

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. 下面程序是输入 n ($n < 13$), 计算 $1! + 3! + 5! + \dots + n!$ 的值。

```
#include <stdio.h>
#include <stdlib.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);
    system("pause");
    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. 已定义一个含有 N 个元素的数组 s , 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。

```
#include <stdio.h>
#include <stdlib.h>
#define N 30
```

```

#define M 5
void fun1(double s[])
{
    int k, i;
    for(k = 2, i = 0; i < N; i++)
    {
        _____ [1] _____;
        k += 2;
    }
}
void fun2(double s[], double w[])
{
    double 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;
    double s[N], w[M];
    fun1(s);
    _____ [4] _____;
    for(i = 0; i < N; i++)
    {
        if(i % 5 == 0) printf("\n");
        printf(" %8.2lf", s[i]);
    }
    printf("\n");
    for(i = 0; i < M; i++)
        printf(" %8.2lf", w[i]);
    system("pause");
    return 0;
}

```

答案:

- [1] $s[i] = k$ 或 $s[i] = (i + 1) * 2$
 [2] $\text{if}((i + 1) \% M == 0)$ 或 $\text{if}((i + 1) / M * M == i + 1)$
 [3] $\text{sum} = 0$ 或 $\text{sum} = 0.0$
 [4] $\text{fun2}(s, w)$

3. 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。

```
#include <stdio.h>
#include <string.h>
#include <stdlib.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);
    system("pause");
    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. 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。

```
#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);
system("pause");
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 $\text{fprintf}(\text{fp}, "%s\n", \text{str})$ 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。

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>
#include <stdlib.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] ____;
}
system("pause");
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. 输出 100~1000 的各位数字之和能被 15 整除的所有数,输出时每 10 个一行。

```

#include <stdio.h>
#include <stdlib.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] ____;
        }
    }
    system("pause");
    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. 从键盘上输入一字符串和一个字符,在字符串中删除与输入字符相同的字符。

```
#include <stdio.h>
#include <stdlib.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);
    system("pause");
    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. 输入一字符串,删除该字符串中的所有数字字符。

```
#include <stdio.h>
#include <stdlib.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);
    system("pause");
    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. 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。

```

#include <stdio.h>
#include <stdlib.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]);
    system("pause");
    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. 数组中元素已递增排序,下面的函数为二分法查找 key 值,若找到 key 则返回对应的下标,否则返回-1。

```
#include <stdio.h>
#include <stdlib.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);
    system("pause");
    return 0;
}
```

答案:

```
[1] low<=high 或 high>=low
[2] high=mid-1
[3] low=mid+1
[4] return mid 或 return(mid)
```

11. 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,则函数的值为 732。

```
#include <stdio.h>
#include <math.h>
#include <stdlib.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));
    system("pause");
    return 0;
}
```

答案:

[1] $n < 10$ 或 $10 > n$
[2] $high \% j == 0$ 或 $!(high \% j)$
[3] `return sum` 或 `return v(sum)`

12. 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"。

```
#include <stdio.h>
#include <stdlib.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("\n 请输入字符串 S:");
    scanf(" %s", s);
    fun(s, t);
    printf("\n 运行结果: %s\n", t);
    system("pause");
    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. 输入 N 个数,最大的元素与最后一个元素交换,最小的元素与第一个元素交换,输出这些数。

```

#include <stdio.h>
#include <stdlib.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);
    system("pause");
    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)

```