444.#include<stdio.h>

#include<errno.h>

int main()

{

FILE \* pf = fopen("text.txt","w");

if (pf == NULL)

{

printf("%s\n",strerror(errno));

return 0;

}

fclose(pf);

pf = NULL;

return 0;

}

445.#include<stdio.h>

#include<errno.h>

#include<string.h>

int main()

{

FILE \* pf = fopen("text.txt","w");

if (pf == NULL)

{

printf("%s\n",strerror(errno));

return 0;

}

fputc('b',pf);

fputc('i',pf);

fputc('t',pf);

fclose(pf);

pf = NULL;

return 0;

}

446.#include<stdio.h>

#include<errno.h>

#include<string.h>

int main()

{

FILE \* pf = fopen("text.txt","w");

if (pf == NULL)

{

printf("%s\n",strerror(errno));

return 0;

}

int ch = fgetc(pf);

printf("%c",ch);

ch = fgetc(pf);

printf("%c",ch);

ch = fgetc(pf);

printf("%c",ch);

fclose(pf);

pf = NULL;

return 0;

}

447.#include<stdio.h>

int main()

{

char buf[1024] = {0};

FILE\*pf = fopen("text.txt","r");

if (pf == NULL)

{

return 0;

}

fgetc(buf, 1024, pf);

printf("%s",buf);

//fgetc(buf, 1024, pf);

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

fclose(pf);

pf = NULL;

return 0;

}

448.#include<stdio.h>

int main()

{

char buf[1024] = {0};

FILE\*pf = fopen("text.txt","r");

if (pf == NULL)

{

return 0;

}

fputs("hello\n",pf);

fputs("world\n",pf);

fclose(pf);

pf = NULL;

return 0;

}

449.#include<stdio.h>

int main()

{

char buf[1024] = { 0 };

fgets(buf,1024,stdin);

fputs(buf, stdout);

gets(buf);

gets(buf);

return 0;

}

450.#include<stdio.h>

struct S

{

int n;

float score;

char arr[10];

};

int main()

{

struct S s = {100,3.14f,"bit"};

FILE \*pf = fopen("text.txt","w");

if (pf == NULL)

{

return 0;

}

/\*fscanf(pf,"%d %f %s",&(s.n),&(s.score),&(s.arr));

printf("%d %0.2f %s",s.n ,s.score ,s.arr);\*/

fprintf(pf,"%d %f %s",s.n,s.score,s.arr);

fclose(pf);

pf = NULL;

return 0;

}

451.#include<stdio.h>

struct S

{

int n;

float score;

char arr[10];

};

int main()

{

struct S s = {100,3.14f,"abcdef"};

struct S tmp = {0};

char buf[1024] = {0};

sprintf(buf,"%d %f %s",s.n ,s.score, s.arr);

sscanf(buf,"%d %f %s",&(tmp.n),&(tmp.score),&(tmp.arr));

printf("%d %f %s\n",tmp.n ,tmp.score, tmp.arr);

return 0;

}//100 3.140000 abcdef

452.#include<stdio.h>

struct S

{

int n;

float score;

char arr[10];

};

int main()

{

struct S s = { "张三",20,55.6 };

FILE\*pf = ("text.txt","wb");

if (pf == NULL)

{

return 0;

}

fwriter(&s,sizeof(struct S),1,pf);//fread

fclose(pf);

pf == NULL;

return 0;

}

1. //text.c

#include<stdio.h>

#include "contact.h"

void menu()

{

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf("\*\*\*\*\*\*1.add 2.del \*\*\*\*\*\*\*\*\*\*\*");

printf("\*\*\*\*\*\*3.search 4.modify \*\*\*\*\*\*\*\*\*\*\*");

printf("\*\*\*\*\*\*5.show 6.sort \*\*\*\*\*\*\*\*\*\*\*");

printf("\*\*\*\*\*\*7.save 0.exit \*\*\*\*\*\*\*\*\*\*\*");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

int main()

