



华中科技大学计算机科学与技术学院 2022~2023 第一学期

“C 语言程序设计” 考试试卷 (A 卷)

考试方式 闭卷 考试日期 2023-02-23 考试时长 150 分钟

专业班级 学 号 姓 名

题号	一	二	三	四	五	总分	核对人
分值	10	14	24	24	28	100	
得分							

注意：试卷后面附有运算符的优先级和结合性表，该页可撕下作草稿。

分 数	
评卷人	

一、计算表达式的值（每个小题相互独立，不会互相影响。本大题共 10 小题，每小题 1 分，共 10 分。）

解答
内容
不得
超过
装订
线

1. 请根据下面的声明，计算 (1) ~ (5) 题表达式的值并填入各题后面的下划线中。

```
short x=4,y=0xcd,z=0x20;
```

```
char s[] = "hello",d;
```

```
int a=3,b=4,c=5;
```

(1) $!(a+b)+c-1\&\&c+b/3$ 值为:

(2) $\text{strlen}(s)$ 值为:

(3) $d=y$ 值为:

(4) $c -= a \mid ++b$ 值为:

(5) $x \ll 2 \mid y$ 值为:

2. 请根据下面的声明，计算 (6) ~ (10) 题表达式的值并填入各题后面的下划线中。

```
struct T{
```

```
    int a[3];
```

```
    char *s;
```

```
    struct T *x;
```

```
}b[2]={{{1,2,3},"HELLO",b},{-1,0,1},"WORLD",b+1}},*p=b;
```

(6) $++p \rightarrow a[1]$ 值为:

(7) $(*++p).s$ 值为:

(8) $(--(b[1].x)) \rightarrow a[2]$ 值为:

(9) $(p+1) \rightarrow a[1] ? 1 : 0$ 值为:

(10) $(p++) \rightarrow a[0] + (b[0].x) \rightarrow a[2]$ 值为:

分 数	
评卷人	

二、**改错题**（划线指出下面各题程序中的错误并改正，使之实现题目的功能。本大题共 7 小题，每小题 2 分，共 14 分。提示：每小题有 2 个错误。）

1. 求 $2/1$ 、 $3/2$ 、 $5/3$ 、 $8/5$ 、 $13/8$ 、 $21/13$ 、 \dots 前 30 项之和。

```
#include <stdio.h>
int main()
{
    int i, t1 = 2, t2 = 1, x;
    float sum = 0;
    for (i = 1; i <= 30; i++) {
        sum += t1 / t2;
        x = t1;
        t1 += t2;
        t2 = x;
    }
    printf("sum = %d\n", sum);
    return 0;
}
```

2. 函数 fun()的功能是计算正整数 num 的各位上的数字平方之和。例如，输入 124，则输出应为 21；若输入 326，则输出应为 49。

```
#include<stdio.h>
long fun(long num)
{
    long k = 1;
    do{
        k+= (num%10)*(num%10);
        num/= 10;
    } while(num);
}
```

```
int main()
{
    long n;
    printf(" Please enter a number: ");
    scanf(" %ld ", &n);
    printf(" %ld\n ", fun(n));
    return 0;
}
```

3. 函数 fun()的功能是将字符串 s 中位于偶数位置（最左边为第 0 位）的字符或 ASCII 码为奇数的字符放入字符串 t 中。例如，若 s 为 ADFESH，则 t 应是 AFES。

```
void fun(char s[ ], char t[ ])
```

```

{
    int i, j = 0;
    for(i = 0; i < strlen(s); i++)
        if(i%2 == 0 || s[i]%2 != 0)
            t[++j] = s[i];
    t[j] = '\0';
}

```

4. 逆序输出数组元素。

```

#include <stdio.h>
#define N 6
int main()
{
    int i, a[N] = {2, 5, 7, 14, 19, 22};
    for(i = N; i >= 0; i--)
        printf("%d ", a[i]);
    return 0;
}

