



第六部分 综合测试题

综合测试题一

【说明】 试卷中的“□”表示空一格,答题必要时也用“□”表示一个空格。

一、填空题(表达式求值)(本大题共 10 小题,每小题 1 分,共 10 分) 设各语句的初始化相同; `int x=3,y=2,z=1;`

- | | |
|---|------------------------|
| 1. <code>x=y==z;</code> | <code>x=()</code> |
| 2. <code>x=!(z>y)&&!x 1;</code> | <code>x=()</code> |
| 3. <code>x=(y++*1/3);</code> | <code>x=()</code> |
| 4. <code>x=((x>y>z)?1:0);</code> | <code>x=()</code> |
| 5. <code>x*=2+3;</code> | <code>x=()</code> |
| 6. <code>x=(++z*y,y++,z%y);</code> | <code>x=()</code> |
| 7. <code>y=-z&&-x;</code> | <code>x=()</code> |
| 8. <code>x=5 x!=y;</code> | <code>x=()</code> |
| 9. <code>x=1.5+7%3*(int)(2.5+4.7)%2/4;</code> | <code>x=()</code> |
| 10. <code>x='a'-'A'</code> | <code>x=()</code> |

二、阅读下列程序段,指出错误并改正(本大题共 5 题小题,每小题 4 分,共 20 分)

说明:在错误处加下划线,然后将改正内容写在行末。

1.

```
#include <stdio.h>
#define MAX(a,b) (a)=b? a:b
void main( ){
    int i;
    char a[6]={'a','1','b','2','c','3'};
    char b[6]="234def";
    for(i=0; a[i]!='\0'; i++)
        printf("i=%d,max=%c\n",i, MAX(a[i],b[i]));
}
```
2.

```
#include <stdio.h>
void main( ){
    int a=9; double m;
    m=f1(3.14159, a);
    printf("%f\n",m);
```



```
    }  
    double f1(int i, double j){  
        return(i*i+j*j);  
    }  
3. #include <stdio. h>  
void main( ){  
    int i; char * str[5];  
    for(i=0; i<5; i++) scanf("%s",str[i]);  
    for(i=0; i<5; i++) printf("%s",str[i]);  
}  
4. #include <stdio. h>  
void main( ){  
    int i, r[10],sum;  
    for(i=0; i<=10; i++){  
        r[i]=i+1;  
        sum+=r[i];  
    }  
    printf("sum=%d",sum);  
}  
5. #include <stdio. h>  
void main( ){  
    int i;  
    struct student{  
        int num=0;  
        float score1=0, score2=0, aver=0;  
    } stu[3]={ {200401,92,83}, {200402,85,70}, {200403,65,89}};  
    for(i=0; i<3;i++){  
        stu[i]. aver=(stu[i]. score1+stu[i]. score2)/2;  
        printf("%d,%f\n",stu[i]. num,stu[i]. aver);  
    }  
}
```

三、阅读分析下列程序,写出运行结果(本大题共7小题,每小题4分,共28分)

```
1. #include <stdio. h>  
void main( ){  
    int i,f1=1,f2=1;  
    for(i=1;i<=4;i++){  
        printf("%6d%6d",f1,f2);  
        if(i%2==0)  
            printf("\n");  
    }  
}
```

1. 运行结果:



```
        f1=f1+f2;
        f2=f2+f1;
    }
}
```

```
2. #include "stdio. h"
void main( ){
int i,x;
for(i=1, x=1; i<7; i++){
    if(x>=7) break;
    if(x%2==1){
        x+=5; continue;
    }
    x-=3;
    printf("i=%d,x=%d\n",i,x);
}
printf("i=%d,x=%d\n",i,x);
}
```

2. 运行结果:

```
3. #include <stdio. h>
fun(int a){
    int b=1;
    static int c=1;
    a++; b++; c++;
    return(a+b+c);
}

void main( ){
    int i=1;
    for( ; i<=3; i++)
        printf("%6d",fun(i));
}
```

3. 运行结果:

```
4. #include <stdio. h>
void main( ){
    int i,j;
    char c, a[ ]="computer", *p=a;
    printf("%s",a);
    for(i=0,j=7; i<j; i++,j--){
        c=p[i];
        *(p+i)=*(p+j);
        p[j]=c;
    }
}
```

4. 运行结果:

```
printf("> %s", a);
}
```

```
5. #include <stdio.h>
void swap2(int * pa, int * pb){
    int * t;
    t=pa; pa=pb; pb=t;
}
void swap3(int * pa, int * pb){
    int t;
    t= * pa; * pa= * pb; * pb=t;
}
void main( ){
    int x=1,y=3, * p1=&x, * p2=&y;
    swap2(p1,p2); printf("%d,%d\n",x,y);
    swap3(p1,p2); printf("%d,%d\n",x,y);
}
```

5. 运行结果:

```
6. #include "stdio.h"
void main( ){
    static struct st1{
        char c[4], * s;
    }s1={"abc","def"};
    static struct st2{
        char * cp;
        struct st1 ssl;
    }s2={"ghi",{"jkl","mno"}};
    printf("%c\t%c\n",s1.c[0], * s1.s);
    ++s2.cp;
    ++s2.ssl.s;
    printf("%s\t%s\n",s2.cp, s2.ssl.s);
}
```

6. 运行结果:

7. 有以下程序,编译后生成可执行程序文件 display。写出执行“display how are you!”命令行后的结果。

```
#include <stdio.h>
main(int argc, char * argv[ ]){
    int i;
    for(i=1; i<argc; i++)
        printf("%s%c", * ++argv, (i<argc-1? ' ': '\n'));
}
```

7. 运行结果:



四、程序填空题(本大题共 2 小题,每空 1 分,共 15 分)

1. 以下程序功能为建立链表及统计链表中所有节点的数据域 n 之和。

```

#include <stdio.h>
#include <malloc.h>
struct node{
    int n;
    struct node * next;
};
struct node * creat( ) { //建立链表
    struct node * head, * p1, * p2;
    head=_____ ;
    p1=p2=_____ ;
    scanf("%d", &p1->n);
    while( p1->n!=0 ){
        if( head==NULL )
            head=_____ ;
        else
            p2->next=_____ ;
        p2=p1;
        p1=_____ ;
        scanf("%d",&p1->n);
    }
    p2->next=_____ ;
    return(_____ );
}
add(_____ ) { //统计链表中所有节点的数据域 n 之和
    struct node * p; int n=0;
    p=head;
    while(p!=NULL){
        n=_____ ;
        p=_____ ;
    }
    return(n);
} //主函数 main()见后第 7 页
main( ) {
    struct node * head; int m;
    head=creat( );
    m=add( head );
    printf("%d\n", m);
}

```



2. 已知正文有文件“abc.txt”, 本题统计该文件中词的个数并求出最长词的长度。正文文件中, 词与词之间由空格符, 制表符, 换行符分隔。程序中, word 存放词的个数, maxLen 存放最长词的长度, length 存放当前词的长度, flag 为状态标记(flag=1 表示当前字符在单词中, flag=0 表示当前字符不在某词中)。

```
#include <stdio. h>
#include <process. h>
void main( ){
    FILE * fp; char ch;
    int word=0, maxLen=0, length=0, flag=0;
    if((fp=fopen( _____ ))==NULL){
        printf("cannot open the file! \n");
        exit(0);
    }
    ch=fgetc(fp);
    while(ch!=EOF){
        if(ch==' ' || ch=='\t' || ch=='\n'){           //‘ ’表示空格
            if(flag==1){
                flag=0;
                if(length>maxLen) _____ ;
            }
        }
        else if(flag==0){
            flag= _____ ;
            length=1;
            word _____ ;
        }
        else
            length++;
        ch= _____ ;
    }
    fclose(fp);
    printf("word=%d, maxLen=%d\n", word, maxLen);
}
```

五、程序设计题 (本大题共 2 小题, 第 1 小题 12 分, 第 2 小题 15 分, 共 27 分)

1. 根据主函数的定义, 编写以下两个函数:

- (1) 求数组 a[4][4] 主对角线上元素的和; (6 分)
- (2) 实现将数组 a[4][4] 的矩阵转置处理。 (6 分)

```
void main(){
```



```
static int a[4][4] = {1,1,1,1,2,2,2,2,3,3,3,3,4,4,4,4};
int *p, i, sum;
sum = total(a, 4);           //调用 total 函数求主对角线上元素的和
printf("%d\n", sum);
p = *a;
convert(p);                 //调用 convert 实现矩阵转置
for(i=0; i<4; i++)
    printf("%d %d %d %d\n", a[i][0], a[i][1], a[i][2], a[i][3]);
}
```

2. 编一个程序,用结构体数组描述 60 位学生的信息,包括:学号、姓名、三门课的成绩和总成绩。并以函数形式实现以下功能:(15 分)

