

## 第 3 部分 补充习题和模拟试卷

### 3.1 补充习题

#### 3.1.1 程序填空题

(1) 以下程序的功能是输入  $n$  ( $n$  是奇数), 计算  $1!+3!+\dots+n!$  的值。请填空使程序完整。

```
#include <stdio.h>
int main()
{
    long int f,s=0;
    int i,j,n;
    scanf("%d",&n);
    for(i=1;i<=n; ___[1]___ )
    {
        f=1;
        for(j=1; ___[2]___ ;j++)
            ___[3]___;
        s=s+f;
    }
    printf("n=%d,s=%ld\n",n,s);
    return 0;
}
```

答案:

[1]  $i+=2$  或  $i=i+2$  或  $i++,i++$

[2]  $j<=i$  或  $i>=j$  或  $j<i+1$  或  $i+1>j$

[3]  $f=f*j$

(2) 设已定义一个含有  $N$  个元素的数组  $s$ , 以下程序中, 函数  $\text{fun1}$  的功能是按顺序分别赋予各元素从 2 开始的偶数, 函数  $\text{fun2}$  则按顺序每  $M$  个元素求一个平均值, 并将该值存放在数组  $w$  中。请填空使程序完整。

```
#include <stdio.h>
#define N 30
#define M 5
void fun1(float s[])
{
    int k,i;
    for(k=2,i=0;i<N;i++)
    {
```

```

    _____ [1] _____ ;
    k+=2;
}
}
void fun2(float s[],float w[])
{
    float sum=0.0;
    int k,i;
    for(k=0,i=0;i<N;i++)
    {
        sum+=s[i];
        _____ [2] _____ ;
        {
            w[k]=sum/M;
            _____ [3] _____ ;
            k++;
        }
    }
}
int main()
{
    int i;
    float s[N],w[M];
    fun1(s);
    _____ [4] _____ ;
    for(i=0;i<N;i++)
    {
        if(i%5==0) printf("\n");
        printf("%8.2f",s[i]);
    }
    printf("\n");
    for(i=0;i<M;i++)
        printf("%8.2f",w[i]);
}

```

答案:

[1]  $s[i]=k$  或  $s[i]=(i+1)*2$

[2]  $if((i+1)\%M == 0)$ 或  $if((i+1)/M*M == i+1)$

[3]  $sum=0$  或  $sum=0.0$

[4]  $fun2(s,w)$

(3) 以下程序将一个字符串从下标为  $m$  的字符开始的全部字符复制后组成另一个字符串。请填空使程序完整。

```

#include <stdio.h>
#include <string.h>

```

```

void strcpy(char *str1,char *str2,int m)
{
    char *p1,*p2;
    [1] _____;
    p2=str2;
    while(*p1)
        [2] _____;
    [3] _____;
}
int main()
{
    int i,m;
    char str1[80],str2[80];
    gets(str1);
    scanf("%d",&m);
    [4] _____;
    puts(str1);
    puts(str2);
    return 0;
}

```

答案:

[1] p1=str1+m

[2] \*p2++=\*p1++ 或 \*(p2++)=\*(p1++) 或 \*p2=\*p1,p2++,p1++ 或 \*p2=\*p1++,p2++ 或 \*p2++=\*p1,p1++

[3] \*p2='\0'或\*p2=0 或\*p2=NULL

[4] strcpy(str1,str2,m)

(4)以下程序从键盘上输入一个字符串,将该字符串升序排列后输出到文件 test.txt 中,然后从该文件读出字符串并显示出来。请填空使程序完整。

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
int main()
{
    FILE *fp;
    char t,str[100],str1[100];
    int n,i,j;
    if((fp=fopen("test.txt","w"))==NULL)
    {
        printf("can't open this file.\n");
        exit(0);
    }
    printf("input a string:\n");
    gets(str);

```

```

    _____ [1] _____;
    for(i=0; _____ [2] _____; i++)
    for(j=0; j<n-i-1; j++)
        if(_____ [3] _____)
        {
            t=str[j];
            str[j]=str[j+1];
            str[j+1]=t;
        }
    _____ [4] _____;
    fclose(fp);
    fp=fopen("test.txt", "r");
    fgets(str1, 100, fp);
    printf("%s\n", str1);
    fclose(fp);
    return 0;
}

```

答案:

[1]  $n=\text{strlen}(\text{str})$  或  $\text{for}(n=0; \text{str}[n] \neq '\backslash 0'; n++)$  或  $\text{for}(n=0; \text{str}[n]; n++)$  或  $\text{for}(n=0; \text{str}[n] \neq 0; n++)$

[2]  $i < n$  或  $n > i$  或  $i < -1 + n$  或  $i < n - 1$  或  $n - 1 > i$  或  $-1 + n > i$  或  $i \leq n - 1$  或  $n - 1 \geq i$  或  $-1 + n \geq i$

[3]  $\text{str}[j] > \text{str}[j+1]$  或  $\text{str}[j+1] < \text{str}[j]$  或  $\text{str}[j+1] < \text{str}[j]$  或  $\text{str}[j+1] \leq \text{str}[j]$

[4]  $\text{fputs}(\text{str}, \text{fp})$  或  $\text{fprintf}(\text{fp}, \%s\backslash n, \text{str})$  或  $\text{fprintf}(\text{fp}, \%s, \text{str})$

(5) 以下程序的功能是: 输入  $n(n \leq 30)$ , 产生并输出如下  $n=7$  形式的方阵。请填空使程序完整。

```

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

```

```

#include <stdio.h>
#define N 30
int main()
{
    int a[N][N];
    int i, j, n;
    scanf("%d", &n);
    for (i=0; i<n; i++)
        for (j=0; j<n; j++)

```

```

        {
            if (____[1]____) a[i][j]=1;
            else if (i<j&& i+j<n-1) ____[2]____;
            else if (i>j&& i+j<n-1) a[i][j]=3;
            else if (____[3]____) a[i][j]=4;
            else a[i][j]=5;
        }
    for (i=0;i<n;i++)
    {
        for (j=0;j<n;j++)
            printf("%4d",a[i][j]);
        ____[4]____;
    }
    return 0;
}

```

答案:

[1]  $i == j || i + j == n - 1$

[2]  $a[i][j] = 2$

[3]  $i < j \ \&\& \ i + j > n - 1$  或  $j > i \ \&\& \ i + j > n - 1$  或  $i < j \ \&\& \ i + n - 1 < j$  或  $j > i \ \&\& \ i + n - 1 < j$

[4]  $\text{printf}("\n")$

(6) 以下程序输出 100~1000 范围内各位数字之和能被 15 整除的所有数, 输出时每 10 个一行。请填空使程序完整。

```

#include <stdio.h>
int main()
{
    int m,n,k,i=0;
    for(m=100;m<=1000;m++)
    {
        ____[1]____;
        n=m;
        do
        {
            k=k+____[2]____;
            n=n/10;
        }
        ____[3]____;
        if (k%15==0)
        {
            printf("%5d",m);
            i++;
            if(i%10==0) ____[4]____;
        }
    }
}

```

```

    return 0;
}

```

答案:

[1] k=0

[2] n%10 或 n-n/10\*10 或 n-10\*(n/10)

[3] while(n>0) 或 while(0<n) 或 while(n!=0) 或 while(0!=n)

[4] printf("\n")

(7) 以下程序从键盘上输入一个字符串和一个字符, 在字符串中删除与输入字符相同的字符。请填空使程序完整。

```

#include <stdio.h>
#define N 80
int main()
{
    char str[N],ch;
    int i,k=0;
    gets(____[1]____);
    ch=getchar();
    for(i=0;____[2]____;i++)
        if(str[i]!=ch)
        {
            ____[3]____;
            k++;
        }
        ____[4]____;
    puts(str);
    return 0;
}

```

答案:

[1] str

[2] str[i]!='\0' 或 str[i]!=NULL 或 str[i]!=0 或 str[i]

[3] str[k]=str[i] 或 \*(str+k)=\*(str+i) 或 str[k]=\*(str+i) 或 \*(str+k)=str[i]

[4] str[k]='\0' 或 \*(str+k]='\0' 或 str[k]=NULL 或 str[k]=0 或 \*(str+k)=0 或 \*(str+k)=

NULL

(8) 以下程序的功能是: 输入一个字符串, 删除该字符串中的所有数字字符。请填空使程序完整。

```

#include <stdio.h>
#define N 80
void delnum(char *s)
{
    int i,j;
    for(i=0,j=0; ____[1]____ '\0' ;i++)

```

```

        if(s[i]<'0' _____ [2] _____ s[i]>'9')
        {
            _____ [3] _____;
            j++;
        }
        s[j]='\0';
    }
int main()
{
    char item[N];
    printf("\n输入一个字符串:\n");
    gets(item);
    _____ [4] _____;
    printf("\n%s",item);
    return 0;
}

```

**答案:**

[1] s[i]!= 或 \*(s+i)!= 或 \*(i+s)!=

[2] ||

[3] s[j]=s[i] 或 \*(s+j)=\*(s+i) 或 s[j]=\*(s+i) 或 \*(s+j)=s[i]

[4] delnum(item)

(9) 以下程序的功能是：输入字符串，统计该字符串中的字母、数字、空格和其他字符的个数。请填空使程序完整。

```

#include <stdio.h>
_____ [1] _____;
int main()
{
    char s1[80];
    int a[4]={0};
    int k;
    gets(s1);
    _____ [2] _____;
    puts(s1);
    for(k=0;k<4;k++)
        printf("%4d",a[k]);
    return 0;
}
void fun(char s[],int b[])
{
    int i;
    for (i=0;s[i]!='\0';i++)
        if ('a'<=s[i]&& s[i]<='z' || 'A'<=s[i]&& s[i]<='Z')
            b[0]++;
}

```

```

else if (____ [3] ____ )
    b[1]++;
else if (____ [4] ____ )
    b[2]++;
else
    b[3]++;
}

```

答案:

[1] void fun(char s[],int b[])或 void fun(char [],int [])

[2] fun(s1,a)

[3] '0'<=s[i] && s[i]<='9' 或 s[i]>='0' && s[i]<='9' 或 '0'<=s[i] && '9'>=s[i] 或 s[i]>='0' && '9'>=s[i] 或 48<=s[i] && s[i]<=57 或 s[i]>=48 && s[i]<=57 或 48<=s[i] && 57>=s[i] 或 s[i]>=48 && 57>=s[i] 或 !(x < 48 || x > 57) 或 !(x < '0' || x > '9')

[4] s[i] == ' ' 或 s[i] == 32

(10) 设数组中的元素已按递增排序，以下程序采用二分法查找 key 值，若找到 key 值，则返回对应的下标，否则返回-1。请填空使程序完整。

```

#include <stdio.h>
#define N 10
int fun(int a[],int n,int key)
{
    int low,high,mid;
    low=0;
    high=n-1;
    while(____ [1] ____ )
    {
        mid=(low+high)/2;
        if(key<a[mid])
            ____ [2] ____ ;
        else if(key>a[mid])
            ____ [3] ____ ;
        else
            ____ [4] ____ ;
    }
    return -1;
}
int main()
{
    int a[N]={1,2,3,4,5,6,7,8,9,10};
    int b,c;
    scanf("%d",&b);
    c=fun(a,N,b);
    if(c==-1)

```



```

    printf("not found");
else
    printf("position %d\n",c);
return 0;
}

```

答案:

[1]  $low \leq high$  或  $high \geq low$

[2]  $high = mid - 1$

[3]  $low = mid + 1$

[4] `return mid` 或 `return (mid)`

(11) 以下程序计算并输出  $high$  以内最大的 10 个素数之和,  $high$  由主函数传给 `fun` 函数, 若  $high$  的值为 100, 则 `fun` 函数的值为 732。请填空使程序完整。

```

#include <stdio.h>
#include <math.h>
int fun(int high)
{
    int sum = 0, n=0, j, yes;
    while ((high >= 2) && (____ [1] ____))
    {
        yes = 1;
        for (j=2; j<=high/2; j++)
            if (____ [2] ____ )
            {
                yes=0;
                break;
            }
        if (yes)
        {
            sum +=high;
            n++;
        }
        high--;
    }
    ____ [3] ____;
}
int main()
{
    int n;
    scanf("%d",&n);
    printf("%d\n", fun (n));
    return 0;
}

```

答案:

[1]  $n < 10$  或  $10 > n$

[2]  $high \% j == 0$  或  $!(high \% j)$

[3] `return sum` 或 `return (sum)`

(12) 以下程序将 `s` 所指字符串的正序和反序进行连接, 形成一个新串放在 `t` 所指的数组中。例如, 当 `s` 串为 `ABCD` 时, 则 `t` 串的内容应为 `ABCDDCBA`。请填空使程序完整。

```
#include <stdio.h>
#include <string.h>
void fun (char *s, char *t)
{
    int i, d;
    d = ____ [1] ____;
    for (i = 0; i < d; ____ [2] ____ )
        t[i] = s[i];
    for (i = 0; i < d; i++)
        t[____ [3] ____] = s[d-1-i];
    t[____ [4] ____] = '\0';
}
int main()
{
    char s[100], t[100];
    printf("\nPlease enter string S:");
    scanf("%s", s);
    fun(s, t);
    printf("\nThe result is: %s\n", t);
    return 0;
}
```

答案:

[1] `strlen(s)`

[2] `i++` 或 `i=i+1` 或 `i+=1` 或 `++i`

[3] `d+i` 或 `i+d`

[4] `2*d` 或 `d*2` 或 `i+d` 或 `d+i`

(13) 以下程序的功能是: 输入 `N` 个数, 将最大的与最后一个数交换, 最小的与第一个数交换, 然后输出这些数。请填空使程序完整。

```
#include <stdio.h>
#define N 10
void input(int [],int);
void max_min(int [],n);
void output(int [],int);
int main()
{
    int n,number[N];
```

```

scanf("%d",&n);
input(number,n);
max_min(number,n);
output(number,n);
return 0;
}
void input(int number[],int n)
{
int i;
for(i=0;____[1]____;i++)
scanf("%d",&number[i]);
}
void max_min(int array[],int n)
{
int *max,*min,k,l;
int *p,*arr_end;
arr_end=array+n;
max=min=array;
for(p=array+1;p<arr_end;p++)
{
if(*p>*max)
max=p;
if(*p<*min)
____[2]____;
}
k=*max;
l=*min;
*p=array[0];
array[0]=1;
____[3]____;
*p=array[n-1];
____[4]____;
k=*p;
}
void output(int array[],int n)
{
int *p;
for(p=array;p<array+n-1;p++)
printf("%d",*p);
printf("\n");
}

```

答案:

[1]  $i < n$  或  $n > i$

[2]  $min=p$

[3] l=\*p

[4] array[n-1]=k

(14) 两个乒乓球队进行比赛，各出3人。甲队为a、b、c，乙队为x、y、z。已抽签决定了比赛名单。有人向队员打听比赛的名单。a说他不和x比，c说他不和x、z比。以下程序用来找出各选手的对阵名单，请填空使程序完整。