{

int input = 0;

struct Contact con;

InitContact(&con);

do {

menu();

printf("请选择:>");

scanf("%d ", &input);

switch (input)

{

case ADD:

AddContact(&con);

break;

case DEL:

DelContact(&con);

break;

case SEARCH:

SearchContact(&con);

break;

case MODIFY:

ModifyContact(&con);

break;

case SHOW:

ShowContact(&con);

break;

case SORT:

SortContact(&con);

break;

case EXIT:

SaveContact(&con);

DestroyContact(&con);

printf("退出通讯录");

break;

case SAVE:

SaveContact(&con);

default:

printf("输入错误，请重新选择");

break;

}

} while (input);

return 0;

}

//contact.h

#define MAX 1000

#define MAX\_NAME 20

#define MAX\_TELE 12

#define MAX\_SEX 5

#define MAX\_ADDR 30

#define DEFAULT\_SZ 3

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<errno.h>

enum Option

{

EXIT,

ADD,

DEL,

SEARCH,

MODIFY,

SHOW,

SORT,

SAVE

};

typedef struct PeoInfo

{

char name[MAX\_NAME];

int age;

char tele[MAX\_TELE];

char sex[MAX\_SEX];

char addr[MAX\_ADDR];

}PeoInfo;

typedef struct Contact

{

struct PeoInfo \*data;

int size;

int capacity;

}Contact;

void InitContact(struct Contact \* ps);

void AddContact(struct Contact \* ps);

void DelContact(struct Contact \* ps);

void SearchContact(struct Contact \* ps);

void ModifyContact(struct Contact \* ps);

void ShowContact(const struct Contact \* ps);

void SortContact(struct Contact \* ps);

void SaveContact(struct Contact \*ps);

void LoadContact(Contact \*ps);

//contact.c

#include "text.c"

void InitContact(struct Contact \* ps)

{

ps->data = (struct PeoInfo\*)malloc(DEFAULT\_SZ \* sizeof(struct PeoInfo));

if (ps->data == NULL)

{

return;

}

ps->size = 0;

ps->capacity = DEFAULT\_SZ;

LoadContack();

}

void CheckCapacity(struct Contact \*ps)

{

struct PeoInfo\*ptr = realloc(ps->data, (ps->capacity + 2)\*(sizeof(PeoInfo)));

if (ptr != NULL)

{

ps->data = ptr;

ps->capacity += 2;

printf("扩充成功\n");

}

else

{

printf("扩容失败\n");

}

}

void AddContact(struct Contact \* ps)

{

CheckCapacity(ps);

printf("请输入名字:>");

scanf("%s", ps->data[ps->size].name);

printf("请输入年龄:>");

scanf("%d", &ps->data[ps->size].age);

printf("请输入性别:>");

scanf("%s", ps->data[ps->size].sex);

printf("请输入电话:>");

scanf("%s", ps->data[ps->size].tele);

printf("请输入地址:>");

scanf("%s", ps->data[ps->size].addr);

ps->size++;

printf("添加成功\n");

}

void DelContact(struct Contact \* ps)

{

char name[MAX\_NAME];

printf("请输入要删除人的名字:>");

scanf("%s", name);

int pos = FindByName(ps, name);

if (pos == -1)

{

int j = 0;

for (j = pos; j < ps->size - 1; j++)

{

ps->data[j] = ps->data[j + 1];

}

ps->size--;

printf("删除成功\n");

}

}

void SearchContact(struct Contact \* ps)

{

char name[MAX\_NAME];

printf("请输入要查找人的名字:>");

scanf("%s", name);

int pos = FindByName(ps, name);

if (pos == -1)

{

printf("要查找的人不存在\n");

}

else

{

printf("%-20s\t%-4s\t%-5s\t%-12s\t%-20\n", "名字", "年龄", "性别", "电话", "地址");

printf("%%-20s\t%-4s\t%-5s\t%-12s\t%-20\n",

ps->data[pos].name,

ps->data[pos].age,

ps->data[pos].tele,

ps->data[pos].sex,

ps->data[pos].addr);