- (1) 输入 60 位学生的前五项信息,并计算每个学生的总成绩。
- (2) 按总成绩递减顺序对成绩表排序。
- (3) 输出排序后的成绩表。

综合测试题二

一、选择题(本大题共 10 小题,每小题 1 分,共 10 分)

1. C 语言中用于结构化程序设计的三种基本结构是_____。
 - A. 顺序结构、选择结构、循环结构
 - B. if、switch、break
 - C. for、while、do-while
 - D. if、for、continue
2. 下列关于 C 语言用户标识符的叙述中正确的是_____。
 - A. 用户标识符中可以出现下划线和中划线
 - B. 用户标识符中不可以出现中划线,但可以出现下划线
 - C. 用户标识符中可以出现下划线,但不可以放在用户标识符的开头
 - D. 用户标识符中可以出现下划线和数字,他们都可放在用户标识符的开头
3. 有以下程序:

```
main(){
    char a,b,c,*p;
    a='\'; b='\xbc'; c='\0xab'; p="\0127";
    printf("%c %c %c %c\n",a,b,c,*p);
}
```

编译时出现错误,以下叙述中正确的是_____。

- A. 程序中只有 a='\';语句不正确
- B. b='\xbc';语句不正确
- C. c='\0xab';语句不正确
- D. a='\';和 c='\0xab';语句都不正确

4. 有以下程序段:

```
int m=0,n=0; char c='a';
scanf("%d%c%d",&m,&c,&n);
printf("%d,%c,%d\n",m,c,n);
```

若从键盘上输入:12A10<回车>,输出结果是_____。

- A. 12,A,10 B. 12,a,10 C. 12,a,0 D. 12,A,0

5. 设有定义:double x; ,则以下正确的输入语句是_____。

- A. scanf("%f",x); B. scanf("%f",&x);
 C. scanf("%lf",&x); D. scanf("%lf",x);

6. 有以下程序:

```
void main(){
    char *s[]={"one","two","three"},*p;
    p=s[1];
    printf("%c,%s\n",*(p+1),s[0]);
}
```

执行后的结果是_____。

- A. n,two B. w,one C. t,one D. o,two

7. 设已有定义:char a[10], *p=a; 下面的赋值语句中,正确的是_____。

- A. a[10]="Turbo C"; B. a="Turbo C";
 C. p="Turbo C"; D. *p="Turbo C";

8. 设有已定义:int n=100, *p=&n, *q=p; 则以下不正确的赋值语句是_____。

- A. p=q; B. *p=*q; C. p=n; D. n=*q;

9. 主调函数中已有定义:int a[3][4],n; 当使用函数调用语句:fun(a,n); 时,被调函数fun()的形参说明表列正确的为_____。

- A. fun(int a[3][],int n){...} B. fun(int a[][4],int n){...}
 C. fun(int a[][],int n){...} D. fun(int *p[4],int n){...}

10. C 程序由函数组成,关于 C 语言函数的说法错误的是_____。

- A. 除整型函数外,其他类型函数必须在定义时给予类型说明
 B. 函数原型说明语句必须给出函数类型的说明
 C. 函数类型决定返回值的类型
 D. 函数返回值的类型就是 return 语句中的表达式值的类型

二、填空题(本大题共 8 小题共 10 个填空,每空 1 分,共 10 分)

1. 试将数学公式 $\left| \cos \frac{\pi}{2} \right|$ 写成 C 表达式为:

2. 设 x, n 已定义为 double 类型变量,试将数学公式 $\frac{3x^n}{2x-1}$ 写成 C 表达式为:



3. 若有定义: `int a=7; float x=2.5, y=4.7;`
 执行 `printf("%f\n", x+a%3*(int)(x+y)%2/4);` 的结果为: _____
4. 若有定义: `int a=3, b=4;`
 执行 `printf("%d\n", (a=++b, a+5, a/5));` 的结果为: _____
5. 若有定义: `int a=3, b=4, c=5;`
 执行 `printf("%d\n", a* =b+c);` 的结果为: _____
6. 若有定义: `int p, a=5;`
 执行 `if(p=a!=0)printf("%d\n", p);`
`else printf("%d\n", p+2);`
 的结果为: _____

7. 若有定义: `int a[3][4]={1,2,3,4,5,6,7,8,9,10,11,12}, (*p)[4]=a;`
 执行 `printf("%d\n", *(*(p+1)+3));` 的结果为: _____

8. 某地刑侦大队对涉及 6 个嫌疑人的一桩疑案进行分析:

- ① A, B 至少有一人作案;
- ② A, E, F 3 人中至少有 2 人参与作案;
- ③ A, D 不可能是同案犯;
- ④ B, C 或同时作案, 或与本案无关;
- ⑤ C, D 中有且仅有一人作案;
- ⑥ 如果 D 没有参与作案, 则 E 也不可能参与作案。

为解这道题, 需要将案情的每一条写成逻辑表达式, 如变量 A 表示作案, !A 表示不作案, 从第 1 条到第 6 条依次用 `cc1, ……`, `cc6` 表示, 请参照以下已经给出的第一个赋值语句形式, 请填写余下的 5 个赋值语句中空缺的逻辑表达式:

```
cc1=(A || B); // A, B 至少有一人作案
cc2 = _____; // A, E, F 3 人中至少有 2 人参与作案
cc3 = _____; // A, D 不可能是同案犯
cc4 = _____; // B, C 或同时作案, 或与本案无关
cc5 = _____; // C, D 中有且仅有一人作案
cc6 = _____; // 如果 D 没有参与作案, 则 E 也不可能参与作案
```

三、阅读并分析下列程序, 写出运行结果(本大题共 5 小题, 每题 6 分, 共 30 分)

1. `#include <stdio. h>`

```
void main(){
    int a[4][4]={{1, 2, -3, -4}, {0, -12, -
                13, 14}, {-21, 23, 0, -24},
                {-31, 32, -33, 0}};

    int i, j, s=0;
    for(i=0; i<4; i++){
        for(j=0; j<4; j++){
```

1. 运行结果:



```
        if(a[i][j]<0) continue;
        if(a[i][j]==0) break;
        s += a[i][j];
    }
    printf("%d\n",s);
}
```

```
2. #include <stdio.h>
   #define UDF_SWAP(a,b) t=a;a=b;b=t;
   void udf_swap(int a,int b){
       int t;t=a;a=b;b=t;
   }
   void main(){
       int x=2,y=3,t=0;
       UDF_SWAP(x,y);
       printf("%d  %d\n",x,y);
       x=2,y=3;
       udf_swap(x,y);
       printf("%d  %d\n",x,y);
   }
```

2. 运行结果:

```
3. #include <stdio.h>
   void mul(void){
       int i=1; static int a=1;
       while(i<=a){
           printf("%5d",a * i); i++;
       }
       a++; printf("\n");
   }
   void main(){
       int a,b; for(a=1;a<4;a++)mul();
   }
```

3. 运行结果:

```
4. #include <stdio.h>
   void main(){
       int i=0,CountX=0,CountY=0;
       char str[80]="The []Programing []Language";
       while(str[i]!='\0'){
           if(str[i]>='a'&&str[i]<='z')
               str[i]-=32;
           switch(str[i]){
```

4. 运行结果:



```

        case 'A':
        case 'E':
        case 'I':
        case 'O':
        case 'U': CountX++;break;
        default : CountY++;break;
    }
    i++;
}
printf("CountX=%d\n",CountX);
printf("CountY=%d\n",CountY);
}
5. #include <stdio. h>
#include <malloc. h>
char * fun1(char * a,int start,int len){
    char * ans; int i;
    ans=(char *)malloc(strlen(a)+1);
    for(i=start;i<start+len;i++)
        ans[i-start]=a[i];
    ans[len]='\0';
    return(ans);
}
void fun2(char * s,char * t){
    while(*s!='\0')s++;
    while((*s++=*t++)!='\0');
}
void main(){
    char a[80]="ABCDEFGHJIJ", b[80]="12345678910", *p;
    int i,j;
    for(i=0,j=1;i<2;i++){
        p=fun1(a,i,++j);
        printf("%s\n",p);
        fun2(p,fun1(b,i,j));
        printf("%s\n",p);
    }
}

```

5. 运行结果：

四、程序填空题 (本大题 2 题共 5 个填空,每空 2 分,共 10 分)

1. 下面的程序把从终端读入的文本复制到一个名为 file1. dat 的新文件中。请在划线



处填空。

```
#include <stdio. h>
#include <process. h>
void main(){
    FILE * fp; char ch;
    if((fp=fopen(_____)) == NULL)
        exit(0);
    while((ch=getchar()) != EOF)
        _____;
    fclose(fp);
}
```