```
#include <stdio.h>
int main()
{
    char i,j,k;/*i 是 a 的对手, j 是 b 的对手, k 是 c 的对手*/
    for(i='x';i<='z';i++)
        for(j='x';j<='z';j++)
        {
            if(____ [1] ____ )
                for(k='x';____ [2] ____ ;k++)
                {
                    if(____ [3] ____ )
                    {
                        if(i!='x'&&k!=____ [4] ____ )
                            printf("比赛顺序: a--%c\tb--%c\tc--%c\n",i,j,k);
                    }
                }
        }
    return 0;
}
```

答案:

[1] i!=j

[2] k<='z' 或 'z'>=k

[3] i!=k&&j!=k

[4] 'z'

(15) 以下程序的功能是：从键盘输入一个字符串，将小写字母全部转换成大写字母，然后输出到一个磁盘文件 test 中保存。输入的字符串以#结束。请填空使程序完整。

```
#include <stdlib.h>
int main()
{
    FILE *fp;
    char str[100];
    int i=0;
    if((fp=fopen("test",____ [1] ____))==NULL)
    {
        printf("cannot open the file\n");
        exit(0);
    }
}
```

```

printf("please input a string:\n");
gets(____[2]____);
while(str[i]!='#')
{
    if(str[i]>='a'&&____[3]____)
        str[i]=str[i]-32;
    fputc(str[i],fp);
    i++;
}
fclose(____[4]____);
fp=fopen("test","r");
fgets(str,strlen(str)+1,fp);
printf("%s\n",str);
fclose(fp);
return 0;
}

```

答案:

[1] "w"

[2] str

[3] str[i]<='z' 或 'z'>=str[i]

[4] fp

(16) 有 N 个学生, 每个学生有 M 门课的成绩。以下程序的功能是: 从键盘输入以上数据 (包括学生号、姓名、3 门课成绩), 计算出平均成绩。设原有的数据和计算出的平均分存放在磁盘文件 stud 中。请填空使程序完整。

```

#include <stdio.h>
#define N 5
#define M 3
struct student
{
    char num[6];
    char name[8];
    int score[M];
    float avr;
} stu[N];
int main()
{
    int i,j;
    float sum;
    FILE *fp;
    for(i=0;i<N;i++)
    {
        printf("\n 请输入 No. %d 的成绩:\n",i);
        printf("stuNo:");
    }
}

```

```

scanf("%s",stu[i].num);
printf("name:");
scanf("%s",stu[i].name);
sum=0;
for(j=0;____[1]____;j++)
{
    printf("score %d.",j+1);
    scanf("%d",&stu[i].score[j]);
    sum+=stu[i].____[2]____;
}
stu[i].avr=sum/M;
}
fp=fopen("stud","w");
for(i=0;i<N;____[3]____)
    if(fwrite(&stu[i],sizeof(____[4]____),1,fp)!=1)
        printf("文件写错误! \n");
fclose(fp);
return 0;
}

```

答案:

[1]  $j < M$  或  $M > j$

[2] score[j]

[3]  $i++$  或  $++i$  或  $i=i+1$

[4] struct student

(17) 有  $n$  个人围成一圈, 顺序排号。从第一个人开始报数 (从 1 到  $m$  报数), 凡报到  $m$  的人退出圈子。以下程序可以求出最后留下的是原来第几号的人。请填空使程序完整。

```

#include <stdio.h>
#define nmax 50
int main()
{
    int i,j,k,m,n,num[nmax],*p;
    printf("请输入总人数和间隔数:");
    scanf("%d %d",&n,&m);
    p=num;
    for(i=0;____[1]____;i++)
        *(p+i)=____[2]____;
    i=0;
    k=0;
    j=0;
    while(j<n-1)
    {
        if(____[3]____!=0) k++;
        if(k==m)

```

```

    {
        *(p+i)=0;
        k=0;
        j++;
    }
    i++;
    if(i==n) i=0;
}
while(____[4]____) p++;
printf("%d 被留下! \n",*p);
return 0;
}

```

答案:

[1]  $i < n$  或  $n > i$

[2]  $i+1$  或  $1+i$

[3]  $*(p+i)$  或  $*(i+p)$  或  $p[i]$

[4]  $*p==0$  或  $0==*p$

(18) 以下程序的功能是：将长整型数中每一位上为奇数的数依次取出，构成一个新数放在 t 中。高位仍在高位，低位仍在低位。请填空使程序完整。

```

#include <conio.h>
#include <stdio.h>
void fun (long s, long *t)
{
    int d;
    long sl=1;
    ____ [1] ____ = 0;
    while (s > 0)
    {
        d = ____ [2] ____;
        if(d%2)
        {
            **t = ____ [3] ____ + *t;
            ____ [4] ____;
        }
        s /= 10;
    }
}
int main()
{
    long s, t;
    printf("\n 请输入一个正整数:");
    scanf("%ld", &s);
    fun(s, &t);
}

```

```

printf("结果是: %ld\n", t);
return 0;
}

```

答案:

[1] \*t

[2] s%10

[3] d \* s1 或 s1 \* d

[4] s1\*=10 或 s1=s1\*10

(19) 以下程序的功能是将一个英文句子中出现的某人的名字替换成指定的另一个人的名字。规定: 名字是一个单独的单词, 而不是一个单词的一部分, 也不由多个单词构成。例如, 原来的英文句子为“Bob is a good boy.We love Bob.”, 要求将 Bob 替换成 John, 则替换后的句子为“John is a good boy.We love John.”。请填空使程序完整。

```

#include <stdio.h>
#include <string.h>
#include <ctype.h>
void fun(char *str,char *orgname,char *newname)
{
    int i,j,k;
    char buf[512];
    for( [1] ;str[i]!='\0';i++)
    {
        for(j=i,[2] ;str[j]==orgname[k];j++,k++);
        if(orgname[k]!='\0'&&(i==0||!isalpha(str[i-1]))&&!isalpha
            (str[j]))
        {
            strcpy(buf,str+j);
            strcpy(str+i,newname);
            strcpy(str+i+strlen(newname), [3] );
        }
    }
}
int main()
{
    char s[512]="Bob is a good boy.We love Bob.";
    char *olds="Bob",*news="John";
    puts(s);
    fun([4]);
    puts(s);
    return 0;
}

```

答案:

[1] i=0



[2] k=0

[3] buf

[4] s,olds,news

(20) 以下程序的功能是：函数 `STU *insert(STU *head,STU *p)` 是将 `p` 指向的节点作为首节点插入 `head` 指向的链表中；`main` 函数接收从键盘输入的一行字符，每接收一个整数后，申请一个新节点保存该整数，并调用 `insert` 函数将新节点插入到链表的表头，最后从表头开始依次输出该链表各节点成员 `c` 的值。请填空使程序完整。

```
#include <stdio.h>
#include <stdlib.h>
typedef struct node
{
    int c;
    struct node *next;
}STU;
STU *insert(STU *head,STU *p);
int main()
{
    char x;
    STU *head=NULL,**p;
    while((ch=getchar())!='\n')
    {
        p=(STU *)malloc(sizeof(STU));
        [1] =ch;
        p->next=NULL;
        [2] ;
    }
    p=head;
    while(p)
    {
        printf("%c",p->c);
        [3] ;
    }
    return 0;
}
STU *insert(STU *head,STU *p)
{
    if(head==NULL)
        head=p;
    else
    {
        [4] ;
        head=p;
    }
}
```

```

    return head;
}

```

答案:

[1] p->c

[2] head=insert(head,p)

[3] p=p->next

[4] p->next=head

### 3.1.2 程序改错题

(1) 输入正整数  $n$  ( $n \geq 3$ ), 计算  $s = 1 + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} + \dots + \frac{1}{2n-3}$  的值。

```

#include <stdio.h>
double fun(int m)
{
    /*****[1]*****/
    double s=1
    int i;
    /*****[2]*****/
    for(i=3; i<m; i++)
        /*****[3]*****/
        s+=1.0/(2i-3);
    return(s);
}
int main()
{
    int n;
    printf("Enter n:");
    scanf("%d", &n);
    printf("\n结果是: %1f\n", fun(n));
    return 0;
}

```

答案:

[1] double s=1;

[2] for(i=3; i<=m; i++) 或 for(i=3; i<m+1; i++)

[3] s+=1.0/(2\*i-3);

(2) 从键盘接收一个字符串, 然后按照字符顺序从小到大进行排序, 并删除重复的字符。

```

#include <stdio.h>
#include <string.h>
int main()

```

```

{
    char str[100], *p, *q, *r, c;
    printf("输入字符串:");
    gets(str);
    /***** [1] *****/
    for(p=str;p;p++)
    {
        for(q=r=p;*q;q++)
            if(*r>*q)
                r=q;
        /***** [2] *****/
        if(r==p)
        {
            /***** [3] *****/
            c=r;
            *r=*p;
            *p=c;
        }
    }
    for(p=str;*p;p++)
    {
        for(q=p;*p==*q;q++);
        strcpy(p+1,q);
    }
    printf("结果字符串: %s\n",str);
    return 0;
}

```

答案:

[1] for(p=str;\*p;p++)

[2] if(r!=p)

[3] c=\*r;

(3) 读入一行英文文本，将其中每个单词的最后一个字母改成大写，然后输出此文本行（这里的“单词”是指由空格隔开的字符串）。例如，若输入“I am a student to take the examination.”，则应输出“I aM A studenT tO takE thE examinatioN.”。

```

#include <conio.h>
#include <stdio.h>
#include <ctype.h>
#include <string.h>
void fun(char *p)
{
    /***** [1] *****/
    integer k=0;

```

```

for( ;*p; p++ )
    if (k)
    {
        /*****[2]*****/
        if (*p==' ')
        {
            k=0;
            /*****[3]*****/
            *(p-1) = toupper(*(p-1));
        }
    }
    else if(*p !=' ')k=1;
    (p-1)=toupper(p-1);
}
int main()
{
    char str[81];
    printf("\n 请输入一个字符串: " );
    gets (str);
    printf("\n\n 变换之前: %s \n ", str);
    fun(str);
    printf("\n 变换之后: %s \n ", str);
    return 0;
}

```

**答案:**

[1] int k=0;

[2] if(\*p=='')

[3] \*(p-1)=toupper(\*(p-1));

(4) 输入 n 个整数,找出其中最大的数及其下标。

```

#include <stdio.h>
#define N 10
/*****[1]*****/
float fun(int *a,int *b,int n)
{
    int *c,max=*a;
    for(c=a+1;c<a+n;c++)
        if(*c>max)
        {
            max=*c;
            /*****[2]*****/
            b=c-a;
        }
}

```

```

    }
    return max;
}
int main()
{
    int a[N],i,max,p=0;
    printf("请输入 N 个整数:\n");
    for(i=0;i<N;i++)
        /***** [3] *****/
        scanf("%d",a[i]);
    /***** [4] *****/
    m=fun(a,p,N);
    printf("max=%d,position=%d",max,p);
    return 0;
}

```

答案:

[1] int fun(int \*a,int \*b,int n)

[2] \*b=c-a;

[3] scanf("%d",&a[i]); 或 scanf("%d",a+i);

[4] max=fun(a,&p,N);

(5) 求出 0~7 这 8 个数字所能组成的奇数个数。

```

#include <stdio.h>
int main()
{
    /***** [1] *****/
    long sum=4;s=4;
    int j;
    for(j=2;j<=8;j++)
    {
        /***** [2] *****/
        printf("\n%d",sum);
        /***** [3] *****/
        if(j<2)
            s*=7;
        else
            s*=8;
        sum+=s;
    }
    printf("\nsum=%ld",sum);
    return 0;
}

```

答案:

[1] long sum=4,s=4;

[2] printf("\n%d",sum);

[3] if(j<=2) 或 if(j<3)

(6) 将 a 所指字符串中的字符和 b 所指字符串中的字符按排列的顺序交叉合并到 c 所指数组中, 过长的剩余字符接在 c 所指数组的尾部。例如, 当 a 所指字符串中的内容为 abcdefg, b 所指字符串中的内容为 1234 时, c 所指数组中的内容应该为 a1b2c3d4efg; 而当 a 所指字符串中的内容为 1234, b 所指字符串中的内容为 abcdefg 时, c 所指数组中的内容应该为 1a2b3c4defg。

```
#include <conio.h>
#include <stdio.h>
#include <string.h>
/*****[1]*****/
fun(char a, char b, char c)
{
    while (*a && *b)
    {
        *c = *a;
        c++;
        a++;
        *c = *b; c++; b++;
    }
    if(*a == '\0')
        /*****[2]*****/
        while(*b) *c = *b; c++; b++; }
    else
        /*****[3]*****/
        while(*a) *c = *a; c++; a++; }
    *c = '\0';
}
int main()
{
    char s1[100], s2[100], t[200];
    printf("\nEnter s1 string : ");
    scanf("%s",s1);
    printf("\nEnter s2 string : ");
    scanf("%s",s2);
    fun( s1, s2, t );
    printf("\nThe result is : %s\n", t);
    return 0;
}
```

答案:

[1] void fun( char \*a, char \*b, char \*c )

[2] while( \*b ) { \*c = \*b; c++; b++; }

[3] while( \*a ) { \*c = \*a; c++; a++; }

(7) 求广义斐波那契级数的第  $n$  项。广义斐波那契级数的前  $n$  项为 1, 1, 1, 3, 5, 9, 17, 31, ..., 项值通过函数值返回 main 函数。例如, 若  $n=15$ , 则应输出“The value is: 2209”。

```
#include <conio.h>
#include <stdio.h>
long fun(int n)
{
    long a=1, b=1, c=1, d=1, k;
    /*****[1]*****/
    for (k=4; k<n; k++)
    {
        d = a+b+c;
        /*****[2]*****/
        a=b;
        b=c;
        c=d;
    }
    /*****[3]*****/
    return k;
}
int main()
{
    int n;
    scanf("%d",&n);
    printf("The value is: %ld\n", fun(n));
    return 0;
}
```

答案:

[1] for (k=4; k<=n; k++)

[2] a=b;

[3] return d;

(8) 编写一个函数, 该函数可以统计一个长度为 3 的字符串(子字符串)在另一个字符串(主字符串)中出现的次数。例如, 假定主字符串为 asdasdfgasdaszx67asdmklo, 子字符串为 asd, 则应输出  $n=4$ 。

```
#include <stdio.h>
#include<string.h>
int fun(char *str,char *substr)
{
    /*****[1]*****/
```

```

int i,n=0
/*****[2]*****/
for(i=0;i<=strlen(str);i++)
    if((str[i]==substr[0])&&(str[i+1]==substr[1])&&(str[i+2]==substr[2]))
        /*****[3]*****/
        ++i;
return n;
}
int main()
{
    char str[81],substr[4];
    int n;
    printf("输入主字符串: ");
    gets(str);
    printf("输入子字符串: ");
    gets(substr);
    puts(str);
    puts(substr);
    n=fun(str,substr);
    printf("n=%d\n",n);
    return 0;
}

```

答案:

[1] int i,n=0;

[2] for(i=0;i<=strlen(str)-3;i++) 或 for(i=0;i<strlen(str)-2;i++)

[3] n++; 或 n+=1; 或 n=n+1; 或 ++n;

(9) 输入一个字符串,过滤此串中的字母,并统计新生成的串中包含的字符个数。例如,输入的字符串为 ab234\$df4,则输出为