```

解答内容不得超过装订线

5. 函数 fun 的功能是删除字符串 s 中第一次出现的字符 'b'，函数返回一个指向最终的字符串 s 的指针，若字符串 s 中不存在字符 'b'，则返回 NULL。例如，输入 qwebrrtb，则输出应为 qwertb；若输入 qwer，则输出应为 Not exist!。

```

#include <stdio.h>
char *fun(char s[ ])
{
    int i, k = 0;
    while(s[k] != '\0') {
        if(s[k] == 'b') continue;
        k++;
    }
    if(s[k] == '\0') return s;
    else {
        do {
            s[k] = s[k+1];
            k++;
        } while(s[k] != '\0');
        return s;
    }
}

int main()
{
    char s[30], *p;
    fgets(s, 30, stdin);
    p = fun(s);
}

```

```

    if(p==NULL) printf("Not exist!\n");
    else printf("%s\n",p);
    return 0;
}

```

6. 函数 fun()的功能是：在字符串 s 中找出 ASCII 码最小的字符，将其放在最左边位置上，并将该字符前的原字符向后顺序移动。例如，调用 fun()函数之前字符串为 ebfAgCDg，调用后字符串为 AebfgCDg。

```
void fun(char s[ ] )
```

```

{
    char min, *q;
    int i = 0;
    min = s[i];
    while(s[i] != 0) {
        if(min>s[i]) {
            q = &(s+i);
            min = s[i];
        }
        i++;
    }
    while(q>s) {
        *q = *q-1;
        q--;
    }
    s[0] = min;
}

```

7. 函数 fun()的功能是：从 n 个学生的成绩中统计出高于平均分的学生人数，人数由函数值返回，平均分存放在形参 average 所指的存储单元中。假如 6 名学生的成绩为：

84 65.5 68 96.5 87 56, 则高于平均分的学生人数为 3 (平均分为 76.1666667)。

```
int fun(float *s, int n, float *average)
```

```

{
    float ave, t = 0;
    int count = 0, k, i;
    for(k = 0; k<n; k++) t+=s[k];
    ave = t/n;
    for(i = 0; i<n; i++)
        if(s[i]>ave) count+1;
    average = ave;
    return count;
}

```

分 数	
评卷人	

三、简答题（本大题共 6 小题，每小题 4 分，共 24 分。）

1. 一个 2 字节的 short 型变量的最大允许值是多少？如果再加 1，结果是多少？
2. 请写出一个 C 表达式，如果字符变量 ch 的值是大写字母，则将其值修改成小写字母；如果字符变量 ch 的值是小写字母，则将其值修改成大写字母；如果 ch 的值不是字母，其值不变。
3. 请写一个 C 表达式，将 int 型变量 x 的最高二进制位设置为 1，其余位不变，要求所写表达式在 int 为 2 或 4 字节的机器上均能正确执行。

解
答
内
容
不
得
超
过
装
订
线

4. 下列程序段定义了 3 个变量 i，类型分别为 int、long 与 float。指出每个变量的存储类，它们分别在那些行进行了声明和使用。

```

1      int  i;
2      void  fun(long i)
3      {
4          long  x=i;
5          {
6              float  i;
7              i=3.4;
8          }
9          x=i+2;
10     }
11     int  *p=&i;
```

5. 判断以下叙述是否正确，如果不正确，请说明原因。
 - (1) 表达式 $z+=x+++y$ 中的词法元素（记号）数目是 7 个。
 - (2) 若有定义 `char x[]="abcde", y[]={ 'a','b','c','d','e' }`; 则数组 x 和数组 y 等价。
 - (3) `typedef char *STRING` 和 `#define STRING char*` 中命名的 STRING 使用上没有区别。
 - (4) 变量可以命名为 INT。

6. 写出下列声明语句。

(1) f 是值为 3 的“只读”型整型变量;

(2) g 是一个无参指针函数，其返回值是指向包含 2 个指针元素的数组的指针，该数组中每个元素是指向有一个字符指针参数且无返回值的函数。

分 数	
评卷人	

四、程序分析题（本大题共 6 小题，每小题 4 分，共 24 分。）

1. 写出下面程序的运行结果。

```
#include<stdio.h>
long fib(int g)
{
    switch (g) {
        case 0: return 0;
        case 1: case 2: return 1;
    }
    return(fib(g-1)+fib(g-2));
}
int main()
{
    long k;
    k=fib(7);
    printf("k=%d\n",k);
    return 0;
}
```