2. 以下程序,数组 a 中存放一个递增数列。输入一个整数 x 并将它插入到数组 a 中,使之仍为一个递增数列。请在划线处填空。

```
#include <stdio. h>
void main(){
    int a[ _____ ]={1,10,20,30,40,50,60,70,80,90};
    int x,i,p;
    scanf("%d",&x);
    for(i=0,p=10;i<10;i++)
        if(x<a[i]){
            p=i;
            _____;
        }
    for(i=9;i>=p;i--)
        a[i+1]=a[i];
    _____;
    for(i=0;i<=10;i++)
        printf("%5d",a[i]);
    printf("\n");
}
```

五、程序设计题(本题共 2 小题,第 1 题 25 分,第二题 15 分,共 40 分)

1. 编程实现以链表形式处理学生选修课程信息。每一位学生选修课程的记录信息包括:学号 sno、课程号 cno,成绩 score;相应数据项如:“PB07210001”,“C0020101”, 95.0。请按要求编写各指定功能函数。相应预处理命令、数据类型、主函数框架等已给出,请按要求编写各指定功能函数。

```
#include <stdio. h>
#include <malloc. h>
#include <process. h>
```



```

struct student{                               //结构体类型定义
    char sno[16], cno[12];
    float score;
    struct student * next;
};
main(){                                       //主函数供编程参考
    struct student * head=NULL, * s;
    char search_sno[16];
    head=create();                           //调用 create()建立链表
    printf("输入查询学号:\n");
    scanf("%s", search_sno);
    serch_print(head, search_sno);          //调用 serch_print()检索选修信息
}

```

(1) 编写一个计算字符串长度的 `str_len()` 函数,要求为:函数的返回值为字符串长度。(4分)

(2) 编写一个字符串比较 `str_cmp()` 函数,设有字符串 `s` 与 `t`,若 `s` 大于 `t` 则返回正值,`s` 等于 `t` 则返回 0,`s` 小于 `t` 则返回负值。(5分)

(3) 编写一个建立链表的 `create()` 函数,要求为:建立结点个数不限,当输入记录的学号字符串长度小于等于 1 时结束,如输入:0 0 0 时结束;函数的返回值为指向链表 `head` 指针。(10分)

(4) 编写一个按学号检索学生选修课程信息的 `serch_print()` 函数,要求:能按指定学号检索该学生所有的选修课程信息,即打印出所有与指定学号相关结点的成员信息。(6分)

2. 如图 6.1 所示,已知某游泳池的长度为 50 米,宽度为 25 米。某人游泳速度为 0.8 米/秒,步行速度为 1.2 米/秒。编程求解从 A 点到 B 点间何处下水游到 C 点时间最短,输出该点距 A 点的长度 x 和所求最短时间 y 。要求:

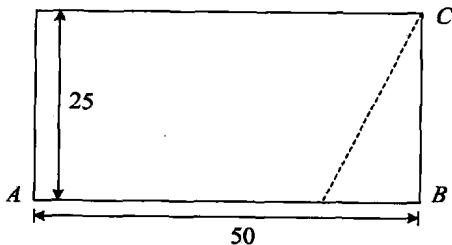


图 6.1 游泳池入水点问题示意图

(1) 给出求解该问题的算法描述。(7分)

(2) 根据算法编写完整的求解程序。(8分)

说明:距离精度控制在 0.1 米或以下。

综合测试题三

一、选择题(本大题共 10 小题,每小题 1 分,共 10 分)

1. 与数学公式 $\frac{3x^n}{2x-1}$ 对应的 C 语言表达式是_____。

A. $3 * x^n / (2 * x - 1)$

B. $3 * x * * n / (2 * x - 1)$

C. $3 * \text{pow}(x, n) / (2 * x - 1)$

D. $3 * \text{pow}(n, x) * (2 * x - 1)$

2. 设 `a` 为整型变量,不能正确表达数学关系 $10 < a < 15$ 的 C 语言表达式为_____。



D. *p 只能用来说明 p 是一个指针变量

二、填空题(本大题共 8 小题共 10 个填空,每空 1 分,共 10 分)

1. 变量存储类别,从作用域角度分有_____变量和_____变量。
2. 设整型变量 x, y, z 均为 5;执行 $x - = y - z$ 后, $x =$ _____。
3. 设 m, n, a, b, c, d 均为 0, 执行 $(m = a = b) \parallel (n = c = d)$ 后, m, n 的值分别为:_____。
4. 若有定义: $\text{int } a = 1, b = 2;$
执行 $\text{printf}(\text{"\%d\n"}, (a = ++b, a + 5, a / 5));$ 的结果为_____。
5. 设 ch 是 char 型变量,其值为 'A', 则执行语句
 $\text{ch} = (\text{ch} > = 'A' \&\& \text{ch} < = 'Z') ? (\text{ch} + 32) : \text{ch}$ 后, ch 的值为_____。
6. 若有定义: $\text{int } x = 3, y = 5;$
执行 $\text{printf}(\text{"\%d\n"}, (x > = y > = 2) ? 1 : 0);$ 的结果为_____。
7. 若定义 i, j 为 int 型, 则以下程序段中内循环体的执行次数为_____。

```
for (i = 5; i ; i -- )
    for (j = 0; j < 4 ; j ++ )
        { ... }
```
8. 设有定义语句

```
struct {
    int x;
    int y;
} d[2] = { { 1, 3 }, { 2, 7 } };
```

 则 $\text{printf}(\text{"\%d\n"}, d[0].y / d[0].x * d[1].x)$ 的输出是_____。

三、阅读程序题:请仔细阅读、分析下列程序,按程序实际执行情况,写出运行结果(本大题共 7 小题,每题 5 分,共 35 分)

```
1. #include <stdio.h>
void main(){
    int a, b;
    for(a = b = 1; a <= 100; a = a + 1){
        if(b >= 10){
            printf("\%d\n", b); break;
        }
        if(b % 3 == 1){
            printf("\%d\n", b = b + 3);
            continue;
        }
    }
    printf("\%d\n", a);
}
```

1. 运行结果:



```
}
```

```
2. #include <stdio.h>
void main(){
    int i=0,j,n=7788,base=8,num[20];
    do{ i++;
        num[i]=n%base;
        n=n/base;
    }while(n!=0);
    for(j=i;j>=1;j--) printf("%d",num[j]);
    printf("\n");
}
```

2. 运行结果:

```
3. #include<stdio.h>
int n=2;
int fun(int n){
    int a=1;static int m=5;
    m = n + m + a;
    printf("%4d%4d\n",n,a);
    return(m);
}
```

3. 运行结果:

```
void main(){
    int a=1;
    printf("%4d\n",fun( n+fun(a)));
}
```

```
4. #include <stdio.h>
void main(){
    int k=1; char c='A';
    do{
        switch(++c){
            case 'A': k++;printf("%4d",k); break;
            case 'B': k--;printf("%4d",k);
            case 'C': k+=2; printf("%4d",k); break;
            case 'D': k%2;printf("%4d",k); break;
            case 'E': k=k*2; printf("%4d",k); break;
            default : k=k/3; printf("%4d",k);
        }
    }while(c<'E');
    printf("\n");
}
```

4. 运行结果:



```
5. #include<stdio. h>
void write_it( char * s ){
    printf("%s\n",s);
    if( * s! ='a')
        write_it( s+1 );
    putchar( * s );
}
void main(){
    char str[40]="image";
    write_it( str );
    printf("\n");
}
```

5. 运行结果:

```
6. #include <stdio. h>
#define SWAP(x,y) temp=x; x=y; y=temp
void swap(int * p, int * q){
    int temp;
    temp= * p; * p= * q; * q=temp;
}
void main(){
    int a=1,b=3, * p=&a, * q=&b, * temp;
    printf("%d,%d\n", * p, * q);
    SWAP(p,q); printf("%d,%d\n", * p, * q);
    swap(p,q); printf("%d,%d\n", * p, * q);
}
```

6. 运行结果:

```
7. #include <stdio. h>
void main(){
    int a[2][3]={{1,2,3},{4,5,6}};
    int b[3][2]={{3,2},{2,0},{1,4}};
    int i,j,k,sum=0,c[2][3];
    for(i=0;i<2;i++)
        for(j=0;j<2;j++)
            for(c[i][j]=k=0;k<3;k++)
                c[i][j]+=a[i][k] * b[k][j];
    for(i=0;i<2;i++){
        for(j=0;j<2;j++){
            printf("%d\t",c[i][j]);
            if(i==j || i+j==1)sum+=c[i][j];
        }
    }
    printf("\n");
}
```

7. 运行结果:



```

    }
    printf("%d\n", sum);
}

```

四、程序填空题 (本大题含 3 小题共 15 分, 每空 1 分)

1. 下面程序的功能是通过调用函数 f 计算代数多项式当 $a=1.7$ 时的值。代数多项式为: $1.1+2.2*a+3.3*a*a+4.4*a*a*a+5.5*a*a*a*a$, 根据程序功能填空。

```

#include <stdio.h>
double f(double x, _____){
    double y=a[0], t=1;
    int i;
    for(i=0; i<5; i++){
        t= _____ ;
        y=y+a[i]*t;
    }
    return(_____);
}
void main(){
    double b[5]={ 1.1, 2.2, 3.3, 4.4, 5.5 };
    printf("%lf\n", f(1.7, b, 5));
}

```

1				
1	1			
1	2	1		
1	3	3	1	
1	4	6	4	1
...