```

The new string is 234$4
There are 5 char in the new string.

```

```

#include <stdio.h>
#define N 80
int fun(char *ptr)
{
    int i,j;
    /*****[1]*****/
    for(i=0,j=0;*(ptr+i)!="\0";i++)
        /*****[2]*****/
        if(*(ptr+i)>'z' || *(ptr+i)<'a' || *(ptr+i)>'Z' || *(ptr+i)<'A')
        {
            /*****[3]*****/

```



```

        (ptr+j)=(ptr+i);
        j++;
    }
    *(ptr+j)='\0';
    return(j);
}
int main()
{
    char str[N];
    int s;
    printf("input a string:");
    gets(str);
    printf("The original string is :");
    puts(str);
    s=fun(str);
    printf("The new string is :");
    puts(str);
    printf("There are %d char in the new string.\n",s);
    return 0;
}

```

答案:

[1] for(i=0,j=0;\*(ptr+i)!='\0';i++)

[2] if( \*(ptr+i)>'z' || \*(ptr+i)<'a'&&\*(ptr+i)>'Z' || \*(ptr+i)<'A' )

[3] \*(ptr+j)=\*(ptr+i);

(10) 求出在字符串中最后一次出现的子字符串的地址，通过函数值返回，在主函数中输出从此地址开始的字符串；若未找到，则函数值为 NULL。例如，当字符串中的内容为 abcdabfabcdx，t 中的内容为 ab 时，输出结果应是 abcdx；当字符串中的内容为 abcdabfabcdx，t 中的内容为 abd 时，则程序输出未找到信息 “Not found, ERROR!”。

```

#include <stdio.h>
#include <string.h>
char * fun(char *s, char *t)
{
    char *p, *r, *a;
    /*****[1]*****/
    a == NULL;
    while (*s)
    {
        p = s;
        r = t;
        while (*r)
            /*****[2]*****/
            if (r == p)
            {

```

```

        r++;
        p++;
    }
    else
        break;
    /*****[3]*****/
    if (*r == '\0') a = s;
    s++;
}
return a;
}
int main()
{
    char s[100], t[100], *p;
    printf("\nPlease enter string S:");
    scanf("%s", s);
    printf("\nPlease enter substring t:");
    scanf("%s", t);
    p = fun(s, t);
    if (p)
        printf("\nThe result is : %s\n", p);
    else
        printf("\nNot found, ERROR !\n");
    return 0;
}

```

**答案:**

- [1] a = NULL;
- [2] if(\*r == \*p) { r++; p++; }
- [3] if(\*r == '\0') a = s;

(11) 在字符串 `str` 中找出 ASCII 码值最小的字符, 将其放在第一个位置上, 并将该字符前的原字符向后顺序移动。例如, 调用 `fun` 函数之前输入字符串 `eBCDAFGH`, 调用后字符串变为 `AeBCDFGH`。

```

#include <stdio.h>
#include <string.h>
void fun(char *p)
{
    char min,*q=p;int i=0;
    min=p[i];
    while(p[i]!=0)
    {
        if(min>p[i])
        {
            min=p[i];

```

```

        /*****[1]*****/
        p=p+i;
    }
    i++;
}
/*****[2]*****/
while(q<p)
{
    /*****[3]*****/
    *q=(q-1);
    q--;
}
p[0]=min;
}
int main()
{
    char str[80];
    printf("Enter a string:");
    gets(str);
    printf("\nThe original string:");
    puts(str);
    fun(str);
    printf("\nThe string after moving:");
    puts(str);
    return 0;
}

```

答案:

[1] q=p+i;

[2] while(q>p)

[3] \*q=(q-1);

(12) 将 s 所指字符串中最后一次出现的 t1 所指子串替换成 t2 所指子串, 形成的新串放在 w 所指的数组中。在此处, 要求 t1 和 t2 所指字符串的长度相同。例如, 当 s 所指字符串中的内容为 abcdabfabc, t1 所指子串中的内容为 ab, t2 所指子串中的内容为 99 时, 在 w 所指的数组中的内容应为 abcdabf99c。

```

#include <stdio.h>
#include <string.h>
int fun(char *s, char *t1, char *t2 , char *w)
{
    int i;
    char *p, *r, *a;
    strcpy(w,s);
    /*****[1]*****/
    while (w)

```

```

{
    p = w;
    r = t1;
    while (*r)
        /*****[2]*****/
        if (*r != *p)
        {
            r++;
            p++;
        }
        else
            break;
        /*****[3]*****/
        if (*r == '\0') a = w;
        w++;
    }
    r = t2;
    while (*r)
    {
        *a = *r;
        a++;
        r++;
    }
}
int main()
{
    char s[100], t1[100], t2[100], w[100];
    printf("\n输入字符串:");
    scanf("%s", s);
    printf("\n输入第一个子字符串:");
    scanf("%s", t1);
    printf("\n输入第二个子字符串:");
    scanf("%s", t2);
    if (strlen(t1) == strlen(t2))
    {
        fun(s, t1, t2, w);
        printf("\n结果是: %s\n", w);
    }
    else
        printf("\n错误 : strlen(t1) != strlen(t2)\n");
    return 0;
}

```

答案:

[1] while (\*w)

[2] if(\*r == \*p)

[3] if(\*r == '\0') a = w;

(13) 在一个已按升序排列的数组中插入一个数，插入后，数组元素仍按升序排列。

```
#include <stdio.h>
#define N 11
int main()
{
    int i,j,t,number,a[N]={1,2,4,6,8,9,12,15,149,156};
    printf("请输入一个整数: \n");
    /***** [1] *****/
    scanf("%d",number);
    printf("插入之前:\n");
    for(i=0;i<N-1;i++)
        printf("%5d",a[i]);
    printf("\n");
    /***** [2] *****/
    for(i=N-1;i>=0;i--)
        if(number<=a[i])
            /***** [3] *****/
            a[i]=a[i-1];
    else
    {
        a[i+1]=number;
        /***** [4] *****/
        exit;
    }
    if(number<a[0]) a[0]=number;
    printf("插入之后:\n");
    for(i=0;i<N;i++)
        printf("%5d",a[i]);
    printf("\n");
    return 0;
}
```

答案:

[1] scanf("%d",&number);

[2] for(i=N-2;i>=0;i--) 或 for(i=N-2;0<=i;i--)

[3] a[i+1]=a[i];

[4] break;

(14) 按以下要求移动一维数组中的内容：若数组中有  $n$  个整数，要求把下标从 0 到  $p$  (含  $p$ ,  $p$  小于或等于  $n-1$ ) 的数组元素平移到数组的最后。例如，一维数组中的原始内容为“1,2,3,4,5,6,7,8,9,10”， $p$  的值为 3，移动后，一维数组中的内容应为“5,6,7,8,9,10,1,2,3,4”。

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

```

#define N 80
void fun(int *w, int p, int n)
{
    /*****[1]*****/
    int i,k=0,b[n];
    /*****[2]*****/
    for(i=1; i < n; i++) b[k++]=w[i];
    /*****[3]*****/
    for(i=0; i <= p; i++) b[k]=w[i];
    for(i=0; i < n; i++) w[i]=b[i];
}
int main()
{
    int a[N]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};
    int i,p,n=15;
    printf("原来的数据:\n");
    for(i=0;i<n;i++)
        printf("%d",a[i]);
    printf("移动步长:");
    scanf("%d",&p);
    /*****[4]*****/
    fun(a[N],p,n);
    printf("\n 移动后的结果: \n");
    for(i=0; i < n; i++)
        printf("%d ",a[i]);
    printf("\n\n");
    return 0;
}

```

答案:

```

[1] int i,k=0,b[N];
[2] for(i=p+1; i < n; i++) b[k++]=w[i];
[3] for(i=0; i <= p; i++) b[k++]=w[i];
[4] fun(a,p,n);

```

(15) `string` 数组内存放了若干字符串，以下程序从中找出最长的字符串（如果有多个长度相同且为最长的字符串，则约定第一个为最长的字符串），并在该字符串中的每个数字字符前插入一个空格。例如，如果 `string` 数组中的字符串为 A1B23CD45、EF2G34、ER45DF678、985GHJ211FGH，则最长的字符串为 985GHJ211FGH，插入空格后该字符串变为“ 9 8 5GHJ 2 1 1FGH”。

函数 `maxstr` 的功能是在存放于 `ss` 数组中的 `m` 个字符串中查找最长的字符串，函数返回最长字符串所在数组行的行下标。

函数 `insert` 的功能是在 `str` 数组中存放的字符串内每个数字字符前插入一个空格。

测试数据: "A1B23CD45","EF2G34","ER45DF678","985GHJ211FGH"。

屏幕输出:

The maxstring: 985GHJ211FGH  
 The changedstring: 9 8 5GHJ 2 1 1 FG H

程序如下:

```

#include <ctype.h>
#include <string.h>
#include <stdio.h>
#include <conio.h>
#define M 4
#define N 80
/*****[1]*****/
int maxstr(char (*ss)[N],int m)
{
    int i,len,maxlen,n;
    maxlen=strlen(ss[0]);
    n=0;
    for(i=1; i<m; i++)
    {
        len = strlen(ss[i]);
        if(len>maxlen)
        {
            maxlen = len;
            n=i;
        }
    }
    return n;
}
void insert(char str[ ])
{
/*****[2]*****/
int i,j,len;
len=strlen(str);
while(str[j])
    if(isdigit(str[j]))
    {
        for(i=len;i>=j;i--)
            str[i+1]=str[i];
        str[j]=' ';
/*****[3]*****/
        j+=3;
        len++;
    }
    else j++;
}
int main()
{

```

```

int n;
char string[ ][N]={"A1B23CD45","EF2G34","ER45DF678","985GHJ211FGH"};
n=maxstr(string,M);
printf("最长字符串:%s\n",string[n]);
/*****[4]*****/
insert(string[]);
printf("变换后的字符串:%s\n",string[n]);
return 0;
}

```

答案:

[1] int maxstr(char (\*ss)[N],int m)

[2] int i,j=0,len;

[3] j+=2;

[4] insert(string[n]);

(16) 以下程序中函数 void csort(char str[][N],int n)对 str 指向的二维数组前 n 行中存储的 n 个字符串分别做如下处理:从中间将字符串一分为二,左半部分字符串按字符的 ASCII 码值从大到小排序(若字符串长度为奇数,则中间字符不参加排序)。

测试数据: "abcdefgh", "123498765"。

屏幕输出:

```

dcbaefgh
432198765

```

代码如下:

```

#include <stdio.h>
#include <string.h>
#define N 80
void csort(char str[][N],int n);
int main()
{
/*****[1]*****/
    char s[2][N]="abcdefgh","123498765";
    int i;
    csort(s,2);
    for(i=0;i<2;i++)
/*****[2]*****/
        puts(s);
    return 0;
}
void csort(char str[][N],int n)
{
    int i,j,k,len,half;
    char temp;

```



```

for (i=0;i<n;i++)
{
    /****** [3] *****/
    len=strlen(str[i][0]);
    /****** [4] *****/
    half=len/2;
    for (j=0;j<half-1;j++)
        for (k=j+1;k<half-1;k++)
            if (str[i][j]<str[i][k])
            {
                temp=str[i][j];
                str[i][j]=str[i][k];
                str[i][k]=temp;
            }
}
}

```

答案:

[1] char s[2][N]={"abcdefgh","123498765"};

[2] puts(s[i]);

[3] len=strlen(str);

[4] half=len/2+1;

(17) 对任意一个各位数字不全相同的四位数(例如 6388), 经如下变换总能得到一个固定的四位数 6174。变换方法是: 先将该数各位上的数字从小到大排列得到一个最小数(例如 3688), 从大到小排列得到一个最大数(例如 8863); 再用最大数减最小数得到一个新的四位数(例如  $8863-3688=5175$ ), 若相减后得到的数不足四位则高位补零; 称此过程为一次变换。再对新的四位数按上述方法实施一次变换又得到一个四位数。如此重复多次, 一定会得到 6174。以下程序用来验证这一结论(程序中设定: 若变换超过 20 次, 则认为上述论断为假)。函数 change 的功能是用整数 n 的各位数字分别组成一个最大数和一个最小数, 用大数减小数作为函数的返回值。

测试数据: 6388。

屏幕输出:

```

8863-3688=5175
7551-1557=5994
9954-4599=5355
5553-3555=1998
9981-1899=8082
8820-288=8532
8532-2358=6174
Change numbers:7

```

代码如下:

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

```
int change(int n)
{
    /*****[1]*****/
    int [], i, j, k, t;
    for (i=0; i<4; i++)
    {
        a[i]=n%10;
        n/=10;
    }
    for (i=0; i<3; i++)
    {
        k=i;
        for (j=i+1; j<4; j++)
            if (a[k]>a[j]) k=j;
        if(k!=i)
            t=a[i], a[i]=a[k], a[k]=t;
    }
    j=t/10;
    for (i=0; i<4; i++)
    {
        j=j*10+a[i];
        /*****[2]*****/
        t=t*10+a[4-i];
    }
    printf("%4d-%4d=%4d\n", t, j, t-j);
    return t-j;
}

int main()
{
    /*****[3]*****/
    int data, n;
    printf("Please input data:");
    scanf("%d", &data);
    while(1)
    {
        n++;
        data=change(data);
        /*****[4]*****/
        if(6174==data || n>=20)
            break;
    }
    if(n>=20)
        printf("failure\n");
    else
        printf("Change numbers:%d\n", n);
}
```

```

    return 0;
}

```

答案:

```
[1] int[4],i,j,k,t;
```

```
[2] t=t*10+a[3-i];
```

```
[3] int data=0,n;
```

```
[4] if(6174==data||n>=20)
```

(18) 以下程序实现子串查找并替换。函数 `substitution` 的功能是在 `s` 指向的字符串（简称 `s` 串）中查找 `t` 指向的子串（简称 `t` 串），并用 `g` 指向的字符串（简称 `g` 串）替换 `s` 串中所有的 `t` 串。

测试数据: `s:aaacdaaaaaaefaaaghaa`

`t:aaa`

`g:22`

屏幕输出:

```
22cd2222aef22ghaa
```

代码如下:

```

#include <stdio.h>
#include <string.h>
void substitution(char *,char *,char *);
int main()
{
    /****** [1] *****/
    char s[80]="aaacdaaaaaaefaaaghaa",t[2]="aaa",g[]="22";
    puts(s);
    substitution(s,t,g);
    puts(s);
    return 0;
}
/****** [2] *****/
void substitution(char *s,char *t,char *g)
{
    int i,j,k;
    char temp[80];
    /****** [3] *****/
    for (i=0;s[i]!='\0';i++)
    {
        for (j=i,k=0;s[j]==t[k]&& t[k]!='\0';j++,k++);
        if (t[k]!='\0')
        {
            /****** [4] *****/
            temp=s+j;

```

```

        strcpy(s+i,g);
        strcat(s,temp);
        i+=strlen(g)-1;
    }
}
}

```

答案:

```

[1] char s[80]="aacdaaaaaaefaaaghaa",t[4]="aaa",g[]="22";
[2] void substitution(char *s,char *t,char *g)
[3] for (i=0;s[i]!='\0';i++)
[4] strcpy(temp,s+j);

```

### 3.1.3 编程题

(1) 编写程序, 利用梯形公式计算  $f(x) = \int_a^b (\sin x + \cos x) dx$  的近似值。

答案:

