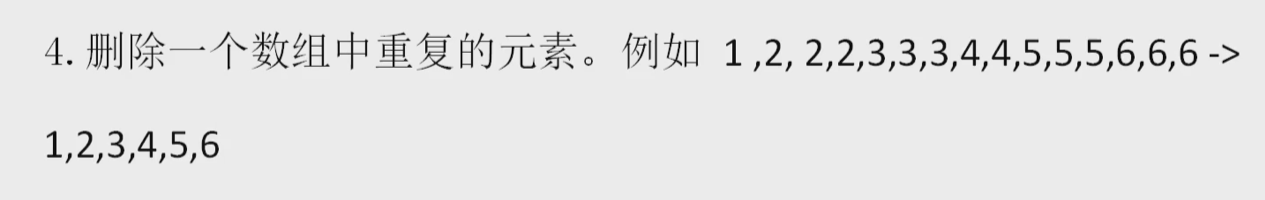
在限定的空间下，

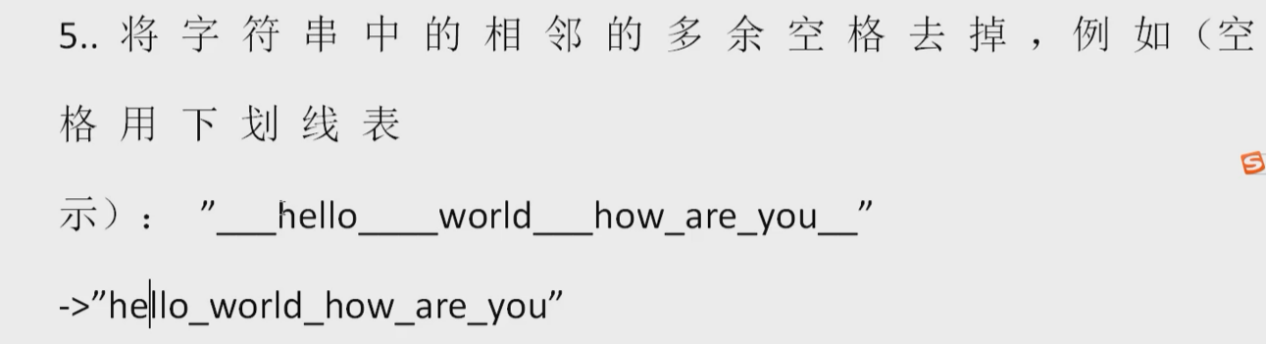


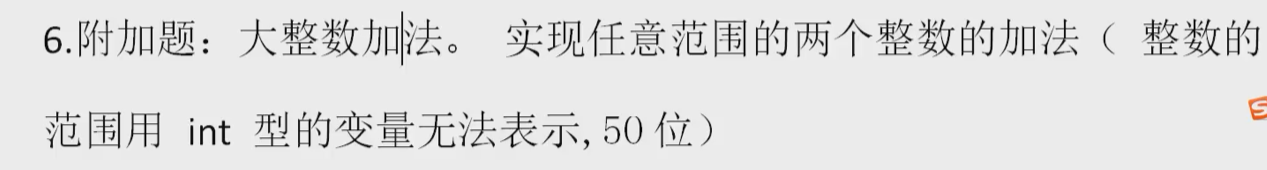






4题有序，若无序一般用位图





6简易直接用char c[51]用字符数组来加减。

1.

void world\_sort(char\* str) {

int num\_count = 0;

char\* p = str;

char\* p\_num = NULL;

char\* p\_world = NULL;

if (\*str < '0' || \*str >'9') {

p\_world = str;

}

else {

p\_num = str;

num\_count++;

}

while (\*str) {

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

p\_num = str;

num\_count++;

}

else if((\*(str - 1) > '0' && \*(str-1) <'9') && (\* str <'0' || \*str >'9') ) {//前一个字符是数字，当前字符是字母，即，找到第一个字母

p\_world = str;

}

if (p\_num != NULL && p\_world != NULL) {

while (p\_world < p\_num) {

char temp = \*p\_num;

\*p\_num = \*(p\_num - 1);

\*(p\_num - 1) = temp;

p\_num--;

}

p\_num = NULL;

//p\_world = str;

p\_world = &p[num\_count];

}

str++;

}

}

2.

//计算需要的足够空间，从后往前改变

void replace\_space(char\* str) {

char\* p = str;

char\* p\_start = str;

char\* p\_end = str;

int space\_num = 0;

int size = 0;

while (\*p\_end) {

if (\*p\_end == ' ') {

space\_num++;

}

p\_end++;

}

size = strlen(str) + space\_num \* 2;//一个字符变成3个字符

p = str + size ;//p指向预留空间后，最后的字符处,不要-1，要为\0留出位置

while (p\_start < p\_end) {

if (\*p\_end == ' ') {

\*p = '0';

\*(p - 1) = '2';

\*(p - 2) = '%';

p = p - 3;

}

else {

\*p = \*p\_end;

p--;

}

p\_end--;

}

puts(str);

}

int main() {

char str[100];

//scanf("%s", str);//scanf %s接受不了空格

char\* find;

fgets(str, 100, stdin);

if (find = strchr(str, '\n')) {//特殊处理一下fgets吞下的回车，有点烦人

\*find = '\0';

}

replace\_space(str);

system("pause");

return 0;

}

3.

void delete\_char(char\* str,char c) {

while (\*str) {

if (\*str == c) {

char\* p = str;

while (\*p) {

\*p = \*(p + 1);

p++;

}

\*p = '\0';

}

str++;

}

}

int main() {

char str[100];

//scanf("%s", str);//scanf %s接受不了空格

char\* find;

fgets(str, 100, stdin);

if (find = strchr(str, '\n')) {//特殊处理一下fgets吞下的回车，有点烦人

\*find = '\0';

}

delete\_char(str,'a');

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

system("pause");

return 0;

}