图 6.2 杨辉三角形

2. 函数 YangHui 的功能是把杨辉三角形的数据赋给二维数组的下半三角, 形式如图 6.2 框内所示, 其构成规律是:

- (1) 第 0 列元素和主对角线元素均为 1。
- (2) 其余元素为其左上方和正上方元素之和。
- (3) 数据的个数每行递增 1。

请将程序补充完整。

```

#define N 6
void YangHui(int x[N][N]){
    int i, j;
    x[0][0]=1;
    for(i=0; i<N; i++){
        x[i][0]=_____ = 1;
        for(j=1; j<i; j++){
            x[i][j]=_____ ;
        }
    }
}

```

3. 本程序首先定义三个函数:



(1) string_to_list 函数,根据一个字符串创建一个链表,每一结点存储一个字符。

(2) print_list 函数,用于输出链表。

(3) concatenate_list 函数,功能为将两个链表连接在一起。

在 main 主函数中通过调用以上函数,建立两个存储“zhang”的链表 a 和存储“san”的链表 b;然后完成将 b 链表连接到 a 链表之后,最后打印输出连接之后的 a 链表,输出结果应为“zhangsan”。请根据程序功能填空。

```
#include<stdio. h>
#include<malloc. h>
```

```
struct list{
    char data;
    structlist * next;
};
struct list * string_to_list( char s[] );
struct list * concatenate_list( struct list * a, struct list * b );
void print_list( struct list * a );
void main( ){
    struct list * head1, * head2;
    char s[]="zhang",t[]="san";
    head1 = string_to_list( s );
    head2 = string_to_list( t );
    concatenate_list( head1, head2 );
    print_list( head1 );
}
```

```
struct list * string_to_list( char s[] ){
    struct list * head=NULL, * p; int i;
    if( s[0] != '\0' ){
        head=_____ ;
        head->data=s[0];
        p=head;
        for( i=1; s[i] != '\0'; ++i ){
            p->next=_____ ;
            p=_____ ;
            p->data= s[i] ;
        }
        p->next=_____ ;
    }
    return(_____ );
}
```



```
    }

    struct list * concatenate_list( struct list * a, struct list * b ){
        struct list * p=a;
        if( p == NULL )
            _____ ;
        else{
            while( p->next != NULL )
                _____ ;
            p->next= _____ ;
        }
        return(a);
    }

    void print_list( struct list * head ){
        struct list * p = head;
        if( p == NULL )
            printf( "NULL! \n" );
        else
            do{
                printf( "%c", p->data );
                _____ ;
            }while( _____ );
        printf("\n");
    }
}
```

五、程序设计题 (本程序设计题,共 30 分)

建立一个 100 人的人口普查信息表,其中包括姓名、年龄、性别、职业及地址,相应结构体类型定义如下所示:

```
struct person{
    char name[30];
    int age;
    char sex, job[30], addr[40];
};
```

要求在主函数中定义一个结构体数组,并采用形实参数结合的数据传递方式定义以下指定功能的用户自定义函数:

- (1) 定义 read 函数,读入普查人员的五项数据。
- (2) 定义 udf_cmpstr 字符串比较函数(注:功能必须由自己具体定义实现,不要直接或间接通过调用库函数实现)。
- (3) 定义 sort 函数,按姓名以递升顺序排序。



(4) 定义 print 函数,将排序后的普查信息表输出到 person_list 文件长期保存。

(5) 定义 binarysearch 函数,以二分查找(亦称折半查找)方式实现按姓名查找该人员在人口普查信息表中的位置,若没有找到则返回-1,表示查找失败。

编写 main 函数,依次调用以上各功能函数,完成具体应用编程。其中调用 binarysearch 函数时,假设查找对象的姓名为“xiaoli”,返回主函数后即打印输出该人员在信息表中的位置,若没有找到则提示“notfound!”信息。

提示:按指定要求编程,结构体数组不要采用全局的结构体数组形式。

```
#include<stdio. h>
#include<process. h>
#define N 100
#define PERSON struct person
struct person{
    char name[30];
    int age;
    char sex, job[30], addr[40];
};
```

综合测试题四

一、选择题(本大题共 10 小题,每小题 1 分,共 10 分)

- 计算机可以进行自动处理的基础是_____。
 - 存储程序
 - 快速运算
 - 计算精度高
 - 能进行逻辑判断
- 计算机的通用性使其可以求解不同的算术和逻辑问题,这主要取决于计算机的_____。
 - 高速运算
 - 指令系统
 - 可编程性
 - 存储功能
- 在计算机内部用于表示数据和指令的是_____编码。
 - 十进制码
 - 二进制码
 - ASCII 码
 - 汉字编码
- 计算机进行数值计算时的高精度主要取决于_____。
 - 计算速度
 - 内存容量
 - 外存容量
 - 基本字长
- 已知字母 A 的 ASCII 码为十进制数 65,且 c2 为字符型,则执行以下语句 c2='A'+'6'-'3'; 后,c2 中的值为_____。
 - D
 - 68
 - 不确定
 - 'C'
- 设已有定义:char a[5], *p=a; 则下面的选项中正确的赋值语句是_____。
 - p="abcd";
 - a="abcd";
 - *p="abcd";
 - *a="abcd";
- 以下错误的描述是_____。
 - break 语句不能用于循环语句和 switch 语句外的任何其他语句
 - 在 switch 语句中使用 break 语句或 continue 语句的作用相同
 - 在循环语句中使用 continue 语句是为了结束本次循环
 - 在循环语句中使用 break 语句是为了使流程跳出循环体,提前结束循环



```
if(x > 6) break;
else if(x % 2 == 1){
    x += 5;
    printf("i=%d,x=%d\n", i, x);
    continue;
}
x -= i;
printf("i=%d,x=%d\n", i, x);
}
```

1. 运行结果:

```
2. #include <stdio.h>
#define PRODUCT(a, b) a * b
void fun(int n){
    static int x = 1;
    printf("x=%d\n", x + n);
    x += PRODUCT(x + n, x - n);
}
int main(){
    int i, x = 1;
    for(i = 1; i <= 3; i++, x++)
        fun(x + i);
}
```

2. 运行结果:

```
3. #include <stdio.h>
void udf_sort( int array[], int n ) {
    int i, j, k, temp;
    for(i = 0; i < n-1; i++){
        k = i;
        for(j = i+1; j < n; j++)
            if(array[j] < array[k])
                k = j;
        if(k != i){
            temp = array[k];
            array[k] = array[i];
            array[i] = temp;
        }
        for(j = 0; j < n; j++)
            printf("%d%c", array[j], (j+1)%5 ? ' ' : '\n');
    }
}
```

3. 运行结果:



```
int main(){
    int i;
    inta[5] = {2, 5, 3, 1, 4};
    udf_sort(a, 5);
    for(i = 0; i < 5; i++)
        printf("%d%c", a[i], (i+1)%5 ? ' ' : '\n');
}
```

4. #include <stdio. h>

```
int multiply(int m, int n){
    int ans;
    if(n == 1)
        ans = m;
    else
        ans = m + multiply(m, n-1);
    printf("%d\n", ans);
    return(ans);
}
```

4. 运行结果:

```
main(){
    printf("%d\n", multiply(6, 3));
}
```

5. #include<stdio. h>

```
void change(int *s[],int n){
    int i, j, t;
    for(i = 0; i < 4; i++)
        for(j = i; j < 4; j++){
            t = *(s[j] + i);
            *(s[j] + i) = *(s[i] + j);
            *(s[i] + j) = t;
        }
}
```

5. 运行结果:

```
main(){
    int a[4][4] = { {1,2,3,4}, {12,13,14,5},
                   {11,16,15,6}, {10,9,8,7} };
    int i, j, *p[4];
    for(i = 0; i < 4; i++)
        p[i] = a[i];
    change(p, 4);
}
```




```

        for(i = 0; i < 4; i++)
            for(j = 0; j < 4; j++)
                printf("%2d%c", *(p[i] + j), j < 3 ? ' ' : '\n');
    }
}

6. #include<stdio.h>
void silly1(int x){
    int y;
    y = x + 2;
    x = x * 2;
}
void silly2(int *x){
    int y;
    y = *x + 2;
    *x = 2 * *x;
}
main(){
    int x=10,y=11;
    silly1(x); printf("x=%d\n", x);
    silly1(y); printf("y=%d\n", y);
    silly2(&x); printf("x=%d\n", x);
    silly2(&y); printf("y=%d\n", y);
}