```

#include <stdio.h>
#include <math.h>
int main()
{
    double s,h,a,b,x;
    long int i,n;
    printf("输入上限和下限(积分区间):");
    scanf("%lf %lf",&a,&b);
    printf("输入梯形数目:");
    scanf("%ld",&n);
    h=(b-a)/n;
    s=((sin(a)+cos(a))+(sin(b)+cos(b)))/2;
    for(x=a,i=1;i<n;i++)
    {
        s=s+(sin(x)+cos(x));
        x=x+h;
    }
    s=s*h;
    printf("s=%lf\n",s);
    return 0;
}

```

(2) 通过函数调用方式, 利用以下公式计算  $\pi$  的值(要求某项小于输入的精度就停止计算)

$$\frac{\pi}{2} = 1 + \frac{1}{3} + \frac{1 \times 2}{3 \times 5} + \frac{1 \times 2 \times 3}{3 \times 5 \times 7} + \frac{1 \times 2 \times 3 \times 4}{3 \times 5 \times 7 \times 9} + \dots + \frac{1 \times 2 \times \dots \times n}{3 \times 5 \times \dots \times (2n+1)}$$

答案:

```
#include <stdio.h>
#include <math.h>
int main()
{
    int i;
    double s=0,t=1,eps;
    printf("请输入精度:");
    scanf("%lf",&eps);
    i=1;
    while(fabs(t)>=eps)
    {
        s=s+t;
        t=t*i/(2*i+1);
        i++;
    }
    printf("PI=%lf\n",2*s);
    return 0;
}
```

(3) 输入正整数  $n$ , 求  $1\sim n$  的所有完数(完数是指一个数恰好等于除它本身外的因子之和。例如 6, 不包括它本身的因子是 1、2、3, 而  $1+2+3=6$ )。

答案:

```
#include <stdio.h>
int main()
{
    int i,n,s,k,p;
    printf("输入一个正整数:");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        k=i;
        s=0;
        p=1;
        while(p<k)
        {
            if(k%p==0)
            {
                s=s+p;
            }
            p++;
        }
        if(s==i)
            printf("%d 是完数\n",i);
    }
}
```

```

    return 0;
}

```

(4) 有一只猴子，第一天摘下若干桃子，当即吃掉了一半，又多吃了一个；第二天将剩下的桃子吃掉了一半，又多吃了一个。按照这样的吃法，每天都吃前一天剩下的一半，又多吃了一个，到第  $n$  天，就只剩下一个桃子。这只猴子第一天共摘下多少个桃子？

答案：

```

#include <stdio.h>
int main()
{
    int i,n,a;
    printf("输入天数: ");
    scanf("%d",&n);
    i=n;
    a=1;
    do
    {
        a=2*(a+1);
        i--;
    }while(i>1);
    printf("一共有%d个桃子!",a);
    return 0;
}

```

(5) 利用递归计算组合数  $C_n^m$  (其中  $1 < m \leq n \leq 30$ )。

答案：

```

#include <stdio.h>
long int combine(int,int);
int main()
{
    int n,m;
    long int p;
    scanf("%d %d",&n,&m);
    p=combine(n,m);
    printf("%ld\n",p);
    return 0;
}
long int combine(int n,int m)
{
    long int p;
    if(n==m)
        p=1;
    else if(m==1)
        p=n;
}

```

```

else
    p=combine(n-1,m)+combine(n-1,m-1);
return p;
}

```

(6) 有  $2n$  个棋子 ( $n \geq 4$ ) 排成一行, 开始位置为白色全部在左边, 黑色全部在右边, 例如  $\bigcirc \bigcirc \bigcirc \bigcirc \bullet \bullet \bullet \bullet$ 。移动棋子的规则是: 每次必须同时移动相邻两个棋子, 颜色不限, 可以左移也可以右移到空位上去, 但不能调换两个棋子的左右位置, 每次移动必须跳过若干棋子 (不能平移), 要求最后能够移成黑白相间的一行棋子。例如, 当  $n=4$  时, 最终排列情况为  $\bigcirc \bullet \bigcirc \bullet \bigcirc \bullet \bigcirc \bullet$ 。

答案:

```

#include <stdio.h>
void move(int);
int main()
{
    int n;
    scanf("%d",&n);
    move(n);
    return 0;
}
void move(int k)
{
    if(k==4)
    {
        printf("4,5-->9,10\n");
        printf("8,9-->4,5\n");
        printf("2,3-->8,9\n");
        printf("7,8-->2,3\n");
        printf("1,2-->7,8\n");
    }
    else
    {
        printf("%d,%d-->%d,%d\n",k,k+1,2*k+1,2*k+2);
        printf("%d,%d-->%d,%d\n",2*k-1,2*k,k,k+1);
        move(k-1);
    }
}

```

(7) 一个有  $n$  个元素的一维数组中, 将所有相同的数删去, 使之没有重复的数。例如一维数组中的数为 2, 2, 2, 3, 4, 4, 5, 6, 6, 6, 6, 7, 7, 8, 9, 9, 10, 10, 10, 删除后一维数组的内容为 2, 3, 4, 5, 6, 7, 8, 9, 10。

答案:

方法一 (先排序):

```
#include <stdio.h>
#define N 100
void sort(int [],int);
int delnum(int [],int);
int main()
{
    int i,n,k,a[N];
    printf("输入元素个数: ");
    scanf("%d",&n);
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    sort(a,n);
    k=delnum(a,n);
    for(i=0;i<k;i++)
        printf("%d",a[i]);
    printf("\n");
    return 0;
}
void sort(int x[],int n)
{
    int i,j,t;
    for(i=0;i<n-1;i++)
        for(j=0;j<n-1-i;j++)
            if(x[j]>x[j+1])
            {
                t=x[j];
                x[j]=x[j+1];
                x[j+1]=t;
            }
}
int delnum(int x[],int n)
{
    int i,j=0,t;
    int *p=x;
    t=p[0];
    for(i=1;i<n;i++)
        if(t==p[i])
            continue;
        else
        {
            x[j]=t;
            t=p[i];
            j++;
        }
    if(i>=n)
```



```

        x[j]=t;
        return j;
    }

```

方法二（直接删除）:

```

#include <stdio.h>
#define N 100
int delete_num(int [],int);
int main()
{
    int i,n,k,a[N];
    printf("输入元素个数:");
    scanf("%d",&n);
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    k=delete_num(a,n);
    for(i=0;i<k;i++)
        printf("%d",a[i]);
    printf("\n");
    return 0;
}
int delete_num(int x[],int n)
{
    int i,j,k;
    for(i=0;i<n;i++)
    {
        j=i+1;
        while(j<n)
        {
            if(x[i]==x[j])
            {
                for(k=j;k<n-1;k++)
                    x[k]=x[k+1];
                n--;
            }
            else
                j++;
        }
    }
    return n;
}

```

方法三（利用链表）:

```

#include <stdio.h>
#include <stdlib.h>

```

```
typedef struct node
{
    int x;
    struct node *next;
}Node;
int main()
{
    Node *head=NULL,*p,*q,*newnode,*r;
    int i,n;
    head=p=(Node *)malloc(sizeof(Node));
    printf("输入节点数:");
    scanf("%d",&n);
    scanf("%d",&p->x);
    for(i=1;i<n;i++)
    {
        newnode=(Node *)malloc(sizeof(Node));
        scanf("%d",&newnode->x);
        p->next=newnode;
        p=p->next;
        p->next=NULL;
    }
    p=head;
    while(p)
    {
        q=head;
        while(q->next!=p&&q->next!=NULL)
        {
            if(p->x==q->next->x)
            {
                q->next=q->next->next;
            }
            else
                q=q->next;
        }
        p=p->next;
    }
    p=head;
    while(p)
    {
        printf("%d",p->x);
        p=p->next;
    }
    printf("\n");
    return 0;
}
```

}

(8) 有  $n$  盏灯, 编号为  $1 \sim n$ 。第 1 个人把所有灯打开, 第 2 个人按下所有编号为 2 的倍数的开关 (这些灯被关掉), 第 3 个人按下所有编号为 3 的倍数的开关 (其中关掉的灯将被打开, 打开的灯将被关掉), 依此类推。一共有  $k$  个人, 最后有哪些灯开着?

答案:

```
#include <stdio.h>
int main()
{
    int a[1001]={0},i,j,n,k;
    scanf("%d %d",&n,&k);
    for(i=1;i<=k;i++)
    {
        for(j=1;j<=n;j++)
        {
            if(j%i == 0)
            {
                if(a[j] == 0)
                    a[j] = 1;
                else
                    a[j] = 0;
            }
        }
    }
    for(i=1;i<=n;i++)
        if(a[i]) printf("%4d",i);
    return 0;
}
```

(9) 输入  $n$  ( $1 < n \leq 500$ ), 输出下面  $n=5$  的逆时针螺旋矩阵。

$$\begin{bmatrix} 1 & 16 & 15 & 14 & 13 \\ 2 & 17 & 24 & 23 & 12 \\ 3 & 18 & 25 & 22 & 11 \\ 4 & 19 & 20 & 21 & 10 \\ 5 & 6 & 7 & 8 & 9 \end{bmatrix}$$

答案:

```
#include <stdio.h>
#define N 500
int a[N][N];
int main()
{
    int i,j,n,k;
    printf("输入矩阵阶数:");
```

```

scanf("%d", &n);
i=1;
j=1;
for(k=1;k<=n*n;k++)
{
    a[i][j]=k;
    if(i+j<n+1 && i+1>=j)
        i=i+1;
    else if(i+j>=n+1&&i>j)
        j=j+1;
    else if(i+j>n+1 && i<=j)
        i=i-1;
    else
        j=j-1;
}
for(i=1;i<=n;i++)
{
    for(j=1;j<=n;j++)
        printf("%5d",a[i][j]);
    printf("\n");
}
return 0;
}

```

(10) 在一个矩阵中，如果某元素在对应的行中最大，且在对应的列中最小，这样的元素称为鞍点。当然，一个矩阵中也可能没有鞍点。编程寻找矩阵中的鞍点位置。如果没有找到，则输出“该矩阵没有鞍点！”

**答案：**

```

#include <stdio.h>
#include <stdlib.h>
#define M 100
#define N 200
int SaddlePoint(int *,int *,int[][N],int ,int);
int main()
{
    int a[M][N],i,j,x,y,m,n;
    printf("输入矩阵行数和列数:");
    scanf("%d %d",&m,&n);
    printf("输入矩阵元素:\n");
    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
            scanf("%d",&a[i][j]);
    if(SaddlePoint(&x,&y,a,m,n))
        printf("鞍点的位置是(%d,%d)\n",x,y);
    else

```

```

        printf("该矩阵没有鞍点! \n");
    return 0;
}
int SaddlePoint(int *row,int *col,int x[][N],int m ,int n)
{
    int max,i,j,k,flag;
    for(i=0;i<m;i++)
    {
        max=0;
        flag=1;
        for(j=1;j<n;j++)
            if(x[i][j]>x[i][max])
                max=j;
        for(k=0;k<n;k++)
        {
            if(x[i][max]==x[i][k]&&max!=k)/*找重复的最大值*/
            {
                flag=0;
                break;
            }
        }
        if(flag==1)
        {
            for(k=0;k<m;k++)
                if(x[k][max]<=x[i][max]&&k!=i)
                {
                    flag=0;
                    break;
                }
        }
        if(flag==1)
        {
            *row=i;
            *col=max;
            return 1;
        }
    }
    return 0;
}

```

(11) 编程计算一个仅含有加法运算的表达式的值。要求表达式的长度不超过 255 个字符，中间没有空格与括号，且计算结果在整数范围以内。例如，输入 12+23+21，则输出结果为 56。

**答案：**

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

```

#include <string.h>
#define M 256
int main()
{
    char str[M];
    long int total=0,len,i,num=0;
    scanf("%s",str);
    len=strlen(str);
    for(i=0;i<len;i++)
        if(str[i]!='+')
        {
            num=num*10;
            num=num+(str[i]-'0');
        }
        else
        {
            total=total+num;
            num=0;
        }
    total=total+num;
    printf("%ld\n",total);
    return 0;
}

```

(12) 编写函数 `int del_name(char s[][20], int n)`, 其功能是在 `s` 指向的数组前 `n` 行存储的 `n` 个字符串中删除重复出现的字符串, 只保留第一次出现的字符串, 函数返回 `s` 指向的数组中剩余的字符串个数。主函数 `main` 中声明 `name` 并用测试数据初始化, 用 `name` 作实参调用函数以及 `del_name`, 将经过删除后 `name` 数组中剩余的字符串个数输出。

**答案:**

```

#include <stdio.h>
#include <string.h>
#define N 10
#define M 20
int del_name(char s[][M], int);
int main()
{
    int n,i,k;
    char name[N][M];
    scanf("%d",&n);
    for(i=0;i<n;i++)
        gets(name[i]);
    k=del_name(name,n);
    for(i=0;i<k;i++)

```

```

        printf("%s\n", name[i]);
    return 0;
}
int del_name(char s[][M], int n)
{
    int i, j, k=n, m, flag;
    for(i=0; i<=k; i++)
    {
        flag=1;
        for(j=i+1; j<=k; j++)
            if(strcmp(s[i], s[j])==0)
            {
                if(flag==1)
                {
                    --k;
                    flag=0;
                }
                for(m=j; m<k; m++)
                    strcpy(s[m], s[m+1]);
            }
    }
    return k;
}

```

(13) 设一个班级有  $N$  个人，学生信息包括姓名、学号以及数学、计算机、英语 3 门课的成绩。编写程序求 3 门课的总分，并将总分与总分最大值相等的同学的等级设置为“优秀”，总分与总分最小值相等的同学的等级设置为“不及格”，其余同学的等级设置为“合格”。

**答案：**

```

#include <stdio.h>
#include <string.h>
#define N 100
typedef struct student
{
    char name[10];
    int num;
    int maths;
    int computer;
    int english;
    int sum;
    char level[10];
} STU;
void fun(STU [], int );
int main()

```

```

{
    STU s[N];
    int i,n;
    scanf("%d",&n); /* 学生人数*/
    for(i=0;i<n;i++)
        scanf("%s %d %d %d %d",s[i].name,&s[i].num,&s[i].maths,&s[i].computer,
            &s[i].english);
    fun(s,n);
    for(i=0;i<n;i++)
        printf("%s %d %d %d %d %d %s\n",s[i].name,s[i].num,s[i].maths,s[i].computer,
            s[i].english,s[i].sum,s[i].level);
    return 0;
}
void fun(STU a[],int n)
{
    int i,mmax=0,mmin=300;
    for(i=0;i<n;i++)
    {
        a[i].sum=a[i].maths+a[i].computer+a[i].english;
        if(mmax<a[i].sum)
            mmax=a[i].sum;
        if(mmin>a[i].sum)
            mmin=a[i].sum;
    }
    for(i=0;i<n;i++)
    {
        if(mmax==a[i].sum)
            strcpy(a[i].level,"优秀");
        else if(mmin==a[i].sum)
            strcpy(a[i].level,"不及格");
        else
            strcpy(a[i].level,"合格");
    }
}

```

(14) 山顶有 10 个山洞，一只狐狸和一只兔子住在各自的山洞里，狐狸总想吃掉兔子。某天兔子对狐狸说：你想吃掉我有一个条件，先把洞从 1 到 10 编号，你从 10 号洞出发，先到 1 号洞找我；第二次隔一个洞找我，第三次隔两个洞找我，依此类推，次数不限，若能找到我，你就可以饱餐一顿。不过在没有找到我以前不能停下来。可是狐狸找了 1000 次也没有找到兔子。兔子躲在哪些山洞里，狐狸找不到？

**答案：**