2. 写出下面程序的运行结果。

```
#include <stdio.h>
int main()
{
    int i,r;
    char s1[80]="bus",s2[80]="book";
    for (i=r=0;s1[i]!='\0'&& s2[i]!='\0';i++)
        if (s1[i]==s2[i]) i++;
        else { r=s1[i]-s2[i]; break; }
    printf("%d",r);
    return 0;
}
```

3. 写出下面程序的运行结果。

```

#include <stdio.h>
#define M 5
#define f(x) x*x
#define ff(x) (x*x)
int main()
{
    int n1,n2;
    n1=100/f(M);
    n2=100/ff(M);
    printf("n1=%d,n2=%d\n",n1,n2);
    return 0;
}

```

4. 写出下面程序的运行结果。

```

#include <stdio.h>
#define M 4
void fun(int *a)
{
    int i,j,k,m;
    for(i=M;i>0;i--) {
        k=(a+M-1);
        for (j=M-2;j>=0;j--)
            *(a+j+1)=*(a+j);
        *a=k;
        for(m=0;m<M;m++)
            printf("%d",*(a+m));
        printf("\n");
    }
}

```

解
答
内
容
不
得
超
过
装
订
线

```

int main()
{
    int a[M]={1,2,3,4};
    fun(a);
    return 0;
}

```

5. 写出下面程序的运行结果。

```

#include <stdio.h>
void amov(int *p, int (*a)[3], int n)
{
    int i,j,s=0;
    for (i=0;i<n;i++)
        for(j=0;j<n;j++) {
            *p=(*(a+j))[i];

```

```

        s=s+*p;
        p++;
    }
    printf("\ns=%d\n",s);
}

int main()
{
    int i,*p,a[3][3]={ {1,3,5},{2,4,6},{7,8,9}};
    p=&a[0][0];
    for (i=0;i<3;i++) printf("%d  ",*(p+i*3));
    putchar('\n');
    for (i=0;i<3;i++) printf("%d  ",*((a+i)+i));
    amov(p,a,3);
    return 0;
}

```

6. 假设磁盘上有 3 个文本文件，其文件名和内容分别为：

文件名 内容

f1 aaa!

f2 bbb!

f3 ccc!

此外，磁盘上还有以下程序，经编译、连接后生成名为 e12_41.exe 的可执行文件。若在系统提示符下输入：e12_42 f1 f2 f3<回车>，则程序的运行结果是什么。

```

#include <stdio.h>
#include <stdlib.h>
void sub(FILE *fp)
{
    char c;
    while((c=getc(fp))!='\n') putchar(c+1);
}

int main(int argc, char *argv[])
{
    FILE *fp;
    void sub(FILE *);
    int i=1;
    while (--argc>0)
        if((fp=fopen(argv[i++], "r"))==NULL) {
            printf("Can't open the file!\n");
            exit(-1);
        }
        else {
            sub(fp);
        }
}

```



```

        fclose(fp);
    }
    return 0;
}

```

分 数	
评卷人	

五、程序完善题（本大题有 4 小题，给出的都是部分程序，在下划线处填入合适代码来完善程序, 使之能够正确实现题目的功能。本大题共 14 空，每空 2 分，共 28 分。）

1. 函数 selSort(int a[], int n)采用选择排序法，对长度为 n 的一维整型数组 a 进行降序排序。

```
void selSort(int a[], int n)
```

```

{
    int i, j;
    for (i=0; i<n-1; i++) {
        int idx = i;
        for (j=i+1; j<n; j++) {
            if ( ① < a[j])
                idx = j;
        }
        if (idx != i)
            a[i] = a[i] + a[idx], a[idx] = ②, a[i] = ③;
    }
}

```

① _____ ② _____ ③ _____

2. 函数 delSubstr(char *str, const char *substr)用于删除母串 str 中所出现的所有子串 substr。该函数在实现时用到了标准函数库 string.h 中的字符串处理函数。

```
#include <string.h>
```

```
void delSubstr (char *str, const char *substr)
```

```

{
    int lenm = strlen(str), lens = strlen(substr);
    char *pch = str;
    int i, j;
    while ((pch=strstr( ①, substr)) != NULL) {
        for (i=0; i<lens; i++) *pch++ = '\0';
    }
    for (pch=str, i=0; i<lenm; i++) {
        if (*(str+i) != '\0')
            *pch++ = ②;
    }
    *pch = ③;
}