```

6. 运行结果:

四、程序设计题 (共三个程序设计题,第1题12分,第2题15分,第3题18分,共45分)

1. 已知一组实验数据:3.62,2.93,3.16,3.73,2.86,3.40,2.86,3.07,3.29,3.24,编一程序分别求出它们的平均值、方差和均方差。

要求:

(1) 定义一个求解平均值、方差和均方差的函数 `udf_vft()`,与主调函数间的数据联系要求采用参数传递方式或 `return` 语句实现,不要采用全局变量传递方式。

(2) 一组实验数据可由数组方式在主函数中构成;在主函数中通过调用 `udf_vft()` 函数得到所需要的平均值、方差和均方差并打印输出,每一结果只保留两位小数。

提示:设平均值、方差和均方差分别用变量 v , f 和 t 表示,由数学知识可知,相应的计算公式为:

$$v = \frac{1}{n} \sum_{i=1}^n x_i \quad f = \frac{1}{n} \sum_{i=1}^n x_i^2 - v^2 \quad t = \sqrt{f}$$

其中, n 表示数据个数, x_i 表示第 i 个数据。

2. 编写一个程序,实现对一段文本中特定的某个单词进行统计的功能。

提示:通常文字处理软件(例如 MS Office Word 或者 WPS 等)中都提供了字符串查找匹配功能。一个程序要达到这样的功能首先需要具备有关字符串操作的一些函数来辅助实

现这个功能。本题主函数编写已经完成,源程序代码清单如下,请编写与斜体字表示的函数调用对应的有关字符串操作函数,以实现这个功能。具体要求参照相关答题处。

设主函数位于其他函数之后且内容如下:

```
#include <stdio. h>
#include <string. h>
int main(){
    int i=0,sum=0,length;
    char word[20];
    char text[]="to modernize the country's industry,agriculture,national \
        defence and science and technology step by step to turn China into \
        a strong and prosperous socialist country with a high level of \
        culture and democracy. ";
    char temp[20]={0};
    printf("please input the word which you want to count. \n");
    udf_gets(word); //udf_gets(word)读字符串函数
    length=udf_strlen(word); // strlen(word)求串长函数
    while(length!=0 && text[i]!='\0'){
        strncpy(temp,text+i,length); //逐一取定长的子串
        if(udf_strcmp(word,temp)==0){ //udf_strcmp(word,temp)字符串比较函数
            sum++; //匹配成功统计数加 1
            i +=length; //下标 i 后移,为取下一个子串作准备
        }
        i++;
    }
    printf("The word \"%s\" appears %d times in the text. \n",word,sum);
}
```

编程要求(1):编写读字符串*udf_gets()*函数。

*udf_gets()*功能为:从标准输入文件 stdin 读入一串字符送到参数所指向的字符串存储空间中去,直到读到换行符 '\n' 结束,此换行符不作为字符串的内容存入串中。若无字符可读(即空串)则返回 NULL,否则返回存放字符串的起始地址。

提示:读入字符功能可调用 *getchar()* 函数实现。本题不允许通过调用系统提供的函数 *gets()* 实现。

函数定义:

编程要求(2):编写*strlen()*函数。

*udf_strlen()*功能为:统计参数所指字符串中字符的个数。

函数定义:



编程要求(3):编写`udf_strcmp()`函数。

`udf_strcmp()`功能为:对字符串 `s1` 和 `s2` 进行比较。如果 $s1 < s2$ 时,则返回负数;如果 $s1 = s2$ 时,则返回值等于 0;如果 $s1 > s2$ 时,则返回正数。

函数定义:

3. 有职工信息列表和已完成的 `main()` 函数如下所示:

```
no    salary
1003  3864.78
1002  3726.46
.....
1004  2965.32
1001  2567.89
0     0

#include<stdio.h>
#include<stdlib.h>
struct person {
    int    no;                //职工编号
    float salary;            //职工工资
    struct person * next;     //指向自身的结构指针
};

main(){
    struct person * head;     //定义头指针
    struct person * CreateListR(void); // CreateListR 函数原型说明
    void writListR(struct person * head); // writListR 函数原型说明

    head= CreateListR();     //调用尾部插入建链表函数构建链表
    writListR(head);         //将链表的职工信息写入 person_list.txt 文件
}
```

请编写 `CreateListR()` 和 `writListR(head)` 函数,以完善该程序的功能实现。具体要求参照相关答题处的编程要求(1)和编程要求(2)。

编程要求(1):编写采用表尾插法算法建立职工信息链表的函数 `CreateList()`。本函数编写请按算法描述和函数定义过程两步完成:

算法描述:

函数定义:

编程要求(2):编写将链表的职工信息写入 person_list.txt 文件的函数。

函数定义:

综合测试题一参考答案

一、填空题(表达式求值)设各语句的初始化相同: `int x=3,y=2,z=1;`

1. 0 2. 1 3. 0 4. 0 5. 15
6. 2 7. 3 8. 1 9. 1 10. 32

二、读下列程序段,指出错误并改正(改错题)

说明:指出错误的表示形式为错误处加下划线:如 `int a[n][n];`

改正形式为在错误语句之后加修改内容:如 `int a[4][4];`

- ① a 数组无 '\0', 改为: `char a[7] = "a1b2c3"`
或: `char a[7] = {'a', '1', 'b', '2', 'c', '3', '\0'}`
② 数组定义过小,不能存放串结束符 '\0' 至少应改为 `a[7], b[7]`。
- ① 函数应先定义后使用,否则事先声明,因此须加函数原型说明:
`double f1(double i, int j);`
② 形实参数的类型不匹配,因此须互换处理:
`m=f1(a, 3.14);` 或 `double f1(double i, int j);`
- 字符指针数组不能直接存放字符串修改方案有 2:
① 定义 `char s[5][80]`; 然后每行的起始地址赋给指针数组的相应元素:
`for(i=0; i<5; i++) str[i]=s[i];`
`for(i=0; i<5; i++) scanf("%s", str[i]);`
② 或直接定义二维数组 `char str[5][80]`; 然后:
`for(i=0; i<5; i++) printf("%s\n", str[i]);`
- ① 累加器应清零: `sum=0;`
② 数组越界使用应改为:
`for(i=0; i<10; i++) sum += r[i];`
- 结构体类型定义时不能赋初值,应改为:

```
struct student{
    int num;
    float score1, score2, aver;
} stu[3]={{200401,92,83},{200402,85,70},{200403,65,89}};
```



三、阅读分析下列程序,写出运行结果(阅读分析题)

1. 运行结果:

```
1 1 2 3
   5 8 13
```

2. 运行结果:

```
i=2,x=3
i=4,x=8
```

3. 运行结果:

```
6 8 10
```

4. 运行结果:

```
computer->retupmoc
```

5. 运行结果:

```
1,3
3,1
```

6. 运行结果:

```
a d
hi no
```

7. 运行结果:

```
how are you!
```

四、程序填空题

1. 本程序功能为建立链表及统计链表中所有节点的数据域 n 之和。

(1) head= NULL ;

(2) p1=p2=(struct node *)malloc(sizeof(struct node)));

(3) head= p1 ;

(4) p2->next= p1 ;

(5) p1=(struct node *)malloc(sizeof(struct node)));

(6) p2->next=NULL;

(7) return(head);

(8) add(struct node * head){

(9) sum=sum+p->n;

(10) p=p->next;

2. 已知正文有文件 abc.txt, 本题统计该文件中词的个数并求出最长词的长度。一个词与另一个词之间由空格符, 制表符, 换行符分隔。其中, word 存放词的个数, maxLen 存放最长词的长度, length 存放当前词的长度, flag 为状态标记(flag=1 表示当前字符在单词中, flag=0 表示当前字符不在某词中)。

(1) if((fp=fopen("abc.txt", "r"))==NULL){

(2) if(length>maxLen) maxLen=length;

(3) flag= 1 ; length=1;



(4) word ++ ;

(5) ch=fgetc(fp) ;

五、程序设计题

1. 根据主函数的定义,编写以下两个函数:

① 求数组 a[4][4]主对角线上元素的和;

② 实现将数组 a[4][4]的矩阵转置处理。

```
void main(){
    static int a[4][4]={1,1,1,1,2,2,2,2,3,3,3,3,4,4,4,4};
    int * p, i, sum;
    sum=total(a,4); printf("%d\n",sum); //调用 total 函数求主对角线上元素的和。
    p= * a; convert(p);                //调用 convert 实现矩阵转置
    for(i=0; i<4; i++)
        printf("%d %d %d %d\n",a[i][0],a[i][1],a[i][2],a[i][3]);
}
```