```

#include <stdio.h>
#define N 10

```



```

#define MAX 1000
int main()
{
    int i,n,k=0,hole[N];
    scanf("%d",&n);
    for(i=0;i<n;i++)
        hole[i]=0;
    for(i=1;i<=MAX;i++)
    {
        k=(k+i)%n;
        hole[k]=1;
    }
    printf("兔子可能躲的山洞号为:\n");
    for(i=0;i<n;i++)
        if(hole[i]==0)
            printf("%4d",i+1);
    printf("\n");
    return 0;
}

```

(15) 某校的惯例是在每学期的期末考试之后发放奖学金。发放的奖学金共有 5 种，获取的条件各不相同：

① 院士奖学金，每人 8000 元，期末平均成绩高于 80 分 ( $> 80$ )，并且在本学期内发表 1 篇或 1 篇以上论文的学生均可获得。

② 五四奖学金，每人 4000 元，期末平均成绩高于 85 分 ( $> 85$ )，并且班级评议成绩高于 80 分 ( $> 80$ ) 的学生均可获得。

③ 成绩优秀奖，每人 2000 元，期末平均成绩高于 90 分 ( $> 90$ ) 的学生均可获得。

④ 西部奖学金，每人 1000 元，期末平均成绩高于 85 分 ( $> 85$ ) 的西部省份学生均可获得。

⑤ 班级贡献奖，每人 850 元，班级评议成绩高于 80 分 ( $> 80$ ) 的学生干部均可获得。只要符合条件就可以得奖，每项奖学金的获奖人数没有限制，每名学生也可以同时获得多项奖学金。例如，姚林的期末平均成绩是 87 分，班级评议成绩 82 分，同时他还是一位学生干部，那么他可以同时获得五四奖学金和班级贡献奖，奖金总数是 4850 元。

现在给出若干学生的相关数据，请计算哪些同学获得的奖金总数最高（假设总有同学能满足获得奖学金的条件）。

**答案：**

```

#include <stdio.h>
#include <string.h>
#define N 100
typedef struct student
{
    char name[30];
    int grade1;

```

```

    int grade2;
    char cadres;
    char west;
    int number;
}Scholar;
int main()
{
    int n,i,sum=0,amount,max=0;
    char temp[30];
    Scholar stu[N];
    scanf("%d",&n);
    sum=0;
    for(i=0;i<n;i++)
    {
        scanf("%s %d %d %c %c %d",stu[i].name,&stu[i].grade1,&stu[i].grade2,
            &stu[i].cadres,&stu[i].west,&stu[i].number);
        amount=0;
        if(stu[i].grade1>80 && stu[i].number>=1)
            amount+=8000;
        if(stu[i].grade1>85 && stu[i].grade2>80)
            amount+=4000;
        if(stu[i].grade1>90)
            amount+=2000;
        if(stu[i].grade1>85 && stu[i].west=='Y')
            amount+=1000;
        if(stu[i].grade2>80 && stu[i].cadres=='Y')
            amount+=850;
        sum+=amount;
        if(amount>max)
        {
            max=amount;
            strcpy(temp,stu[i].name);
        }
    }
    printf("%s\n%d\n%d\n",temp,max,sum);
    return 0;
}

```

## 3.2 模拟试卷及参考答案

### 3.2.1 笔试模拟试卷

#### 一、单项选择题

- (1) C语言规定,在一个C程序中,main函数( )。
- A. 必须在系统调用的库函数之后      B. 必须在程序的开始

C. 必须在程序的最后

D. 可以在任意位置

答案: D

(2) 以下叙述中错误的是 ( )。

A. 用户定义的标识符允许使用关键字

B. 用户定义的标识符应尽量做到见名知意

C. 用户定义的标识符开头必须为字母或下划线

D. 用户定义的标识符中大小写字母代表不同的标识符

答案: C

(3) 若定义  $\text{int } a=7; \text{float } x=2.5, y=4.7;$ , 则表达式  $x+a\%3*(\text{int})(x+y)\%2/4$  的值为 ( )。

A. 2.500000

B. 2.750000

C. 3.500000

D. 0.000000

答案: A

(4) 设:  $\text{int } a=1, b=2, c=3, d=4, m=2, n=2;$ , 执行  $(m=a>b) \&\& (n=c>d)$  后  $n$  的值为 ( )。

A. 1

B. 2

C. 3

D. 4

答案: B

(5) 若有数学式  $\frac{3ae}{bc}$ , 则不正确的 C 语言表达式是 ( )。

A.  $a/b/c*e*3$

B.  $3*a*e/b/c$

C.  $3*a*e/b*c$

D.  $a*e/c/b*3$

答案: C

(6) 若定义  $x$  为 `double` 型变量, 则能正确输入  $x$  值的语句是 ( )。

A. `scanf("%f", x);`

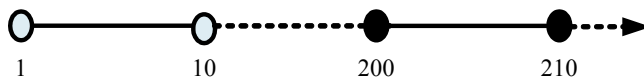
B. `scanf("%f", &x);`

C. `scanf("%lf", &x);`

D. `scanf("%5.1f", &x);`

答案: C

(7) 能正确表示下图中坐标轴实线部分的表达式是 ( )。



A.  $(x>=1) \&\& (x<=10) \&\& (x>=200) \&\& (x<=210)$

B.  $(x>=1) \parallel (x<=10) \parallel (x>=200) \parallel (x<=210)$

C.  $(x>1) \&\& (x<10) \parallel (x>=200) \&\& (x<=210)$

D.  $(x>=1) \parallel (x<=10) \&\& (x>=200) \parallel (x<=210)$

答案: C

(8) 若对两个数组  $a$  和  $b$  进行初始化:

```
char a []="ABCDEF";
```

```
char b []={'A', 'B', 'C', 'D', 'E', 'F'};
```

则下列叙述正确的是 ( )。

A.  $a$  与  $b$  数组完全相同

B.  $a$  与  $b$  数组长度相同

C.  $a$  与  $b$  数组都存放字符串

D. 数组  $a$  比数组  $b$  长度长

答案: D

(9) 下面不能正确进行字符串赋值操作的是 ( )。

- A. `char s[]={"ABCDE"};`                      B. `char s[5]={'A','B','C','D','E'};`  
 C. `char *s ; s="ABCDE";`                      D. `char str[6],*s=str; scanf("%s",s);`

答案: B

(10) 以下正确的函数定义是 ( )。

- A. `double fun(int x, int y)`  
`{`  
`z=x+y;`  
`return z;`  
`}`
- B. `double fun(int x,y)`  
`{`  
`int z;`  
`return z;`  
`}`
- C. `fun(int x,int y);`  
`{`  
`double z;`  
`z=x+y;`  
`return z;`  
`}`
- D. `double fun(int x, int y)`  
`{`  
`double z;`  
`z=x+y;`  
`return z;`  
`}`

答案: D

(11) 若调用一个非 void 型函数, 且此函数中没有 return 语句, 则正确的说法是 ( )。

- A. 该函数没有返回值                      B. 该函数返回若干系统默认值  
 C. 该函数返回一个用户所希望的值                      D. 该函数返回一个不确定的值

答案: D

(12) 若有定义:

```
double a[] = { 2.1, 3.6, 9.5};
double b = 6.0;
```

则以下赋值语句中错误的是 ( )。

- A. `b = a[2];`      B. `b = a + a[2];`      C. `a[1] = b;`      D. `b = a[0] + 7;`

答案: B

(13) 变量的指针实际上是变量的 ( )。

- A. 值                      B. 地址                      C. 名                      D. 一个标志

答案: B

(14) 若有说明 `int *p,m=5,n;`, 以下程序段正确的是 ( )。

- A. `p=&n;`  
`scanf("%d",&p);`
- B. `p=&n;`  
`scanf("%d",*p);`
- C. `scanf("%d",&n);`  
`*p=n;`
- D. `p = &n ;`  
`*p = m ;`

答案: D

(15) 下面程序段的运行结果是 ( )。

```
char *s="abcde";
```

```
s+=2; printf("%s",s);
```

- A. cde            B. 字符'c'            C. 字符'c'的地址            D. 不确定

答案: A

(16) 已有函数 max(a,b), 为了让函数指针变量 p 指向函数 max, 正确的赋值方法是 ( )。

- A. p=max;            B. p=max(a,b);            C. \*p=max;            D. \*p=max(a,b);

答案: A

(17) 以下程序的运行结果是 ( )。

```
#include <stdio.h>
#define ADD(x) x+x
int main(){
    int m=1,n=2,k=3,sum;
    sum = ADD(m+n)*k;
    printf("%d\n",sum);
    return 0;
}
```

- A. 9            B. 10            C. 12            D. 18

答案: B

(18) 有如下说明语句:

```
typedef struct stu{
    int a;
    float b;
} stutype;
```

则下面叙述不正确的是 ( )。

- A. struct 是结构体类型的关键字            B. stutype 是结构体变量名  
C. stutype 是结构体类型名            D. a 和 b 都是结构体成员名

答案: B

(19) 以下对结构体变量成员不正确的引用是 ( )。

```
struct pupil {
    char name[20]; int age; int sex;
} pup[5], *p=pup;
```

- A. scanf("%s",pup[0].name);            B. scanf("%d",&pup[0].age);  
C. scanf("%d",&(p->sex));            D. scanf("%d",p->age);

答案: D

(20) C 语言结构体类型变量在程序执行期间 ( )。

- A. 所有成员一直驻留在内存中            B. 占内存最小的成员驻留在内存中  
C. 占内存最大的成员驻留在内存中            D. 没有成员驻留在内存中

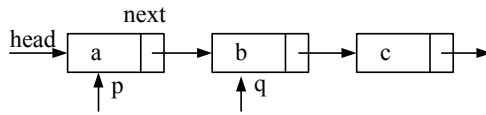
答案: A

(21) C 语言要读取计算机 D 盘上 test 文件夹下的 out.dat 文件, 则函数 fopen 中文件名字符串常量正确的输入是 ( )。

- A. "D:test/out.dat"                      B. "D:\test\out.dat"  
C. "D:/test/out.dat"                      D. "D:\\test\\out.dat"

答案: D

(22) 假定已经建立以下链表结构, 且指针 p 和 q 已经指向如图所示的节点。



则以下选项中可将 q 所指节点从链表中删除并释放该节点的语句组是 ( )。

- A. (\*p).next>(\*q).next; free(p);      B. p=q->next; free(q);  
C. p=q; free(q);                          D. p->next=q->next; free(q);

答案: D

(23) 若运行以下程序时输入 2345↵, 则程序的运行结果是 ( )。

```
#include <stdio.h>
int main()
{
    int c;
    while ((c=getchar( )) !='\n')
        switch (c-'2')
        {
            case 0:
            case 1: putchar(c+4);
            case 2: putchar(c+4); break;
            case 3: putchar(c+3);
            default: putchar(c+2); break;
        }
    printf("\n");
    return 0;
}
```

- A. 6677887                      B. 6689667                      C. 6677877                      D. 6688766

答案: A

(24) 以下程序段的运行结果是 ( )。

```
char a[ ]="nanjing", *p;
p=a;
while (*p!='\0') { printf("%c", *p-32); p++; }
```

- A. NANJING                      B. nanjing                      C. NA                              D. NAJIG

答案: A

## 二、填空题

(1) C 源程序的基本单位是\_\_\_\_\_。

答案：函数

(2) 设  $x$ 、 $y$  均为 `int` 型变量，写出描述“ $x$ 、 $y$  中有一个为非负数”的 C 语言表达式：

\_\_\_\_\_。

答案：`x>=0||y>=0`

(3) 至少执行一次循环体的循环语句是\_\_\_\_\_。

答案：`do while`

(4) 已知 `char c[]="he\t0will"`，则 `printf("%d",strlen(c))`的结果是\_\_\_\_\_。

答案：4

(5) 已知有二维数组声明 `int a[][4]={1,2,3,4,5,6,7,8,9}`；，则 `a` 数组有\_\_\_\_\_行。

答案：3

(6) 若有定义 `int a[2][3]={2,4,6,8,10,12}`；，则 `*(&a[0][0]+2*2+1)`的值是\_\_\_\_\_。

答案：12

### 三、程序阅读题

(1) 输入 5，9 回车后，以下程序输出结果的第一行是\_\_\_\_\_，第二行是\_\_\_\_\_。

```
#include <stdio.h>
void swap(int *,int *);
int main()
{
    int a, b;
    scanf("%d,%d",&a,&b);
    swap (&a,&b);
    printf("a=%d,b=%d\n",a,b);
    return 0;
}
void swap (int *p1,int *p2)
{
    int p;
    p=*p1;
    *p1=*p2;
    *p2=p;
    printf("*p1=%d,*p2=%d\n",*p1,*p2);
}
```

答案：`*p1=9,*p2=5, a=9,b=5`

(2) 输入 1234567，以下程序的运行结果是\_\_\_\_\_。

```
#include <stdio.h>
int main()
{
    unsigned int n,k=0,t=1;
    scanf("%u",&n);
    while(n)
    {
```

```

        if((n%10)%2==0)
        {
            k=k+(n%10)*t;
            t=t*10;
        }
        n=n/10;
    }
    printf("%u\n",k);
    return 0;
}

```

答案: 246

(3) 以下程序的运行结果是\_\_\_\_\_。

```

#include <stdio.h>
int f(int);
int main()
{
    int z;
    z=f(4);
    printf("%d\n",z);
    return 0;
}
int f(int x)
{
    if(x==0||x==1)
        return 3;
    else
        return x*x-f(x-2);
}

```

答案: 15

(4) 以下程序的运行结果是\_\_\_\_\_。

```

#include <stdio.h>
int fun(int);
int main()
{
    int i,a=5;
    for(i=0;i<3;i++)
        printf("%d ", fun(a));
    printf("\n");
    return 0;
}
int fun(int a)
{
    int b=0;
    static int c=3;
}

```



```

    b++;
    c++;
    return(a+b+c);
}

```

答案: 10 11 12

(5) 当运行以下程序时, 从键盘输入 “AhaMA Aha” ⏎, 则以下程序的运行结果是\_\_\_\_\_。

```

#include <stdio.h>
int main()
{
    char s[80],c='a';
    int i=0;
    gets(s);
    while(s[i]!='\0')
    {
        if(s[i]==c) s[i]=s[i]-32;
        else if (s[i]==c-32) s[i]=s[i]+32;
        i++;
    }
    puts(s);
    return 0;
}

```

答案: ahAMa ahA

(6) 以下程序的运行结果是\_\_\_\_\_。

```

#include <stdio.h>
#include <string.h>
void inv(char *);
int main()
{
    char a[]="nuist";
    inv(a);
    puts(a);
    return 0;
}
void inv(char *x)
{
    char *p=x,*q=x+strlen(x)-1,t;
    while(p<q)
    {
        t=*p;
        *p=*q;
        *q=t;
    }
}

```

