## task1:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAXSIZE 500

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

{

FILE \*fp=NULL,\*fpread;

char \*buf=NULL;

size\_t len=0;

int num;

char szbuf[MAXSIZE+2]={0},readbuf[MAXSIZE+2]={0};

int i,maxsize\_len=0;

if(argc !=3)

{

printf("pls input filename and size\n");

return 0;

}

fp=fopen(argv[1],"w+");

if(fp==NULL)

{

printf("open error\n");

return -1;

}

maxsize\_len=atoi(argv[2]);

for(i=0;i<maxsize\_len;i++)

{

szbuf[i]='a';

}

printf("szbuf=%s\n",szbuf);

if( fwrite(szbuf,1,strlen(szbuf),fp) <0)

{

printf("write error\n");

return -1;

}

fclose(fp);

fpread=fopen(argv[1],"r");

if(fpread==NULL)

{

printf("fopen error\n");

return -1;

}

fread(readbuf,1,sizeof(readbuf),fpread);

printf("read buf is %s\n",readbuf);

fseek(fpread,0L,SEEK\_SET);

while((num=getline(&buf,&len,fpread))!=-1)

{

printf("num=%d,len=%d,content=%s\n",num,(int)len,buf);

}

//fclose(fp);

free(buf);

fclose(fpread);

return 0;

}

## task2:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main()

{

char buf[100][100];

int i;

int num=0;

printf("please input num\n");

scanf("%d",&num);

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

{

scanf("%s",(char \*)&buf[i]);

}

printf("you input :\n");

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

printf("%s\n",buf[i]);

return 0;

}

## task3:

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#define SIZE 100

int main(){

int strNum = 0;

char \*\*strArray = NULL;

int i = 0;

char buf[SIZE];

printf("please input the num of string \n");

scanf("%d",&strNum);

strArray = (char \*\*)malloc(strNum \* sizeof(char \*));

if(NULL == strArray){

perror("allocate memory failed");

exit(2);

}

for(i = 0; i < strNum; i++){

strArray[i] = (char \*)malloc(SIZE \* sizeof(char));

if(NULL == strArray[i]){

perror("allocate memory failed");

exit(2);

}

memset(buf,'\0',sizeof(buf));

printf("input the %d str:\n",i+1);

scanf("%s",buf);

strcpy(strArray[i],buf);

}

printf("now, printf all strings\n");

for(i = 0; i < strNum; i++){

memset(buf,'\0',sizeof(buf));

strcpy(buf,strArray[i]);

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

free(strArray[i]);

}

free(strArray);

return 0;

}

## task4:

#include <stdio.h>

#include <stdlib.h>

int cmpfunc (const void \* a, const void \* b)

{

return ( \*(int\*)a - \*(int\*)b );

}

int main()

{

int \*values = NULL;

int i;

int total = 0;

printf("printf the num\n");

scanf("%d",&total);

values = (int \*)malloc(total \* sizeof(int));

printf("Before sorting the list is: \n");

for( i = 0 ; i < total; i++ ) {

scanf("%d",&values[i]);

}

qsort(values, total, sizeof(int), cmpfunc);

printf("After sorting the list is: \n");

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

{

printf("%d\n", values[i]);

}

free(values);

return(0);

}

## task5:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define LINENUM 4096

int cmpstring(const void \* a, const void \* b)

{

char \* str1 = \*(char \*\*)a;

char \* str2 = \*(char \*\*)b;

return strcmp(str1, str2);

}

int main(int argc, char \*\*argv)

{

char \*buf = NULL;

size\_t len = 0;

int num;

if(argc != 2){

printf("the arguments is wrong! \n usage: readfile filename\n");

return -1;

}

char \* filename = argv[1];

FILE \*test = fopen(filename, "r");

if ( test == NULL){

printf("the file %s doesn't exist!\n", filename);

}

//allocate the memory to store the content in the file

char \*\*lines = (char \*\*)malloc( LINENUM \* sizeof(char\*));

int lineno = 0;

while ( (num = getline ( &buf, &len, test )) != -1){

//copy the content in buf to lines

lines[lineno]= (char\*)malloc(len \* sizeof(char));

strcpy(lines[lineno], buf);

lineno ++;

}

//sort the string lines according to the alphabet sort

qsort(lines, lineno, sizeof(char \*), cmpstring);

int i = 0;

for(i=0; i < lineno; i++){

printf("%s",lines[i]);

free(lines[i]);

}

free(lines);

}

## task6:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define LINENUM 4096

int cmpstring(const void \* a, const void \* b)

{

char \* str1 = \*(char \*\*)a;

char \* str2 = \*(char \*\*)b;

return strcmp(str1, str2);

}

int main(int argc, char \*\*argv)

{

char \*buf = NULL;

size\_t len = 0;

int num;

if(argc != 2){

printf("the arguments is wrong! \n usage: readfile filename\n");

return -1;

}

char \* filename = argv[1];

FILE \*test = fopen(filename, "r");

if ( test == NULL){

printf("the file %s doesn't exist!\n", filename);