//参考程序清单如下:

```
#include <stdio. h>
```

```
void convert(int * p){
```

```
    int i, j, t;
```

```
    for(i=0; i<4; i++)
```

```
        for(j=i; j<4; j++){
```

```
            t= * (p+4 * i+j);
```

```
            * (p+4 * i+j)= * (p+4 * j+i);
```

```
            * (p+4 * j+i)=t;
```

```
        }
```

```
}
```

```
int total(int a[][4], int n){
```

```
    int i,sum=0;
```

```
    for(i=0; i<n; i++)
```

```
        if(a[i][i]!=0)
```

```
            sum+=a[i][i];
```

```
    return(sum);
```

```
}
```

```
void main(){
```

```
    static int a[4][4]={1,1,1,1,2,2,2,2,3,3,3,3,4,4,4,4};
```

```
    int * p, i, sum;
```

```
    p= * a;
```

```
    sum=total(a,4); printf("%d\n",sum);
```

```
    convert(p);
```



```
for(i=0; i<4; i++)
    printf("%d %d %d %d\n",a[i][0],a[i][1],a[i][2],a[i][3]);
}
```

(2) 编一个程序,用结构体数组描述 60 位学生的信息,包括:学号、姓名、三门课的成绩和总成绩。并以函数形式实现以下功能:

- ① 输入 60 位学生的前五项信息,并计算每个学生的总成绩。
- ② 按总成绩递减顺序对成绩表排序。
- ③ 输出排序后的成绩表。

注:类型定义和主函数。

//参考程序清单如下:

```
#define N60
#define STUDENT struct student
STUDENT{
    int num;
    char name[16];
    int score[4];
};
void main(){
    STUDENT stu[N];
    void read(STUDENT *p, int n);
    void sort(STUDENT s[], int n);
    void print(STUDENT *p, int n);
    read(stu,N); sort(stu,N); print(stu,N);
}
void read(STUDENT *p, int n){
    int i,j,s;
    for(i=0; i<n; i++,p++){
        scanf("%d%s",&p->num,p->name);
        for(s=j=0; j<3; j++){
            scanf("%d",&p->score[j]);
            s=s+p->score[j];
        }
        p->score[3]=s;
    }
}
void sort(STUDENT s[], int n){
    int i,j,k; STUDENT temp;
    for(i=0; i<n-1; i++){
        k=i;
```



```
        for(j=i+1; j<n; j++){
            if(s[k].score[3]<s[j].score[3]) k=j;
        }
        if(k!=i){
            temp=s[i]; s[i]=s[k]; s[k]=temp;
        }
    }
}

void print(STUDENT *p, int n){
    int i,j;
    for(i=0; i<n; i++,p++){
        printf("%6d%16s",p->num,p->name);
        for(j=0; j<4; j++)
            printf("%6d",p->score[j]);
        printf("\n");
    }
}
```

综合测试题二参考答案

一、选择题

1. A 2. B 3. D 4. A 5. C
6. B 7. C 8. C 9. B 10. D

二、填空题

1. fabs(cos(3.14159/2)) 或 a=3.14159, fabs(cos(a/2))
2. $3 * \text{pow}(x, n) / (2 * x - 1)$
3. 2.5
4. 1
5. 27
6. 1
7. 8
8. cc2 = (A&&E) || (A&&F) || (E&&F) ;//A,E,F3人中至少有2人参与作案;
cc3 = (A&&D) ;// A,D不可能是同案犯;
cc4 = (B&&C) || (! B&&! C) ;// B,C或同时作案,或与本案无关;
cc5 = C&&D ;// C,D中有且仅有一人作案;
cc6 = D || ! E ;//如果D没有参与作案,则E也不可能参与作案。



三、阅读程序题:请仔细阅读分析下列程序,写出运行结果

1. 运行结果:

```
3
3
26
58
```

2. 运行结果:

```
3 2
2 3
```

3. 运行结果:

```
1
2 4
3 6 9
```

4. 运行结果:

```
CountX=8
CountY=15
```

5. 运行结果:

```
AB
AB12
BCD
BCD234
```

四、程序填空题

1. 下面的程序把从终端读入的文本复制到一个名为 file1.dat 的新文件中。请在划线处填空。

- (1) ("file1.dat", "w"或"w+")
- (2) fputc(ch,fp) 或 fprintf(fp, "%c",ch);

2. 以下程序,数组 a 中存放一个递增数列。输入一个整数 x 并将它插入到数组 a 中,使之仍为一个递增数列。请在划线处填空。

- (1) 11 或比 11 大的数
- (2) break
- (3) a[i+1]=x 或 a[p]=x

五、程序设计题

编程实现以链表形式处理学生选修信息。每一学生选修课程的记录信息包括:学号 sno、课程号 cno、成绩 score,相应数据项如:"PB07210001","C0020101",95.0。请按要求编写各指定功能函数。(说明:本题已给出包含必要的头文件的预处理命令、结构体类型定义和放在程序末尾的主函数,供编程参照使用。)

- (1) 编写一个计算字符串长度的 str_len() 函数,要求为:函数的返回值为字符串长度。



//参考程序清单如下:

```
int str_len(char *p){ //求字符串长度
    int i=0;
    while(*p++!='\0')i++;
    return(i);
}
```

或数组形式编写。

(2) 编写一个字符串比较 str_cmp() 函数,可采用数组或指针方法编写;设有字符串 s 与 t,若 s 大于 t 则返回正值,s 等于 t 则返回 0,s 小于 t 则返回负值。

//参考程序清单如下:

```
intstr_cmp(char s[],char t[]){//数组形式
    int i=0;
    while(s[i]==t[i])
        if(s[i++]=='\0') return(0);
    return(s[i]-t[i]);
}
```

或

```
int str_cmp(char *s,char *t){//指针形式
    while(*s==*t&&*s!='\0'){s++;t++;}
    return(*s-*t);
}
```

(3) 编写一个建立链表的 create() 函数,要求为:建立结点个数不限,当输入记录的学号字符串长度小于等于 1 时结束,如输入:0 0 0 时结束;函数的返回值为指向链表 head 指针。

//参考程序清单如下:

```
struct student * create() { //建立链表函数,表尾插入算法
    struct student * head, * p1, * p2;
    head=NULL;
    p1=p2=(struct student *)malloc(sizeof(struct student));
    scanf("%s%s%f",&p1->sno,&p1->cno,&p1->score);
    while(str_len(p1->sno)>1){
        if(head==NULL)head=p1;
        else p2->next=p1;
        p2=p1;
        p1=(struct student *)malloc(sizeof(STRUCT STUDENT));
        scanf("%s%s%f",&p1->sno,&p1->cno,&p1->score);
    }
    p2->next=NULL; free(p1);
    return(head);
}
```



或采用表头插入方式、有序插入方式等编写。

(4) 编写一个按学号检索学生选修课程信息的 serch_print() 函数, 要求为: 能按指定学号检索该学生所有的选修课程信息, 即打印出所有与指定学号相关的结点的成员信息。

//参考程序清单如下:

```
void serch_print(struct student * head, char * search_sno){//查找打印信息
    STRUCT STUDENT * p; p=head;
    if(head != NULL)
        do{ if(str_cmp(p->sno, search_sno)==0)
            printf("%16s %12s %f\n", p->sno, p->cno, p->score);
            p=p->next;
        }while(p != NULL);
}
```

2. 如图 6.3 所示, 已知某游泳池的长度为 50 米, 宽度为 25 米。某人游泳速度为 0.8 米/秒, 步行速度为 1.2 米/秒。编程求解从 A 点到 B 点间何处下水游到 C 点时间最短, 输出该点距 A 点的长度 x 和所求最短时间 y 。要求:

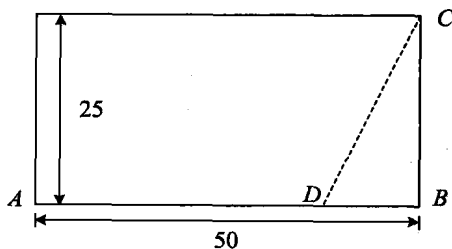


图 6.3 游泳池入水点问题示意图

(1) 给出求解该问题的算法描述。

(2) 根据算法编写完整的求解程序。

说明: 距离的长度精度控制在 0.1 米或以下。

(1) 求解问题的算法描述:

//参考程序清单如下:

本题采用枚举法(亦称穷举法)求解。

分析:

① 假设在 AB 边上行进 x_1 后的 D_1 点下水, 历时 $t_1 = x_1 / 1.2$;