```

        p++;
        q--;
    }
}

```

答案: tsiun

(7) 以下函数的功能是\_\_\_\_\_。

```

void conj(char *s1,char *s2)
{
    while (*s1)s1++;
    while (*s2)
        { *s1=*s2; s1++,s2++; }
    *s1='\0';
}

```

答案: 字符串连接

(8) 以下程序的运行结果是\_\_\_\_\_。

```

#include <stdio.h>
int main()
{
    int i,b,k=0;
    for (i=1; i<=5 ;i++)
    {
        b=i%2;
        while (b--==0) k++;
    }
    printf("%d,%d",k,b);
    return 0;
}

```

答案: 2, 0

#### 四、程序填空题

(1) 对于一个三位正整数, 如果它的每位上的数字的三次方和等于它本身, 那么就称这个数为三位水仙花数。以下程序的功能是输出所有仙花数。

```

#include <stdio.h>
#include <stdlib.h>
int main()
{
    int a,b,c,n;
    for(n=100;n<1000;n++)
    {
        a=n%10;
        b=_____ ;
        c=n/100;
    }
}

```

```

        if(_____)
            printf("%d\n",n);
    }
    return 0;
}

```

答案:  $(n/10)\%10$ 或 $(n\%100)/10$ ,  $n==a*a*a+b*b*b+c*c*c$

(2) 以下程序的功能是折半查找输入的数  $x$  是否在按从小到大排好序的数组  $a$  中, 请填写使程序完整。

```

#include <stdio.h>
#define N 8
int main()
{
    int a[N]={6,12,18,42,44,52,67,94};
    int low=0,mid,high=N-1,found=0,x;
    scanf("%d",&x);
    while((_____) &&(found==0))
    {
        mid=(low+high)/2;
        if(x>a[mid]) low=mid+1;
        else if(x<a[mid]) _____;
        else {found=1;break;}
    }
    if(found==1)
        printf("Search Successful:The index is:%d\n",mid);
    else
        printf("Can't search!\n");
    return 0;
}

```

答案:  $low\leq high$ ,  $high=mid-1$

(3) 函数 fun 的功能是: 计算  $f(x)$  的前  $n$  项和。若  $x=2.5$ ,  $n=15$ , 函数值为 1.917914。

$$f(x)=1+x-\frac{x^2}{2!}+\frac{x^3}{3!}-\frac{x^4}{4!}+\cdots+(-1)^{n-2}\frac{x^{n-1}}{(n-1)!}+(-1)^{n-1}\frac{x^n}{n!}$$

请填写使程序完整。

```

#include <stdio.h>
double fun(double x, int n)
{
    double f, t; int i;
    f = _____;
    t = -1;
    for (i=1; i<n; i++)
    {
        t *= (_____) * x / i;
    }
}

```

```

        f += _____;
    }
    return f;
}
int main()
{
    double x, y;
    x=2.5;
    y=fun(x, 15);
    printf("\nThe result is :\n");
    printf("x=%-12.6f y=%-12.6f\n", x, y);
    return 0;
}

```

答案: 1, -1, t

### 五、程序改错题

改正下面程序中的错误（注：错误在注释语句下面1~3行以内）。

以下程序中函数 fun 的功能是：根据输入的3个边长(整型值)，判断能否构成三角形，若能，则构成的是等边三角形还是等腰三角形。若能构成等边三角形，函数返回3；若能构成等腰三角形，函数返回2；若能构成一般三角形，函数返回1；若不能构成三角形，函数返回0。

```

#include <stdio.h>
/*****[1]*****/
void fun(int a,int b,int c)
{
    if(a+b>c && b+c>a && a+c>b)
    {
        /*****[2]*****/
        if(a==b || b==c)
            return 3;
        else if(a==b||b==c||a==c)
            return 2;
        else return 1;
    }
    else return 0;
}
int main()
{
    int a,b,c,shape;
    scanf("%d%d%d",&a,&b,&c);
    printf("a=%d, b=%d, c=%d\n",a,b,c);
    shape=fun(a,b,c);
    if (shape==3) printf("构成等边三角形\n");
    else if(shape==2) printf("构成等腰三角形\n");
    else if(shape==1) printf("构成一般三角形\n");
    else printf("不能构成三角形\n");
}

```

```

    return 0;
}

```

答案:

[1] int fun(int a, int b, int c)

[2] if(a==b||b==c)

## 六、编程题

(1) 编写函数 fun, 其功能是计算并输出下列式子的值:

$$s = 1 + \frac{1}{1+2} + \frac{1}{1+2+3} + \cdots + \frac{1}{1+2+\cdots+n}$$

在主函数中从键盘输入 n, 并输出计算结果。

```

#include <stdio.h>
double fun(int n)
{

}

int main()
{
    int n;
    double s;
    scanf("%d", &n);
    s=fun(n);
    printf("s=%lf\n\n", s);
    return 0;
}

```

答案:

```

double fun(int n)
{
    double s=0;
    int i,t=0;
    for(i=1;i<=n;i++)
    {
        t=t+i;
        s=s+1.0/t;
    }
    return s;
}

```

(2) 有 40 个学生, 每个学生的数据包括学号、姓名和 3 门课的成绩, 这些学生的信息都存放在文件 in.dat 中, 编写一个 C 语言程序实现对学生 3 门课的总分进行降序排序, 并将最后 5% 的学生评定为不合格, 其余为合格, 并将学生信息 (包括学号, 姓名, 3 门课的成绩、总分、等级) 写入 out.dat 文件。

答案:

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define N 40
struct student
{
    int num;
    char name[10];
    float s1,s2,s3;
    float sum;
    char level[10];
};
int main()
{
    struct student st[N],t;
    int i,j,k;
    FILE *fp1,*fp2;
    if((fp1=fopen("in.dat","r"))==NULL)
    {
        printf("Can't open the file\n");
        exit(0);
    }
    for(i=0;i<N;i++)
    {
        fscanf(fp1,"%d %s %f %f %f",&st[i].num,st[i].name,&st[i].s1,
        &st[i].s2,&st[i].s3);
        st[i].sum=st[i].s1+st[i].s2+st[i].s3;
    }
    fclose(fp1);
    for(i=0;i<N-1;i++)
    {
        k=i;
        for(j=i+1;j<N;j++)
            if(st[j].sum>st[k].sum) k=j;
        if(i!=k)
        {
            t=st[i];
            st[i]=st[k];
            st[k]=t;
        }
    }
    for(i=0;i<N;i++)
        if(i<=0.95*N)
            strcpy(st[i].level,"合格");
        else
```

```

        strcpy(st[i].level,"不合格");
    if((fp2=fopen("out.dat","w"))==NULL)
    {
        printf("Can't open file!\n");
        exit(0);
    }
    for(i=0;i<N;i++)
        fprintf(fp2,"%d %s %f %f %f %f %s\n",st[i].num,st[i].name,
st[i].s1,st[i].s2,st[i].s3,st[i].sum,st[i].level);
    fclose(fp2);
    return 0;
}

```

### 3.2.2 期中机试模拟试卷

#### 一、单项选择题

(1) 计算机高级语言程序的运行方法有编译执行和解释执行两种，以下叙述中正确的是 ( )。

- A. C 语言程序仅可以编译执行
- B. C 语言程序仅可以解释执行
- C. C 语言程序既可以编译执行又可以解释执行
- D. 以上说法都不对

答案: A

(2) 一个 C 语言程序由 ( ) 组成。

- A. 若干过程
- B. 若干子程序
- C. 函数
- D. 一个主程序和若干子程序

答案: C

(3) 一个 C 程序的执行 ( )。

- A. 从本程序的第一个函数开始,到本程序的 main 函数结束
- B. 从本程序的第一个函数开始,到本程序的最后一个函数结束
- C. 从本程序的 main 函数开始,到本程序的最后一个函数结束
- D. 从本程序的 main 函数开始,到 main 函数结束

答案: D

(4) 以下叙述正确的是 ( )。

- A. 在 C 程序中,main 函数必须位于程序的最前面
- B. 在对一个 C 程序进行编译的过程中,可发现注释中的拼写错误
- C. C 语言本身没有输入输出语句
- D. 程序的每行中只能写一条语句

答案: C

(5) 以下叙述不正确的是 ( )。

- A. 在 C 程序中, 注释只能位于一个语句的后面
- B. C 程序的基本组成单位是函数
- C. 一个 C 源程序可由一个或多个函数组成
- D. 一个 C 源程序必须包含一个 main 函数

答案: A

(6) 以下叙述正确的是 ( )。

- A. C 语言程序是由过程和函数组成的
- B. C 语言函数不可以单独编译
- C. C 语言中除了 main 函数, 其他函数不可以单独文件形式存在
- D. C 语言函数可以嵌套调用, 例如 fun(fun(x))

答案: D

(7) 设: int a=5,b=4,c=3; 下列表达式中值不为 0 的是 ( )。

- A. !(a<0||a>5)
- B. !a||b
- C. a&&!b
- D. a>b>c

答案: A

(8) 下列表达式中, 不满足“当 x 的值为偶数时值为真, 为奇数时值为假”的要求的是 ( )。

- A. (x/2\*2-x)==0
- B. !(x%2)
- C. !(x%2==0)
- D. x%2==0

答案: C

(9) 数学式  $\frac{1+\sqrt{|x+y|}}{e^x y}$  正确的 C 语言表达式是 ( )。

- A. 1+sqrt(|x+y|)/exp(x)\*y
- B. (1+sqrt(|x+y|))/exp(x)\*y
- C. (1+sqrt(|x+y|))/(exp(x)\*y)
- D. (1+sqrt(fabs(x+y)))/(exp(x)\*y)

答案: D

(10) C 语言中的标识符只能由字母、数字和下划线 3 种字符组成,且第一个字符 ( )。

- A. 必须为下划线
- B. 必须为字母
- C. 可以是字母、数字和下划线中的任意一种



D. 必须为字母或下画线

答案: D

(11) 在 C 语言中, 要求运算数必须是整型的运算符是 ( )。

- A. %
- B. /
- C. ++
- D. !=

答案: A

(12) 若有定义语句 `int k1=10, k2=20;`, 执行表达式 `(k1=k1>k2)&&(k2=k2>k1)` 后, `k1` 和 `k2` 的值分别为 ( )。

- A. 0 和 1
- B. 0 和 20
- C. 10 和 1
- D. 10 和 20

答案: B

(13) 以下关于 C 语言常量的叙述中错误的是 ( )。

- A. 所谓常量, 是指在程序运行过程中其值不能被改变的量
- B. 常量分为整型常量、实型常量、字符常量和字符串常量
- C. 常量可分为数值型常量和非数值型常量
- D. 经常被使用的变量可定义成常量

答案: D

(14) 以下选项中能表示逻辑值“假”的是 ( )。

- A. 0.000001
- B. 0
- C. 100.0
- D. 1

答案: B

(15) 以下 4 个选项中, 3 个用户标识符均不合法的选项是 ( )。

- A. `_123`      `temp`      `INT`
- B. `A`          `p_o`          `do`
- C. `b-a`        `goto`        `int`
- D. `float`      `lao`          `_A`

答案: C

(16) 以下程序段中, 与语句 `k=a>b?(b>c?1:0):0;` 功能相同的是 ( )。

- A. `if((a>b)&&(b>c)) k=1;`  
    `else k=0;`
- B. `if((a>b)||(b>c)) k=1;`  
    `else k=0;`

- C. if(a>b) k=1;  
     else if(b>c) k=1;  
     else k=0;
- D. if(a<=b) k=0;  
     else if(b<=c) k=1;

答案: A

(17) 有以下程序段:

```
int k, j, s;
for(k = 2; k < 6; k++, k++)
{
    s = 1;
    for(j = k; j < 6; j++)
        s += j;
}
printf("%d\n", s);
```

其输出结果是 ( )。

- A. 15  
 B. 10  
 C. 24  
 D. 9

答案: B

(18) 有以下程序:

```
#include <stdio.h>
int main()
{
    int x=1,a=0,b=0;
    switch (x)
    {
        case 0: b++;
        case 1: a++;
        case 2: a++;b++;
    }
    printf("a=%d,b=%d", a, b);
    return 0;
}
```

该程序的输出结果是 ( )。

- A. a=2,b=2  
 B. a=2,b=1  
 C. a=1,b=1  
 D. a=1,b=0

答案: B

(19) 设  $j$  和  $k$  都是 `int` 型变量, 则下面的 `for` 循环语句 ( )。

```
for(j=0,k=0;j<=9&&k!=876;j++) scanf("%d",&k);
```

- A. 最多执行 9 次
- B. 一次也不执行
- C. 最多执行 10 次
- D. 是无限循环

答案: C

(20) 有以下程序:

```
int main()
{
    int num = 0;
    while( num <= 2 ) { num++; printf("%d,",num ); }
    return 0;
}
```

其输出结果是 ( )。

- A. 1,2,3,
- B. 1,2,3,4,
- C. 1,2,
- D. 0,1,2

答案: A

(21) 有以下程序:

```
int main()
{
    int x, i;
    for(i = 1; i <= 100; i++)
    {
        x = i;
        if( ++x % 2 == 0)
            if( ++x % 3 == 0)
                if( ++x % 7 == 0)
                    printf("%d ", x);
    }
    printf("\n");
    return 0;
}
```

其输出结果是 ( )。

- A. 28 70
- B. 39 81
- C. 42 84

D. 26 68

答案: A

(22) 有以下程序:

```
int main()
{
    int i;
    for(i=0; i<3; i++)
        switch(i)
        {
            case 1: printf("%d", i);
            case 2: printf("%d", i);
            default : printf("%d", i);
        }
    return 0;
}
```

其输出结果是 ( )。

- A. 011122
- B. 120
- C. 012020
- D. 012

答案: A

(23) 有以下语句:

```
int j=-1;
if(j<=1) printf("****\n");
else     printf("%%%\n");
```

其输出为 ( )。

- A. %%%%c
- B. 有错, 执行不正确
- C. \*\*\*\*
- D. %%%%

答案: C

(24) C语言中 while 和 do...while 循环的主要区别是 ( )。

- A. while 的循环控制条件比 do...while 的循环控制条件更严格
- B. do...while 的循环体至少无条件执行一次
- C. do...while 的循环体不能是复合语句
- D. do...while 允许从外部转到循环体内

答案: B

(25) for(x=0,y=0;(y=123)&&(x<4);x++);的执行次数是 ( )。

- A. 4次

- B. 是无限循环
- C. 3 次
- D. 循环次数不定

答案: A

(26) 有以下程序:

```
int main()
{
    int x=1,y=0,a=0,b=0;
    switch(x)
    {
        case 1:switch(y)
            {
                case 0:a++;break;
                case 1:b++;break;
            }
        case 2:a++;b++;break;
        case 3:a++;b++;break;
    }
    printf("a=%d,b=%d\n",a,b);
}
```

其输出结果是 ( )。

- A. a=2,b=1
- B. a=2,b=2
- C. a=1,b=0
- D. a=1,b=1

答案: A

(27) 有以下程序段:

```
int i, j, m = 0;
for(i = 1; i <= 15; i += 4)
    for(j = 3; j <= 19; j += 4)
        m++;
printf("%d\n", m);
```