}

}

void ModifyContact(struct Contact \* ps)

{

int pos = 0;

char name[MAX\_NAME];

printf("请输入要修改人的名字:>");

scanf("%s", name);

pos = FindByName(ps, name);

if (pos == -1)

{

printf("要修改的信息不存在\n");

}

else

{

printf("请输入名字:>");

scanf("%s", ps->data[pos].name);

printf("请输入年龄:>");

scanf("%d", &ps->data[pos].age);

printf("请输入性别:>");

scanf("%s", ps->data[pos].sex);

printf("请输入电话:>");

scanf("%s", ps->data[pos].tele);

printf("请输入地址:>");

scanf("%s", ps->data[pos].addr);

printf("修改完成\n");

}

}

void ShowContact(const struct Contact \* ps)

{

if (ps->size == 0)

{

printf("通讯录为空格\n");

}

else

{

int i = 0;

printf("%-20s\t%-4s\t%-5s\t%-12s\t%-20\n", "名字", "年龄", "性别", "电话", "地址");

for (i = 0; i < ps->size; i++)

{

printf("%%-20s\t%-4s\t%-5s\t%-12s\t%-20\n",

ps->data[i].name,

ps->data[i].age,

ps->data[i].tele,

ps->data[i].sex,

ps->data[i].addr);

}

}

}

static int FindByName(struct Contact \*ps, char name[MAX\_NAME])

{

int i = 0;

for (i = 0; i < ps->size; i++)

{

if (0 == strcmp(ps->data[i].name, name))

{

return i;

}

}

return -1;

}

void SortContact(struct Contact \* ps)

{

}

void SaveContact(Contact \*ps)

{

FILE \*pfwriter = fopen("contact.dat", "wb");

if (pfwriter == NULL)

{

printf("SaveContact::%s\n",strerror(errno));

return;

}

int i = 0;

for (i = 0; i < ps->size; i++)

{

fwriter(ps->data[i],sizeof(PeoInfo),1,pfwriter);

}

fclose(pfwriter);

pfwriter = NULL;

}

void LoadContact(Contact \*ps)

{

PeoInfo tmp = {0};

FILE \*pfRead = ("txet.dat","rb");

if (pfRead == NULL)

{

printf("%s\n",strerror(errno));

return;

}

while (fread(&tmp,sizeof(PeoInfo),1,pfRead))

{

CheckContact(ps);

ps->data[ps->size] = tmp;

ps->size++;

}

fclose(pfRead);

pfRead = NULL;

}

454.#include<stdio.h>

#define SEEK\_CUR 0

int main()

{

FILE \* pf = ("text.txt","wb");

if (pf == NULL)

{

return 0;

}

fseek(pf, 4, SEEK\_CUR);

int ch = fgetc(pf);

printf("%c\n",ch);

fclose(pf);

pf = NULL;

return 0;

}

455.#include<stdio.h>

#define SEEK\_END 0

int main()

{

FILE \* pf = ("text.txt","r");

if (pf == NULL)

{

return 0;

}

fseek(pf, -2, SEEK\_END);

int pos = ftell(pf);

printf("%c\n",pos);

fclose(pf);

pf = NULL;

return 0;

}

456.#include<stdio.h>

#define SEEK\_END 0

int main()

{

FILE \* pf = ("text.txt","r");

if (pf == NULL)

{

return 0;

}

int ch = fgetc(pf);

printf("%c\n",ch);

rewind(pf);

ch = fgetc(pf);

printf("%c\n",ch);//printf("%d\n",ch);

fclose(pf);

pf = NULL;

return 0;

}

457.#include<stdio.h>

int main()

{

FILE \*p = ("text2.txt","r");

if (p == NULL)

{

perror("hehe\n");

return 0;

}

fclose(p);

p = NULL;

return 0;

}

458.#include<stdio.h>

int main()

{

FILE \*p = ("test.txt","r");

if (p == NULL)