② 计算从 D_1 点到 C 点的距离和 D_1 点到 C 点历时

$$t_2 = \sqrt{\text{WIDTH} * \text{WIDTH} + (\text{LENGTH} - x_1) * (\text{LENGTH} - x_1)} / 0.8$$

③ 则从 A 点到 B 点间的 D 点下水游到 C 点时间的为 $t = t_1 + t_2$, 显然, 从 A 点到 B 点间的 D_1 点下水游到 C 点最短计算如存在的话, 肯定在 0 到 50 之间, 而这之间 D_1 点在一定精度的步长内个数是有限的, 所以可以设计一个基于枚举法的算法, 设用 time 存储其最小值, dx 存储距 A 点的长度, 其最短时间算法描述如下:

预设 $x=0.0$, $\text{time} = 32768$, $\text{dx}=0$

```
while(x <= 50.0){
```

```
    temp = x / 1.2
```

```
    + sqrt(WIDTH * WIDTH + (LENGTH - x) * (LENGTH - x)) / 0.8;
```

```
    if(若所求时间小于等于前次所求最短时间){
```

```
        更新保存最短时间和相应距离两变量的值;
```

```
    }
```

```
    x = x + 0.1; //距离增步进长 0.1, 这里步进长精度为 0.1
```



}

(2) 根据算法编写完整的求解程序如下:

//参考程序清单如下:

```
#include<stdio. h>
#include<math. h>
#define LENGTH 50. 0
#define WIDTH 25. 0
#define V1 0. 8
#define V2 1. 2

void main(){
    //枚举法求解 1
    float x=0. 0,time,temp,ax=0;
    time =32768;
    while(x<=50. 0){
        temp=x/V2+sqrt(WIDTH * WIDTH+(LENGTH-x) * (LENGTH-x))/V1;
        if(temp<=time){
            time=temp;
            ax=x;
        }
        x=x+0. 1;
    }
    printf("time min=%f, x=%f\n",time,ax);
    printf("(50+25)/1. 2 =%f\n", (LENGTH+WIDTH)/V2);
}
```

注:此题解法甚多。

综合测试题三参考答案

一、选择题

1. C 2. A 3. C 4. B 5. A
6. C 7. D 8. B 9. C 10. B

二、填空题

1. 全局、局部
2. 5
3. 1,0
4. 0
5. 'a' 或 97
6. 0



7. 20

8. 6

三、阅读程序题(请仔细阅读分析下列程序,按程序实际执行情况写出运行结果)

1. 运行结果:

4

7

10

4

2. 运行结果:

17154

3. 运行结果:

1 1

9 1

17

4. 运行结果:

0 2 4 4 8

5. 运行结果:

image

mage

age

ami

6. 运行结果:

1, 3

3, 1

1, 3

7. 运行结果:

10 14

28 32

84

四、程序填空题

填空第1题参考答案:

(1) double a[5], int n

double a[5], int n

double *a, int n

(2) t * x

(3) y

填空第2题参考答案:



(4) $x[i][i]$

(5) $x[i-1][j-1]+x[i-1][j]$

填空第 3 题参考答案:

(6) `(struct list *)malloc(sizeof(struct list))`

(7) `(struct list *)malloc(sizeof(struct list))`

(8) `p->next`

(9) `NULL`

(10) `head`

(11) `a=b`

(12) `p=p->next`

(13) `b`

(14) `p=p->next`

(15) `p != NULL`

五、程序设计题

1. 建立一个 100 人的人口普查信息表,其中包括姓名、年龄、性别、职业、及地址,相应结构体类型定义如下所示:

```
struct person{
    char name[30];
    int age;
    char sex, job[30], addr[40];
};
```

要求在主函数中定义一个结构体数组,并采用形实参数结合的数据传递方式定义以下指定功能的用户自定义函数:

(1) 定义 `read` 函数,读入普查人员的五项数据。

(2) 定义 `udf_cmpstr` 字符串比较函数(注:功能必须由自己具体定义实现,不要直接或间接通过调用库函数实现)。

(3) 定义 `sort` 函数,按姓名以递升顺序排序。

(4) 定义 `print` 函数,将排序后的普查信息表输出到 `person_list` 文件长期保存。

(5) 定义 `binarysearch` 函数,以二分查找(亦称折半查找)方式实现按姓名查找该人员的在人口普查信息表的位置,若没有找到则返回-1,表示查找失败。

编写 `main` 函数,依次调用以上各功能函数,完成具体应用编程。其中调用 `binarysearch` 函数时,假设查找对象的姓名为“xiaoli”,返回主调函数后即打印输出该人员在信息表中的位置,若没有找到则提示“notfound!”信息。

提示:按指定要求编程,结构体数组不要采用全局的结构体数组形式。

```
#include<stdio.h>
#include<process.h>
#define N 100
#define PERSON struct person
```



```
struct person{
    char name[30];
    int age;
    char sex, job[30], addr[40];
};
```

程序设计参考答案:

(1) 定义 read 函数, 读入普查人员的五项数据;

```
void read(PERSON s[], int n){
    int i;
    for(i=0; i<n; i++){
        scanf("%s %d %c %s
%s", s[i]. name, &s[i]. age, &s[i]. sex, s[i]. job, s[i]. addr);
    }
}
```

(2) 定义 udf_cmpstr 字符串比较函数(注: 功能必须由自己具体定义实现, 不要直接或间接通过调用库函数实现);

```
int udf_cmpstr(char s[], char t[]){
    int i=0;
    while(s[i]==t[i] && s[i]!='\0')
        ++i;
    return(s[i]-t[i]);
}
```

(3) 定义 sort 函数, 按姓名以递升顺序排序;

```
void sort(PERSON s[], int n){
    int i, j, k; PERSON temp;
    for(i=0; i<n-1; i++){
        k=i;
        for(j=i+1; j<n; j++){
            if(udf_cmpstr(s[k]. name, s[j]. name)>0) k=j;
        }
        if(k!=i){
            temp=s[i]; s[i]=s[k]; s[k]=temp;
        }
    }
}
```



(4) 定义 print 函数,将排序后的普查信息表输出到 person_list 文件长期保存

```
void print(PERSON * p, int n){
    FILE * fp; int i;
    if((fp=fopen( "person_list", "w" ))==NULL){
        printf("cannot open this file. \n"); exit(0);
    }
    for(i=0; i<n; i++,p++){ //也可采用 fwrite 等形式完成
        fprintf(fp,"%s %d %c %s %s\n",
                p->name,p->age,p->sex,p->job,p->addr);
    }
    fclose(fp);
}
```

(5) 定义 binarysearch 函数,以二分查找(亦称折半查找)方式实现按姓名查找该人员在人口普查信息表的位置,若没有找到则返回-1,表示查找失败。

```
int binarysearch(PERSON s[],int n,char * key){
    int low, mid, high;
    low=0; high=n-1;
    while(low<=high){
        mid=(low+high)/2;
        if(udf_cmpstr(key,s[mid]. name)>0)
            low=mid+1;
        else if(udf_cmpstr(key,s[mid]. name)==0)
            return(mid);
        else
            high=mid-1;
    }
    return -1;
}
```

编写 main 函数,依次调用以上各功能函数,完成具体应用编程。其中调用 Binarysearch 函数时,假设查找对象的姓名为“xiaoli”,返回主函数后即打印输出该人员在信息表中的位置,若没有找到则提示“not found!”信息。

```
void main(){
    PERSON pe[N]; int i;
    read(pe,N);
```




```
sort(pe,N);
print(pe,N);
i=binarysearch(pe,N,"xiaoli");
if(i! =-1)
    printf("%d %s\n",i,pe[i]. name);
else
    printf("not found!");
}
```

综合测试题四参考答案

一、选择题(本大题共 10 小题,每小题 1 分,共 10 分)

1. A 2. C 3. B 4. D 5. B
6. A 7. B 8. C 9. C 10. D

二、填空题(本大题共 8 小题 12 个填空,每空 1 分,共 12 分)

1. 执行什么样的操作 或 执行的运算
2. 保留字
3. $T = 4 * t * t / (t + 2) - 20$ 或 $T = 4 * \text{pow}(t, 2) / (t + 2) - 20$
4. 1
5. 1
6. n 的值是 2, m 的值是 8, p 的值是 6
7. x 的值是 21, y 的值是 1, z 的值是 23
8. 如果 n 的值是 4 并且 m 的值是 5,表达式 $++(n * m)$ 的值是 21 吗?
解释你的答案:不是,表达式非法。递增运算不能用于 $(n * m)$ 这样的表达式。

三、阅读程序题:请仔细阅读分析下列程序,按程序实际执行情况,写出运行结果(本大题共 6 小题,第 1 题 5 分,其余各题 6 分,共 33 分)