其输出结果是 ( )。

- A. 15
- B. 12
- C. 20
- D. 25

答案: C

(28) 有以下程序:

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

```

int main()
{
    int a=10;
    unsigned char c;
    float f=10000;
    double x;
    c=f/=a*(x=2.56);
    printf("%d %d %3.2f %3.2lf\n",a,c,f,x);
    return 0;
}

```

其输出结果是 ( )。

- A. 25 400 400.00 2.56
- B. 25 144 400.00 2.56
- C. 26 144 400.00 2.56
- D. 25.6 400 400.00 2.56

答案: B

(29) 设有定义 `int a=1,b=2,c=3;`, 以下语句中执行效果与其他 3 个不同的是 ( )。

- A. `if(a>b){ c=a;a=b;b=c;}`
- B. `if(a>b) c=a;a=b;b=c;`
- C. `if(a>b){ c=a;a=b;b=c;}`
- D. `if(a>b) c=a;a=b;b=c;`

答案: D

(30) 以下程序的功能是: 按顺序读入 10 名学生 4 门课程的成绩, 计算每位学生的平均分并输出。

```

int main()
{
    int n,k;
    float score,sum,ave;
    sum=0.0;
    for(n=1;n<=10;n++)
    {
        for(k=1;k<=4;k++)
        {
            scanf("%f",&score);
            sum+=score;
        }
        ave=sum/4.0;
        printf("NO%d:%f\n",n,ave);
    }
    return 0;
}

```

上述程序运行后结果不正确，调试中发现有一条语句出现在不正确的位置。这条语句是（ ）。

- A. ave=sum/4.0;
- B. sum=0.0;
- C. sum+=score;
- D. printf("NO%d: %f\n",n,ave);

答案: B

(31) 若函数调用时的实参为变量，以下关于函数形参和实参的叙述中正确的是（ ）。

- A. 同名的实参和形参占同一存储单元
- B. 形参和实参分别占用不同的存储单元
- C. 实参和其对应的形参共用同一存储单元
- D. 形参只是形式上的存在，在调用之前不占用具体存储单元

答案: D

(32) 函数 funA 调用了函数 funB，函数 funB 又调用了函数 funA，这种情况（ ）。

- A. 称为函数的循环调用
- B. 在 C 语言中不允许出现
- C. 称为函数的间接递归调用
- D. 称为函数的直接递归调用

答案: C

(33) 设函数中有整形变量 n，为保证其在未赋初值的情况下初值为 0，应该选择的存储类别是（ ）。

- A. register
- B. auto
- C. auto 或 register
- D. static

答案: D

(34) 以下语句中正确的是（ ）。

- A. scanf("%d", 0xAB);
- B. int a=b=c\0;
- C. int a[3]=0;
- D. printf("%d", "\\);

答案: D

(35) 若有以下声明: float x=1234.567;int n=4;char m;，则表达式 m=n+'A'+120UL+x 的值的类型是（ ）。

- A. char
- B. double
- C. float
- D. unsigned long

答案: B

(36) 以下选项中定义的标识符 p 中不能做 p++运算的是 ( )。

- A. int p;
- B. int a[10],\*p=a;
- C. int a[2],\*p[2]={a};
- D. int a[2][2],(\*p)[2]=a;

答案: C

(37) 若有以下声明: int a[4][3];int (\*ptr)[3]=a, \*p=a[0];, 则以下能正确引用数组元素 a[1][2]的表达式是 ( )。

- A. \*((ptr+1)[2])
- B. (\*(p+5))
- C. (\*ptr+1)+2
- D. (\*(a+1)+2)

答案: D

(38) 以下二维中数组声明中合法的是 ( )。

- A. int b[][3]={{1,2,3},{},{4,5,6}};
- B. int b[][sizeof(int)]= {0};
- C. int b[2][2]={{1,2,3},{4,5,6}};
- D. int b[2][]={{1,2,3},{4,5,6}};

答案: A

(39) 以下对字符串的操作中错误的是 ( )。

- A. char \*s[2]={"aaa","bbb"};
- B. char \*s[2];strcpy(s[0],"aaa");strcpy(s[1],"bbb");
- C. char s[2][5]={"aaa","bbb"};
- D. char s[2][5];strcpy(s[0],"aaa");strcpy(s[1],"bbb");

答案: B

(40) 下列语句中 ( ) 定义了一个能存储 20 个字符的数组。

- A. char b[20];
- B. char c[21];
- C. int a[21];
- D. int d[20];

答案: A

## 二、编程题

(1) 函数 fun 用于求下面分数数列的前 n 项的和:

$$s = -\sqrt{\frac{2}{1}} + \sqrt{\frac{3}{2}} - \sqrt{\frac{5}{3}} + \sqrt{\frac{8}{5}} - \dots$$

输入项数 n 得到前 n 项和 s。例如, 输入 n=10, s=-0.219960。注意: 不要修改 main 函数的结构, 只在 fun 函数的两个注释行之间编写函数代码。

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



```

#include <math.h>
double fun(int n)
{
    /*****Begin*****/

    /*****End*****/
}
int main()
{
    int n,i;
    double s;
    FILE *in,*out;
    scanf("%d",&n);
    s=fun(n);
    printf("s=%lf\n",s);
    in=fopen("in.dat","r");
    out=fopen("out.dat","w");
    while(!feof(in))
    {
        fscanf(in,"%d\n",&n);
        s=fun(n);
        fprintf(out,"%lf\n",s);
    }
    fclose(in);
    fclose(out);
    return 0;
}

```

**参考答案:**

```

int i,sign=-1;
double a=2,b=1,s=0;
for(i=1;i<=n;i++)
{
    s=s+sign*sqrt(a)/sqrt(b);
    a=a+b;
    b=a-b;
    sign=-sign;
}
return s;

```

(2) 编写函数 `int fun(int lim, int aa[MAX])`，该函数的功能是求出小于或等于 `lim` 的所有素数并放在 `aa` 数组中，该函数返回所求出的素数的个数。注意，部分源程序如下，请勿改动 `main` 函数和其他函数中的任何内容，仅在函数 `fun` 的两个注释行之间填入所编写的若干语句。

```

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#define MAX 100
int fun(int lim, int aa[MAX])
{
/*****Begin*****/

/*****End*****/
}
int main()
{
    FILE *wf,*in;
    int limit,i,sum;
    int aa[MAX];
    printf("输入一个整数");
    scanf("%d",&limit);
    sum=fun(limit,aa);
    for(i=0;i<sum;i++)
    {
        if(i%10==0&&i!=0) /*每行输出 10 个数*/
            printf("\n ");
        printf("%5d ",aa[i]);
    }
    in=fopen("in.dat","r");
    fscanf(in,"%d",&limit);
    wf=fopen("out.dat","w");
    sum=fun(limit,aa);
    for(i=0;i<sum;i++)
    {
        fprintf(wf,"%d\n",aa[i]);
    }
    fclose(wf);
    fclose(in);
    return 0;
}

```

**参考答案:**

```

int i,j,k=0,t;
for(i=2;i<lim;i++)
{
    t=(int)sqrt(i);
    for(j=2;j<=t;j++)
        if(i%j==0)

```

```

        break;
    if (j>t)
        aa[k++]=i;
    }
    return k;

```

### 三、程序填空题

给定程序中，函数 fun 的功能是：有  $N \times N$  矩阵，以主对角线为对称轴，对称元素相加并将结果存放在左下三角元素中，右上三角元素置为 0。例如，若  $N=3$ ，有下列矩阵：

```

1  2  3
4  5  6
7  8  9

```

计算结果为

```

1   0  0
6   5  0
10  14  9

```

注意，部分源程序如下。请勿改动 main 函数和其他函数中的任何内容，仅在函数 fun 代码中的横线上填入所编写的若干表达式或语句。

试题程序：

```

#include <stdio.h>
#define N 4
void fun(int _____)
{
    int i,j;
    for (i=1;i<N;i++)
    {
        for (j=0;j<i;j++)
        {
            _____=t[i][j]+t[j][i];
            _____=0;
        }
    }
}
int main()
{
    int i,j,t[][N]={12,21,33,44,52,45,36,27,18,9,66,73,84,45,36,
    27,18};
    printf("\nThe original array:\n ");
    for (i=0;i<N;i++)
    {
        for (j=0;j<N;j++)
            printf("%4d",t[i][j]);
    }
}

```

```

    printf("\n");
}
fun(t);
printf("\nThe result is:\n");
for(i=0;i<N;i++)
{
    for(j=0;j<N;j++)
        printf("%4d",t[i][j]);
    printf("\n");

}
return 0;
}

```

答案:

t[N][N] (或 t[][N]), t[i][j], t[j][i]

#### 四、程序改错题

以下程序中, 函数 fun 的功能是: 求表达式  $s=aa\cdots aa\cdots\cdots aaa-aa-a$  的值 (此处  $aa\cdots aa$  表示  $n$  个  $a$ ,  $a$  和  $n$  的值为  $1\sim 9$ )。例如  $a=3, n=6$ , 则以上表达式为

$s=333\ 333-33\ 333-3333-333-33-3$

其值是 296 298。a 和 n 是函数 fun 的形参, 表达式的值作为函数值传回 main 函数。请改正程序中的错误, 使它能计算出正确的结果。注意: 不要改动 main 函数, 不得增行或删行, 也不得更改程序的结构。

```

#include <conio.h>
#include <stdio.h>
long fun(int a,int n)
{
    int j;
    /***** [1] *****/
    long s=0,t=1;
    /***** [2] *****/
    for(j=0;j<=n;j++)
        t=t*10+a;
    s=t;
    for(j=1;j<n;j++)
    {
        /***** [3] *****/
        t=t%10;
        s=s-t;
    }
    return(s);
}
int main()

```

```

{
    int a,n;
    printf("\nPlease enter a and n: ");
    scanf("%d%d",&a,&n);
    printf("\nThe value of function is %ld\n", fun(a,n));
    return 0;
}

```

答案:

[1] long s=0,t=0;或 long int s=0,t=0;

[2] for(j=0;j<n;j++)

[3] t=t/10;或 t/=10;

### 3.2.3 期末机试模拟试卷

#### 一、单项选择题

(1) 一个 C 程序的执行 ( )。

- A. 从本程序的第一个函数开始,到本程序的 main 函数结束
- B. 从本程序的第一个函数开始,到本程序的最后一个函数结束
- C. 从本程序的 main 函数开始,到本程序的最后一个函数结束
- D. 从本程序的 main 函数开始,到 main 函数结束

答案: D

(2) 以下叙述中错误的是 ( )。

- A. C 语言编写的每个函数都可以独立编译并执行
- B. C 语言编写的函数源程序,其文件名后缀可以是 C
- C. 一个 C 语言程序只能有一个主函数
- D. C 语言编写的函数都可以作为一个独立的源程序文件

答案: A

(3) 已知有 float x,i,j; 则以下选项中含有语法错误的表达式是 ( )。

- A.  $0 \leq x < 100$
- B.  $i=j==0$
- C.  $!(x+i=x+j)$
- D.  $(char)(i+j)$

答案: C

(4) 在 C 语言中, char 型数据在内存中的存储形式是 ( )。

- A. 补码
- B. 原码
- C. ASCII 码
- D. 反码

答案: C

(5) 以下关于 C 语言数据类型使用的叙述中错误的是 ( )。

- A. 若要处理“人员工资”的相关数据, 应使用单精度类型
- B. 若只处理“真”和“假”两种逻辑值, 应使用逻辑类型
- C. 若要保存带有多位小数的数据, 应使用双精度类型
- D. 若要准确无误差地表示自然数, 应使用整数类型

答案: B

(6) 若 a 是数值类型, 则逻辑表达式  $(a==1) \vee (a!=1)$  的值是 ( )。

- A. 不知道 a 的值, 不能确定表达式的值
- B. 0
- C. 2
- D. 1

答案: D

(7) 判断字符型变量 c1 是否为数字字符的正确表达式为 ( )。

- A.  $(c1 >= '0') \&\& (c1 <= '9')$
- B.  $(c1 >= 0) \&\& (c1 <= 9)$
- C.  $('0' >= c1) \vee ('9' <= c1)$
- D.  $'0' <= c1 <= '9'$

答案: A

(8) 表达式  $18/4 * \sqrt{4.0}/8$  值的数据类型为 ( )。

- A. char
- B. double
- C. float
- D. 不确定

答案: B

(9) 设 x 为 double 型变量, 则以下语句中能将 x 中的数值保留小数点后三位, 第四位进行四舍五入的是 ( )。

- A.  $x = x * 1000 + 0.5 / 1000.0$
- B.  $x = (x * 1000 + 0.5) / 1000.0$
- C.  $x = (\text{int})(x * 1000 + 0.5) / 1000.0$
- D.  $x = (\text{int})(x / 1000 + 0.5) * 1000.0$

答案: C

(10) if 语句的基本形式是“if(表达式)语句”, 以下关于“表达式”值的叙述中正确的是 ( )。

- A. 必须是正数
- B. 可以是任意合法的数值
- C. 必须是整数值
- D. 必须是逻辑值

答案: B

(11) 为了避免在嵌套的条件语句 if...else 中产生二义性, C 语言规定 else 子句总是

与 ( ) 配对。

- A. 其后面最近的 if
- B. 其前面最近的 if
- C. 缩进位置相同的 if
- D. 同一行上的 if

答案: B

(12) 有以下程序段: `int a=3,b=2,c=1; if(a>b>c)a=b; else a=c;`, 则 a 的值为 ( )。

- A. 2
- B. 3
- C. 1
- D. 0

答案: C

(13) 以下描述中正确的是 ( )。

- A. `do...while` 循环中, 根据情况可以省略 `while`
- B. `do...while` 循环中循环体只能是一条可执行语句, 不能使用复合语句
- C. `do...while` 循环由 `do` 开始, 到 `while` 结束, 在 `while` (表达式) 后面不能写分号
- D. 在 `do...while` 循环中, 一定要有能使 `while` 后面表达式的值变为零 (“假”) 的操作

答案: D

(14) 有以下程序:

```
#include <stdio.h>
int main()
{
    int a=7;
    while(a--);
    printf("%d\n",a);
    return 0;
}
```

程序运行后的输出结果是 ( )。

- A. -1
- B. 1
- C. 0
- D. 7

答案: A

(15) 设 a、b 和 c 都是 int 型变量, 且 a=3, b=4, c=5, 则以下的表达式中值为 0 的是 ( )。

- A. `a||b+c&&b-c`
- B. `a<=b`

C. !((a<b)&&!c||1)

D. a&&b

答案: C

(16) 有以下程序:

```
#include <stdio.h>
int main()
{
    int a1,a2; char c1,c2;
    scanf("%d%c%d%c",&a1,&c1,&a2,&c2);
    printf("%d,%c,%d,%c",a1,c1,a2,c2);
    return 0;
}
```

若想通过键盘输入,使得 a1 的值为 12, a2 的值为 34, c1 的值为字符 a, c2 的值为字符 b, 程序输出结果是“12, a, 34, b”, 则正确的输入格式是 ( )。