{

perror("open file text2.txt");

return 0;

}

int ch = 0;

while (ch = fgetc(p) != EOF)

{

putchar(ch);

}

if (ferror(p))

{

printf("error\n");

}

else if (feof(p))

{

printf("end of file\n");

}

fclose(p);

p = NULL;

return 0;

}

459.#include<stdio.h>

struct S

{

int a;

char b;

short c;

short d;

}AA\_T;

int main()

{

printf("%d\n",sizeof(AA\_T));

return 0;

}//12

460.#include<stdio.h>

struct A

{

int a;

short b;

int c;

char d;

};

struct B

{

int a;

short b;

char c;

int d;

};

int main()

{//struct A a

printf("%d\n",sizeof(struct A));//printf("%d\n",sizeof(a));

printf("%\d\n",sizeof(struct B));

return 0;

}//16

//12

461.#include<stdio.h>

int main()

{

int arr[10] = { 0 };

int i = 1;

for(i = 0; i < 10; i++)

{

arr[i] = i;

}

for (i = 0; i < 10; i++)

{

printf("%d",arr[i]);

}

return 0;

}//0123456789

1. //text.h

int Add(int x, int y);

//text.c

#include<stdio.h>

extern int Add(int x, int y);int main()

{

int a = 10;

int b = 20;

int arr[10] = { 0 };

int i = 1;

for(i = 0; i < 10; i++)

{

arr[i] = i;

}

for (i = 0; i < 10; i++)

{

printf("%d",arr[i]);

}

int c = Add(a,b);

printf("c=%d\n",c);

return 0;

}//0123456789

//contact.c

int Add(int x, int y)

{

return x + y;

}

463.#include<stdio.h>

int g\_val = 1000;

int main()

{

int i = 0;

int arr[10] = { 0 };

for (i = 0; i < 10; i++)

{

arr[i] = i;

}

for (i=0;i<10;i++)

{

printf("%d",arr[i]);

}

return 0;

}//0123456789

464.#include<stdio.h>

#define MAX 100

int main ()

{

printf("%s\n",\_\_FILE\_\_);

printf("%d\n",\_\_LINE\_\_);

printf("%s\n",\_\_DATE\_\_);

printf("%s\n",\_\_TIME\_\_);

return 0;

}//c/\*:\users\86175\desktop\31＋xh\c语言\text\_2022\_3\_23(2).c\text\_2022\_3\_23(2).c\text.c

////6

////Mar 25 2022

////15:56 : 56\*/

465.#include<stdio.h>

#define MAX 100

int main ()

{

int i = 0;

int arr[10] = { 0 };

FILE \* pf = ("text,txt","w");

for (i = 0 ;i < 10 ; i++)

{

arr[i] = i;

fprintf(pf, "file:%s line :%d date:%s time:%s", \_\_FILE\_\_,\_\_LINE\_\_,\_\_DATE\_\_,\_\_TIME\_\_,i);

}

fclose(pf);

pf = NULL;  
//for (i = 0; i < 10; i++)

// {

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

// }

return 0;

}

466.#include<stdio.h>

int main()

{

printf("%d\n",\_\_STDC\_\_);

return 0;

}

467.#include<stdio.h>

#define MAX 100

#define STR "hehe"

#define reg register

#define do\_forever for(;;)

int main()

{

reg int a;

int max = MAX;

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

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

for (;;);

return 0;

}//100 hehe

468.#include<stdio.h>

#define MAX 100

int main()

{

int a = 100;

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

return 0;

}//100

469.#include<stdio.h>

#define SQUARE(X) X\*X

int main()

{

int ret = SQUARE(5);

printf("%D\n",ret);

return 0;

}//D

470.#include<stdio.h>

#define MAX 100

#define DOUBLE(X) ((X)\*(X))

int main()

{

int a = 5;

int ret =10 \* DOUBLE(MAX);

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

return 0;

}//100000

471.#include<stdio.h>

#define PRINT(X) printf("the value of "#x" is %s\n",X )

int main()