}

//allocate the memory to store the content in the file

char \*\*lines = (char \*\*)malloc( LINENUM \* sizeof(char\*));

int lineno = 0;

while ( (num = getline ( &buf, &len, test )) != -1){

//copy the content in buf to lines

lines[lineno]= (char\*)malloc(len \* sizeof(char));

strcpy(lines[lineno], buf);

lineno ++;

}

//sort the string lines according to the alphabet sort

qsort(lines, lineno, sizeof(char \*), cmpstring);

int i = 0;

for(i=0; i < lineno; i++){

printf("%s",lines[i]);

}

// delete the same line

// create new file, and write the result to new file

char \* relfilename = strcat(filename, "-new");

FILE \*rel = fopen(relfilename, "w ");

fwrite(lines[0], 1,strlen(lines[0]),rel);

for(i = 1; i < lineno; i++){

if(strcmp(lines[i], lines[i-1]) == 0)

continue;

else {

fwrite(lines[i], 1, strlen(lines[i]),rel);

}

}

for(i=0; i < lineno; i++){

free(lines[i]);

}

free(lines);

fclose(test);

fclose(rel);

free(buf);

}

## task7://并集

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define LINENUM 4096

int main(int argc, char \*\* argv)

{

if ( argc != 3){

printf("usage: jbcset file1 file2\n");

return -1;

}

char \* file1name = argv[1];

char \* file2name = argv[2];

FILE \* file1 = fopen(file1name, "r");

FILE \* file2 = fopen(file2name, "r");

FILE \* bfile = fopen("file1\_b\_file2", "w ");

char \*buf1 = NULL;

char \*buf2 = NULL;

size\_t len1 = 0, len2 = 0;

int num1, num2;

int cmprel;

num1 = getline( &buf1, &len1, file1);

num2 = getline( &buf2, &len2, file2);

while( num1 != -1 && num2 != -1){

cmprel = strcmp(buf1, buf2);

if( cmprel < 0 ){

fwrite(buf1, 1, strlen(buf1), bfile);

num1 = getline( &buf1, &len1, file1);

}

else if( cmprel > 0 ){

fwrite(buf2, 1, strlen(buf2), bfile);

num2 = getline( &buf2, &len2, file2);

}

else {

fwrite(buf1, 1, strlen(buf1), bfile);

num1 = getline( &buf1, &len1, file1);

num2 = getline( &buf2, &len2, file2);

}

}

while( num1 != -1){

fwrite(buf1, 1, strlen(buf1), bfile);

num1 = getline( &buf1, &len1, file1);

}

while( num2 != -1){

fwrite(buf2, 1, strlen(buf2), bfile);

num2 = getline( &buf2, &len2, file2);

}

free(buf1);

free(buf2);

fclose(file1);

fclose(file2);

fclose(bfile);

}

## Task8://交集

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define LINENUM 4096

int main(int argc, char \*\* argv)

{

if ( argc != 3){

printf("usage: jbcset file1 file2\n");

return -1;

}

char \* file1name = argv[1];

char \* file2name = argv[2];

FILE \* file1 = fopen(file1name, "r");

FILE \* file2 = fopen(file2name, "r");

FILE \* jfile = fopen("file1\_j\_file2", "w ");

char \*buf1 = NULL, \*buf2 = NULL;

size\_t len1 = 0, len2 = 0;

int num1, num2;

int cmprel;

num1 = getline( &buf1, &len1, file1);

num2 = getline( &buf2, &len2, file2);

while( num1 != -1 && num2 != -1){

cmprel = strcmp(buf1, buf2);

if( cmprel < 0 ){

num1 = getline( &buf1, &len1, file1);

}

else if( cmprel > 0 ){

num2 = getline( &buf2, &len2, file2);

}

else {

fwrite(buf1, 1, strlen(buf1), jfile);

num1 = getline( &buf1, &len1, file1);

num2 = getline( &buf2, &len2, file2);

}

}

free(buf1);

free(buf2);

fclose(file1);

fclose(file2);

fclose(jfile);

}

## Task9 差集

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define LINENUM 4096

int main(int argc, char \*\* argv)

{

if ( argc != 3){

printf("usage: jbcset file1 file2\n");

return -1;

}

char \* file1name = argv[1];

char \* file2name = argv[2];

FILE \* file1 = fopen(file1name, "r");

FILE \* file2 = fopen(file2name, "r");

FILE \* cfile= fopen("file1\_c\_file2", "w ");

char \*buf1 = NULL, \*buf2 = NULL;

size\_t len1 = 0, len2 = 0;

int num1, num2;

int cmprel;

num1 = getline( &buf1, &len1, file1);

num2 = getline( &buf2, &len2, file2);

while( num1 != -1 && num2 != -1){

cmprel = strcmp(buf1, buf2);

if( cmprel < 0 ){

fwrite(buf1, 1, strlen(buf1), cfile);

num1 = getline( &buf1, &len1, file1);