1. 运行结果:

```
i=2,x=6
i=3,x=3
i=4,x=8
```

2. 运行结果:

```
x=3
x=6
x=14
```

3. 运行结果:

```
1 5 3 2 4
1 2 3 5 4
```



1 2 3 5 4

1 2 3 4 5

1 2 3 4 5

4. 运行结果:

6

12

18

18

5. 运行结果:

1 12 11 10

2 13 16 9

3 14 15 8

4 5 6 7

6. 运行结果:

x=10

y=11

x=20

y=22

四、程序设计题(共三个程序设计题,第1题12分,第2题15分,第3题18分,共45分)

1. 已知一组实验数据:3.62,2.93,3.16,3.73,2.86,3.40,2.86,3.07,3.29,3.24,编一程序分别求出它们的平均值、方差和均方差。

要求:

(1) 定义一个求解平均值、方差和均方差的函数 `udf_vft()`,与主调函数间的数据联系要求采用参数传递方式或 `return` 语句实现,不要采用全局变量传递方式。

(2) 一组实验数据可由数组方式在主函数中构成,并在主函数中通过调用 `udf_vft()` 函数得到所需要的平均值、方差和均方差并打印输出,每一结果只保留两位小数。

提示:

设平均值、方差和均方差分别用变量 v , f 和 t 表示,由数学知识可知,相应的计算公式为:

$$v = \frac{1}{n} \sum_{i=1}^n x_i \quad f = \frac{1}{n} \sum_{i=1}^n x_i^2 - v^2 \quad t = \sqrt{f}$$

其中, n 表示数据个数, x_i 表示第 i 个数据。

解: #include<stdio.h>

#include<math.h>

#define N 10

// N 等于待处理数据的个数

int main(){

double a[10]={3.62,2.93,3.16,3.73,2.86,3.40,2.86,3.07,3.29,3.24};

double v,f,t,s1=0,s2=0;



```
void udf_vft(double a[],int n,double * v,double * f,double * t);
udf_vft(a, N, &v, &f, &t);
printf("平均值 = %5.2f\n",v);
printf("方差 = %5.2f\n",f);
printf("均方差 = %5.2f\n",t);
}
```

```
void udf_vft(double a[],int n,double * v,double * f,double * t){
    double s1=0,s2=0;
    int i;
    for(i=0;i<N;i++){
        s1 += a[i];
        s2 += a[i] * a[i];
    }
    * v=s1/N;
    * f=s2/N;
    * t=sqrt(* f);
}
```

注:本题还可采用主函数定义数组、动态申请存储空间等方法由参数传递返回平均值、方差和均方差。但要注意本题不能使用局部变量或数组返回其值。

2. 编写一个程序,实现对一段文本中特定的某个单词进行统计的功能。

提示:通常文字处理软件(例如 MS Office Word 或者 WPS 等)中都提供了字符串查找匹配功能。一个程序要达到这样的功能首先需要具备有关字符串操作的一些函数来辅助实现这个功能。本题主函数编写已经完成,源程序代码清单如下,请编写与斜体字表示的函数调用对应的有关字符串操作函数,以实现这个功能。具体要求参照相关答题处。设主函数位于其他函数之后且内容如下:

```
#include <stdio.h>
#include <string.h>
int main(){
    int i=0,sum=0,length;
    char word[20];
    char text[]="to modernize the country's industry,agriculture,national \
defence and science and technology step by step to turn China into \
a strong and prosperous socialist country with a high level of \
culture and democracy. ";
    char temp[20]={0};
    printf("please input the word which you want to count. \n");
    udf_gets(word,20);//udf_gets(word)读字符串函数
    length=udf_strlen(word);// strlen(word)求串长函数
```



```
while(length! =0 && text[i]! ='\0'){
    strncpy(temp,text+i,length);    //逐一取定长的子串
    if(udf_strcmp(word,temp)=0){    //udf_strcmp(word,temp)字符串比较函数
        sum++;                      //匹配成功统计数加1
        i +=length;                 //下标 i 后移,为取下一个子串作准备
    }
    i++;
}
printf("The word \"%s\" appears %d times in the text.\n",word,sum);
}
```

编程要求(1):编写udf_gets()函数。

udf_gets()功能为:从标准输入文件 stdin 读入一串字符送到参数所指向的字符串存储空间中去,直到读到换行符'\n'结束,此换行符不作为字符串的内容存入串中。若无字符可读(即空串)则返回 NULL,否则返回存放字符串的起始地址。

提示:读入字符功能可调用 getchar()函数实现。本题不允许通过调用系统提供的函数 gets()实现。

函数定义:

```
char * udf_gets(char s[], int lim){
    int c, i;
    for(i=0;i<lim-1&&(c=getchar())!=EOF&&c!='\n';++i)
        s[i]=c;
    s[i]='\0';          //添加串结束符
    if(i! =0)
        return(s);
    else
        return(NULL); //返回串的长度 i
}
```

编程要求(2):编写strlen()函数。

udf_strlen()功能为:统计参数所指字符串中字符的个数。

函数定义:

```
int udf_strlen(char s[]){
    int i=0;
    while(s[i]! ='\0') i++;
    return(i);
}
```

编程要求(3):编写udf_strcmp()函数。

udf_strcmp()功能为:对字符串 s1 和 s2 进行比较。如果 s1<s2 时,则返回负数;如果



$s1=s2$ 时,则返回值等于 0;如果 $s1>s2$ 时,则返回正数。

函数定义:

```
int udf_strcmp( char s[ ], char t[ ] ){
    int i=0;
    while( s[i]==t[i] )
        if( s[i++]=='\0' )
            return( 0 );
    return( s[i] - t[i] );
}
```

注:本函数的功能实现方法略有可能不同,要求以功能实现正确、简洁为判分依据。

3. 有职工信息列表和已完成的 main() 函数如下所示:

```
no    salary.
1003  3864.78
1002  3726.46
.....
1004  2965.32
1001  2567.89
0     0

#include<stdio.h>
#include<stdlib.h>
struct person {
    int    no;                //职工编号
    float salary;            //职工工资
    struct person * next;    //指向自身的结构指针
};

main(){
    struct person * head;    //定义头指针
    struct person * CreateListR(void); // CreateListR 函数原型说明
    void writListR(struct person * head); // writListR 函数原型说明
    head= CreateListR();    //调用尾部插入建链表函数构建链表
    writListR(head);        //将链表的职工信息写入 person_list.txt 文件
}
```

请编写 CreateListR() 和 writListR(head) 函数,以完善该程序的功能实现。具体要求参照相关答题处的编程要求(1)和编程要求(2)。

编程要求(1):编写采用表尾插法算法建立职工信息链表的函数 CreateList()。本函数编写请按算法描述和函数定义过程两步完成:

算法描述:

```
head = NULL;    //置空链表
```



```
输入一个循环控制数据；
while(数据值不是结束标志) {
    申请新结点,用指针 p 指向该结点；
    给新结点赋予相关数据值；
    if(链表为空) {
        表示新插入的结点是首结点,
        头指针指向首结点 head = p;
    }
    else {
        将新结点插入到尾指针指向的结点之后, rear->next=p;
        使尾指针指向新结点, rear = p;
    }
    输入一个循环控制数据；
}
```

函数定义:

```
struct person * CreateListR(void) {
    struct person * head;          //头指针
    struct person * rear;         //尾指针
    struct person * p;           //新结点指针
    head=NULL;
    p=(struct person *)malloc(sizeof(struct person));    //申请新结点
    printf("\n 输入一个职工的工号和工资");
    scanf("%d%f",&p->no,&p->salary);
    while(p->no!=0){              //读入职工号不是结束标志(0)时做循环
        if(head==NULL)
            head=p;              //将 p 指向的新结点插入空表
        else
            rear->next=p;        //新结点插入到表尾结点(rear 指向的结点)之后
            rear=p;              //表尾指针指向新的表尾结点
            p=(struct person *)malloc(sizeof(struct person));    //申请新结点
            printf("\n 输入一个职工的工号和工资");
            scanf("%d%f",&p->no,&p->salary);
    }
    rear->next=NULL;              //终端结点置空
    free(p);
    return head;                 //返回表头指针
}
```

编程要求(2):编写将链表的职工信息写入 person_list.txt 文件的函数。



函数定义:

```
void writListR(struct person * head){
    FILE * fp;
    struct person * p ;
    if((fp=fopen("person_list.txt","w"))==NULL){
        printf("cannot open this file .\n");
        exit(0);
    }
    p=head;
    while(p!=NULL){
        fprintf(fp,"%d %f\n", p->no,p->salary);
        p=p->next;
    }
}
```

运行结果

输入:

```
1003 3864.78
1002 3726.46
1005 3567.89
1004 2965.32
1001 2567.89
0 0
```

person_list.txt 文件内容:

```
1003 3864.78
1002 3726.46
1005 3567.89
1004 2965.32
1001 2567.89
```