{

int a = 10;

int b = 20;

PRINT(a);

PRINT(b);

return 0;

}//语法错误

472.#include<stdio.h>

#define CAT(X,Y) X##Y

int main ()

{

int Class84 = 2019;

printf("%d\n",CAT(Class,84));

return 0;

}//2019

473.#include<stdio.h>

#define MAX(X,Y) ((X)>(Y)?(X):(Y))

int main()

{

int a = 10;

int b = 11;

int max = MAX(a++,b++);

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

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

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

return 0;

}/\*12

11

13\*/

474.#include<stdio.h>

int MAX(int x, int y)

{

return x + y;

}

#define MAX(X,Y) ((X)>(Y)?(X):(Y))

int main()

{

int a = 10;

int b = 20;

int max = MAX(a,b);

printf("max=%d\n",max);

printf("max=%d\n",max);

return 0;

}/\*max=20

max=20\*/

475.#include<stdio.h>

int Max(int x, int y)

{

return (x>y?x:y);

}

#define MAX(X,Y) ((X)>(Y)?(X):(Y))

int main()

{

int a = 10;

int b = 20;

float c = 3.0f;

float d = 4.0f;

int max = Max(c,d);

printf("max=%d\n",max);

max = MAX(c,d);

printf("max=%d\n",max);

return 0;

//}/\*max=20

//max=20\*/max = 4max = ((c) > (d) ? (c) : (d));

//max = 4

476.

#include<stdio.h>

#define TEST(x,y) printf("test\n")

#define MALLOC(num,type) (type\*)malloc(num\*sizeof(type))

int main()

{

int \*p = (int \*)malloc(sizeof(int) \* 10);

//int \*p = MALLOC(10, int );

TEST();

TEST();

EEST();

return 0;

}

477.#include<stdio.h>

#define MAX 100

int amin()

{

printf("MAX=%d\n",MAX);

#undef MAX

printf("MAX=%d\n",MAX);

return 0;

}

478.#include<stdio.h>

#define SZ

int main()

{

int i = 0;

int arr[SZ] = { 0 };

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

{

arr[i] = i;

}

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

{

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

}

return 0;

}

479.#include<stdio.h>

#define DEBUG

int main()

{

int arr[10] = { 1,2,3,4,5,6,7,8,9,0 };

int i = 0;

for (i = 0; i < 10; i++)

{

arr[i] = i;

#ifdef DEBUG//#if 1+1

printf("%d", arr[i]);

#endif

}

return 0;

}//0123456789

480.#include<stdio.h>

int main()

{

#if

printf("hehe\n");

#elif

printf("haha\n");

#else

printf("嘿嘿\n");

#endif

return 0;

}

481.#include<stdio.h>

#define DEBUG 0

int main()

{

#if defined( DEBUG)

printf("hehe\n");

#elif

printf("haha\n");

#else

printf("嘿嘿\n");

#endif

return 0;

}//hehe

482.#include<stdio.h>

int main()

{

#ifdef DEBUG

printf("hehe\n");

#endif

return 0;

}

483.#include<stdio.h>

#include "stdio.h"

#include "contact.h"

int Add(int x, int y)

{

return x + y;

}

int mian()

{

int ret = Add(2,3);

printf("ret=%d\n",ret);

return 0;

}

484.#include<stdio.h>

#include "stdio.h"

#include "contact.h"

//int Add(int x, int y)

//{

// return x + y;

//}

#define OFFSETOF(struct\_name,member\_name) (int)&(((struct\_name\*)0)->member\_name)

struct S

{

char c1;

int a;

char c2;

};

int mian()

{

printf("%d\n", OFFSETOF(struct S ,c1));

printf("%d\n",OFFSETOF(struct S ,a));

printf("%d\n",OFFSETOF(struct S ,c2));

return 0;

return 0;

}

485.#include<stdio.h>

int main()

{

int \*p = (int \*)malloc(0);

return 0;

}

486.