}

else if( cmprel > 0 ){

num2 = getline( &buf2, &len2, file2);

}

else {

num1 = getline( &buf1, &len1, file1);

num2 = getline( &buf2, &len2, file2);

}

}

while(num1 != -1){

fwrite(buf1, 1, strlen(buf1), cfile);

num1 = getline( &buf1, &len1, file1);

}

free(buf1);

free(buf2);

fclose(file1);

fclose(file2);

fclose(cfile);

}

## //task10 拷贝

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_SIZE 1024

void copy(FILE \*form\_file,FILE \*to\_file);

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

{

FILE \*from\_file;

FILE \*to\_file;

if(argc!=3){

printf("Please input a right expression!\n");

exit(1);

}else{

if((from\_file=fopen(argv[1],"rb"))==NULL)

{

printf("Can't open the file of %s\n",argv[1]);

exit(1);

}

if((to\_file=fopen(argv[2],"wb"))==NULL)

{

printf("Can't open the file of %s!\n",argv[2]);

exit(1);

}

copy(from\_file,to\_file);

}

return 0;

}

void copy(FILE \*from\_file,FILE \*to\_file)

{

char buff[MAX\_SIZE];

long file\_size;

fseek(from\_file,0L,SEEK\_END);

file\_size = ftell(from\_file);

fseek(from\_file,0L,SEEK\_SET);

// printf("The file size is:%d\n",file\_size);

while(!feof(from\_file))

{

memset(buff,0,MAX\_SIZE);

fread(buff,MAX\_SIZE,1,from\_file);

if(file\_size<MAX\_SIZE){

fwrite(buff,file\_size,1,to\_file);

}else{

fwrite(buff,MAX\_SIZE,1,to\_file);

file\_size -= MAX\_SIZE;

}

}

fclose(from\_file);

fclose(to\_file);

}

## //task11

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main()

{

FILE \*fp=NULL,\*fpwrite=NULL;

char \*buf=NULL;

size\_t len=0;

int num;

char \*pstr;

char strname[50]={0};

int sublen=0;

fp=fopen("/etc/passwd","r");

fpwrite=fopen("user\_name","w");

if(fp==NULL || fpwrite==NULL)

{

printf("fopen error\n");

return -1;

}

fseek(fp,0L,SEEK\_SET);

while((num= getline(&buf,&len,fp)) !=-1)

{

//printf("%s\n",buf);

pstr=strchr(buf,':');

sublen=pstr-buf;

//printf("sublen=%d\n",sublen);

memset(strname,0,50);

strncpy(strname,buf,sublen);

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

strcat(strname,"\n");

fwrite(strname,1,strlen(strname),fpwrite);

}

free(buf);

fclose(fp);

fclose(fpwrite);

return 0;

}

## //附加

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main()

{

FILE \*fp=NULL,\*fpwrite=NULL;

char \*buf=NULL;

size\_t len=0;

int num;

char \*pstr;

char strname[50]={0};

int sublen=0;

fp=fopen("/etc/passwd","r");

fpwrite=fopen("user\_name","w");

if(fp==NULL || fpwrite==NULL)

{

printf("fopen error\n");

return -1;

}

fseek(fp,0L,SEEK\_SET);

while((num= getline(&buf,&len,fp)) !=-1)

{

//printf("%s\n",buf);

pstr=strchr(buf,':');

sublen=pstr-buf;

//printf("sublen=%d\n",sublen);

memset(strname,0,50);

strncpy(strname,buf,sublen);

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

strcat(strname,"\n");

fwrite(strname,1,strlen(strname),fpwrite);

}

free(buf);

fclose(fp);

fclose(fpwrite);

return 0;

}

//added

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAXLEN 100

void lcsLength(char \*x, char \*y, int m, int n, int c[][MAXLEN], int b[][MAXLEN]){

int i,j;

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

c[i][0] = 0;

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

c[0][j] = 0;

for(i = 1; i <= m; i++){

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

if(x[i-1] == y[j-1]){

c[i][j] = c[i-1][j-1]+1;

b[i][j] = 0;

}

else if(c[i][j-1] > c[i-1][j]){

c[i][j] = c[i][j-1];

b[i][j]= -1;

}

else{

c[i][j] = c[i-1][j];

b[i][j]= 1;

}

}

}

}

void printLCS(int b[][MAXLEN], char \*x, int i, int j){

if(i == 0 || j ==0)

return;

if(b[i][j] == 0){

printLCS(b, x, i-1,j-1);

printf("%c", x[i-1]);

}

else if(b[i][j] == 1){

printLCS(b, x, i-1, j);

}

else{

printLCS(b, x, i, j-1);

}

}

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

if (argc != 3){

printf("usage: lcs string1 string2\n");

return -1;

}

int b[MAXLEN][MAXLEN];

int c[MAXLEN][MAXLEN];

int m,n;

m = strlen(argv[1]);

n = strlen(argv[2]);

lcsLength(argv[1], argv[2], m, n,c, b);

printLCS(b,argv[1],m,n);

printf("\n");

return 0;

}