A. 12,a,34,b ↵

B. 12 a34 b ↵

C. 12a34b ↵

D. 12 a 34 b ↵

答案: C

(17) 以下程序的功能是计算  $s=1+\frac{1}{2}+\frac{2}{3}+\frac{3}{4}+\cdots+\frac{8}{9}+\frac{9}{10}$ 。

```
#include <stdio.h>
int main()
{
    int n;
    double s;
    s=1.0;
    for(n=2;n<=10;n++)
        s=s+(n-1)/n;
    printf("%lf\n",s);
    return 0;
}
```

该程序的输出结果是错误的, 导致错误结果的代码行是 ( )。

A. s=1.0;

B. for(n=2;n<=10;n++);

C. s=s+(n-1)/n;

D. printf("%lf\n",s);

答案: C

(18) 有以下程序:



```
#include <stdio.h>
int main()
{
    int x=1,y=0;
    if(!x) y++;
    else if(x==0)
    if(x)y+=2;
    else y+=3;
    printf("%d\n",y);
    return 0;
}
```

程序运行后的输出结果是 ( )。

- A. 3
- B. 2
- C. 0
- D. 1

答案: C

(19) 有以下程序:

```
#include <stdio.h>
int f(int x,int y)
{return ((y-x)*x);}
int main()
{
    int a=3,b=4,c=5,d;
    d=f(f(a,b),f(a,c));
    printf("%d\n",d);
}
```

程序运行后的输出结果是 ( )。

- A. 10
- B. 8
- C. 7
- D. 9

答案: D

(20) 有以下程序:

```
#include <stdio.h>
int f()
{
    static int x=1;
    x+=1;
    return x;
}
```

```
int main()
{
    int i,s=1;
    for(i=1;i<=5;i++) s+=fun();
    printf("%d\n",s);
    return 0;
}
```

程序运行后的输出结果是 ( )。

- A. 21
- B. 6
- C. 120
- D. 11

答案: A

(21) 有以下程序:

```
#include <stdio.h>
int fun (int x,int y)
{
    if (x!=y)
        return ((x+y)/2);
    else
        return (x);
}
int main()
{
    int a=4,b=5,c=6;
    printf("%d\n",fun(2*a,fun(b,c)));
    return 0;
}
```

程序运行后的输出结果是 ( )。

- A. 12
- B. 6
- C. 6.5
- D. 8

答案: B

(22) 下列叙述中正确的是 ( )。

- A. C 语言规定必须用 **main** 作为主函数名, 程序将从此函数开始执行, 到此函数结束
- B. **main** 可作为用户标识符, 用以命名任意一个函数作为主函数
- C. 可以在程序中由用户指定任意一个函数作为主函数, 程序将从此开始执行
- D. C 语言程序将从源程序中第一个函数开始执行

答案: A

(23) 如果函数 funA 调用了函数 funB, 函数 funB 又调用了函数 funA, 这种情况 ( )。

- A. 称为函数的循环调用
- B. 在 C 语言中不允许出现
- C. 称为函数的间接递归调用
- D. 称为函数的直接递归调用

答案: C

(24) 若有代数式  $\sqrt{n^x + e^x}$  (其中 e 代表自然对数的底数, 不是变量), 则以下能够正确表示该代数式的 C 语言表达式是 ( )。

- A. sqrt(fabs(pow(n,x)+exp(x)))
- B. sqrt(fabs(pow(n,x)+pow(x,e)))
- C. sqrt(fabs(pow(x,n)+exp(x)))
- D. sqrt(abs(n<sup>x</sup>+e<sup>x</sup>))

答案: A

(25) 以下数组声明合法的是 ( )。

- A. int x[10];
- B. int x(10);
- C. int n,x[n];
- D. int x[10]

答案: A

(26) 若对两个数组 a 和 b 进行初始化:

```
char a[]="ABCDEF";
char b[]={ 'A', 'B', 'C', 'D', 'E', 'F'};
```

则下列叙述正确的是 ( )。

- A. a 与 b 数组都存放字符串
- B. 数组 a 比数组 b 长度长
- C. a 与 b 数组完全相同
- D. a 与 b 数组长度相同

答案: B

(27) 若有以下定义:

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

则下列表达式中正确的是 ( )。

- A. a[a[0]=2] = 2;
- B. a = &a[0];
- C. &a[0] = a;
- D. \*(a[0]+1)=1;

答案: A

(28) 已知下列程序段:

```
char a[3], b[]="Hello";
a=b;
printf("%s", a);
```

则 ( )。

- A. 运行后将输出 Hello
- B. 运行后将输出 He
- C. 运行后将输出 Hel
- D. 编译出错

答案: D

(29) 若要对 a 进行自增运算, 则以下对 a 的说明正确的是 ( )。

- A. int b[10], \*a=b;
- B. char (\*a)[3];
- C. int a[3][2];
- D. char \*a[ ]={"12","ab"};

答案: A

(30) 设有程序段: char s[ ]="china"; char \*p; p=s;, 则以下叙述正确的是 ( )。

- A. \*p 与 s[0]相等
- B. 数组 s 中的内容和指针变量 p 中的内容相等
- C. s 数组长度和 p 所指向的字符串长度相等
- D. s 和 p 完全相同

答案: A

(31) 若有说明 int \*p1, \*p2, m=5, n;, 以下均是正确赋值语句的选项是 ( )。

- A. p1=&m; p2=p1;
- B. p1=&m; \*p1=\*p2;
- C. p1=&m; p2=&n; \*p1=\*p2;
- D. p1=&m; p2=&p1;

答案: A

(32) 若有定义 int (\*p)[4];, 则标识符 p ( )。

- A. 是一个指针数组名
- B. 定义不合法
- C. 是一个指向整型变量的指针
- D. 是一个指针, 它指向一个含有 4 个整型元素的一维数组

答案: D

(33) 设有以下程序段:

```
char a[ ]="language", *p;
p=a;
while (*p!='u') { printf("%c", *p-32); p++; }
```

其运行结果是 ( )。

- A. language
- B. langUAGE
- C. LANGUAGE
- D. LANG

答案: D

(34) 在“文件包含”预处理命令形式中, 当#include 后面的文件名用""(双引号)括起时, 寻找被包含文件的方式是 ( )。

- A. 仅搜索源程序所在目录
- B. 直接按系统设定的标准方式搜索目录
- C. 先在源程序所在目录中搜索, 再按系统设定的标准方式搜索
- D. 仅搜索当前目录

答案: C

(35) 若有以下定义:

```
#define N 2
#define Y(n) ((N+1)*n)
```

则执行语句  $z=2*(N+Y(5))$ ; 后,  $z$  的值为 ( )。

- A. 无确定值
- B. 语句有错误
- C. 34
- D. 70

答案: C

(36) 以下对结构体类型变量的定义中不正确的是 ( )。

A. struct  
{  
    int num;  
    float age;  
} student;  
struct student std1;

B. struct  
{  
    int num;  
    float age;  
} std1;

C. struct student  
{  
    int num;  
    float age;  
}std1;

```
D. #define STUDENT struct student
    STUDENT
    {
        int num;
        float age;
    } std1;
```

答案: A

(37) 当定义一个结构体变量时, 系统分配给它的内存是 ( )。

- A. 结构体中最后一个成员所需内存量
- B. 各成员所需内存量中的最大值
- C. 结构体中第一个成员所需内存量
- D. 各成员所需内存量的总和

答案: D

(38) 当定义一个结构体变量时, 系统分配给它的内存是 ( )。

- A. 各成员所需内存量的总和
- B. 结构体中第一个成员所需内存量
- C. 各成员所需内存量中的最大值
- D. 结构体中最后一个成员所需内存量

答案: A

(39) 若执行 `fopen` 函数时发生错误, 则函数的返回值是 ( )。

- A. EOF
- B. 1
- C. 地址值
- D. 0

答案: D

(40) 以下能作为输入文件名字符串常量的是 ( )。

- A. `c:user\text.txt`
- B. `c:\user\text.txt`
- C. `"c:\user\text.txt"`
- D. `"c:\user\text.txt"`

答案: C

## 二、程序设计题

(1) 编写函数 `fun`, 其功能是: 根据整型形参 `n`, 计算如下公式的值前 `n` 项和。

$$A_1 = 1, A_2 = \frac{1}{\sqrt{1+A_1}}, A_3 = \frac{1}{\sqrt{1+A_2}}, \dots, A_n = \frac{1}{\sqrt{1+A_{n-1}}}$$

例如, 若 `n=10`, 则应输出 `7.754769`。注意: 只能在 `fun` 函数的两个注释行之间编写若干条语句, 不要改动 `main` 函数, 不得更改程序的结构。

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

```

#include<math.h>
double fun(int n)
{
/*****Begin*****/

/*****End*****/
}
int main()
{
    int n;
    FILE *in,*out;
    printf("\nPlease enter n: ");
    scanf("%d",&n);
    printf("S%d=%lf\n",n,fun(n));
    in=fopen("in74.dat","r");
    out=fopen("out74.dat","w");
    while(!feof(in))
    {
        fscanf(in,"%d",&n);
        fprintf(out,"%lf\n",fun(n));
    }
    fclose(out);
    fclose(in);
    return 0;
}

```

**参考答案:**

```

int i;
double s=0,t=1;
for(i=1;i<=n;i++)
{
    s=s+t;
    t=1.0/sqrt(1+t);
}

```

(2) 设一个班级有  $N$  个学生, 学生信息包括姓名、学号以及数学、计算机、英语 3 门课成绩。函数 `fun` 的功能是求 3 门课的总分, 并将总分与总分最大值相等的同学的等级设置为优秀, 总分与总分最小值相等的同学的等级设置为不及格, 其余同学的等级设置为合格。注意: 只能在 `fun` 函数的两个注释行之间编写若干条语句, 不要改动 `main` 函数, 不得更改程序的结构。

```

#include <stdio.h>
#include <string.h>
#define N 5

```

```
typedef struct student
{
    char name[10];
    int num;
    int maths;
    int computer;
    int english;
    int sum;
    char level[10];
} STU;
void fun(STU a[],int n)
{
    /*****Begin*****/

    /*****End*****/
}
int main()
{
    FILE *in,*out;
    STU s[6*N]={{ "A001",1,34,67,80},{ "B003",3,78,87,90},{ "A002",2,90,98,99},
    { "B002",4,56,78,98},{ "A005",5,35,67,79}};
    int i,n;
    fun(s,N);
    for(i=0;i<N;i++)
        printf("%s %d %d %d %d %d %s\n",s[i].name,s[i].num,s[i].maths,
        s[i].computer,
        s[i].english,s[i].sum,s[i].level);
    in=fopen("in75.dat","r");
    out=fopen("out75.dat","w");
    i=0;
    while(!feof(in))
    {
        fscanf(in,"%s %d %d %d %d\n",s[i].name,&s[i].num,&s[i].maths,
        &s[i].computer,&s[i].english);
        i++;
    }
    n=i;
    fun(s,n);
    for(i=0;i<n;i++)
        fprintf(out,"%s %d %d %d %d %d %s\n",s[i].name,s[i].num,s[i].maths,
        s[i].computer,s[i].english,s[i].sum,s[i].level);
    fclose(in);
}
```



```

    fclose(out);
    return 0;
}

```

### 参考答案:

```

int i,mmax=0,mmin=300;
for(i=0;i<n;i++)
{
    a[i].sum=a[i].maths+a[i].computer+a[i].english;
    if(mmax<a[i].sum)
        mmax=a[i].sum;
    if(mmin>a[i].sum)
        mmin=a[i].sum;
}
for(i=0;i<n;i++)
{
    if(mmax==a[i].sum)
        strcpy(a[i].level,"优秀");
    else if(mmin==a[i].sum)
        strcpy(a[i].level,"不及格");
    else
        strcpy(a[i].level,"合格");
}

```

### 三、程序填空题

某专业两个班已经分别对计算机程序设计课程的考试成绩进行了排名，排名后的成绩分别存放在数组 a 和数组 b 中，现要按专业来进行排名。注意：部分源程序如下。仅在横线上填入若干表达式或语句，其他部分不要改动。

```

#include <stdio.h>
#define N 8
#define M 6
void merge(int a[],int b[],int c[],int n,int m)
{
    int i=0,j=0,k=0;
    while(____[1]____)
    {
        if(a[i] < b[j])
            c[k++]=b[j++];
        else
            c[k++]=a[i++];
    }
    while(____[2]____)
    {
        c[k]=a[i];
        i++;
    }
}

```

```

        k++;
    }
    while(____ [3] ____ )
    {
        c[k]=b[j];
        j++;
        k++;
    }
}
int main()
{
    int j;
    int a[N] = {98,91,75,64,55,54,43,32};
    int b[M] = {90,84,70,61,23,15};
    int c[____ [4] ____];
    printf("a 数组:\n");
    for(j=0;j<N;j++)
        printf("%5d",a[j]);
    printf("\n");
    printf("b 数组:\n");
    for(j=0;j<M;j++)
        printf("%5d",b[j]);
    printf("\n");
    merge(____ [5] ____);
    printf("合并 a 和 b 后:\n");
    for(j=0;j<N+M;j++)
        printf("%5d",c[j]);
    printf("\n");
    return 0;
}

```

答案:

[1]  $i < n \ \&\& \ j < m$ 或  $j < m \ \&\& \ i < n$ 、 $n > i \ \&\& \ m > j$ 、 $m > j \ \&\& \ n > i$ 、 $(i < n) \ \&\& \ (j < m)$ 、 $(j < m) \ \&\& \ (i < n)$ 、 $(n > i) \ \&\& \ (m > j)$ 、 $(m > j) \ \&\& \ (n > i)$

[2]  $i < n$  或  $n > i$

[3]  $j < m$ 或  $m > j$

[4]  $N+M$  或  $M+N$

[5]  $a,b,c,N,M$

#### 四、程序改错题

N 个字典序的整数已放在一维数组中，在以下程序中，函数 fun 的功能是：利用折半查找算法查找整数 m 在数组中的位置。若找到，则返回其下标值；反之，则返回-1。请改正程序中的错误，使它能得出正确的结果。注意：不得增行或删行，也不得更改程序的结构。

```

#include <stdio.h>
#define N 10
int fun(int *a,int m)
{ int low=0,high=N-1,mid;
/*****[1]*****/
    while(low>high)
    {
        mid=(low+high)/2;
/*****[2]*****/
        if(m>*(a+mid))
            high=mid-1;
/*****[3]*****/
        else if(m<=(a+mid))
            low=mid+1;
        else return(mid);
    }
    return(-1);
}
int main()
{
    int i,a[N]={-3,4,7,9,13,24,67,89,100,180},k,m;
    printf("a 数组中的数据如下:");
    for(i=0;i<N;i++)
        printf("%d ",a[i]);
    printf("Enter m: ");
    scanf("%d",&m);
/*****[4]*****/
    k=fun(a[N],m);
    if (k>=0)
        printf("m=%d,index=%d\n",m,k);
    else
        printf("Not be found!\n");
    return 0;
}

```

答案:

[1] while(low<=high) 或 while(high>=low)

[2] if(m<\*(a+mid)) 或 if(\*(a+mid)>m), if(m<a[mid]), if(a[mid]>m)

[3] else if(m>\*(a+mid)) 或 else if(\*(a+mid)<m), else if(a[mid]<m), else if(m>a[mid])

[4] k=fun(a,m);