```

① _____ ② _____ ③ _____

解答内容不得超过装订线

3. 函数 SimProc(unsigned char inst)对单字节指令 inst 进行译码并执行指令对应的功能函数，返回执行结果。inst 的最高 2 bit 是指令类型码，取值：00、01、10 和 11，分别对应 4 个功能函数：func0、func1、func2 和 func3，这 4 个功能函数已实现；inst 的低 6 bit，包含了执行功能函数所需用到的参数：arg0、arg1 和 arg2。4 类指令低 6 bit 格式不同，如下表所示：

函数	7	6	5	4	3	2	1	0	函数	7	6	5	4	3	2	1	0
func0	0	0	x	x	x	x	x	x	func1	0	1	x	x	x	x	x	x
	类型码		arg0							类型码		arg0			arg1		
func2	1	0	x	x	x	x	x	x	func3	1	1	x	x	x	x	x	x
	类型码		arg0		arg1		arg2			类型码		arg0			arg1		arg2

```

struct cmd0 { unsigned char ty: 2; unsigned char arg0: 6; };
struct cmd1 { unsigned char ty: 2; unsigned char arg0: 3; unsigned char arg1: 3; };
struct cmd2 { unsigned char ty: 2; unsigned char arg0: 2; unsigned char arg1: 2; unsigned char arg2: 2;};
struct cmd3 { unsigned char ty: 2; unsigned char arg0: 3; unsigned char arg1: 2; unsigned char arg2: 1;};
union inscode {
    unsigned char code;  struct cmd0 ins0;  struct cmd1 ins1;  struct cmd2 ins2;  struct cmd3 ins3;
};
int func0(void);
int func1(void);
int func2(void);
int func3(void);
int arg0, arg1, arg2;

```

```

int SimProc(unsigned char inst)
{
    union inscode oneins;
    int typ = ①;
    int (*exe[])(void) = {func0, func1, func2, func3};
    oneins.code = inst;
    switch (typ) {
        case 0: arg0 = ②; break;
        case 1: arg0 = oneins.ins1.arg0; arg1 = oneins.ins1.arg1; break;
        case 2: arg0 = oneins.ins2.arg0; arg1 = oneins.ins2.arg1; arg2 = oneins.ins2.arg2; break;
        case 3: arg0 = oneins.ins3.arg0; arg1 = oneins.ins3.arg1; arg2 = oneins.ins3.arg2; break;
    }
    return ③;
}
① _____ ② _____ ③ _____

```

4. 下面是十进制超大整数加法运算程序中的几个函数，请完善代码。

```

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

```

```

struct intnode {

```

```

    int data;
    struct intnode *next;
};

/* 采用后进先出方式建立链表，存放输入的十进制超大整数 */
void inputdata( ① phd)
{
    char ch;
    struct intnode *p;

    while ((ch=getchar()) != '\n') {
        p = (struct intnode *)malloc(sizeof(struct intnode));
        p->data = ch - '0';
        p->next = ②;
        *phd = p;
    }
}

```

解答内容不得超过装订线

```

/* 超大整数的加法运算，采用先进先出方式创建和值链表 */
struct intnode *exadder(struct intnode *p1, struct intnode *p2)
{
    struct intnode *head, *tail, *p;
    int carry = 0;

    head = tail = (struct intnode *)malloc(sizeof(struct intnode));
    while (p1!=NULL && p2!=NULL) {
        tail->next = (struct intnode *)malloc(sizeof(struct intnode));
        tail = ③;
        carry += p1->data + p2->data;
        tail->data = carry % 10 + '0';
        carry /= 10;
        p1 = p1->next, p2 = p2->next;
    }

    p = p1!=NULL ? ④;

    while (p != NULL) {
        tail->next = (struct intnode *)malloc(sizeof(struct intnode));
        tail = tail->next;
        carry += p->data;
        tail->data = carry % 10 + '0';
        carry /= 10;
        p = p->next;
    }
}

```

```

if (carry) {
    tail->next = (struct intnode *)malloc(sizeof(struct intnode));
    tail = tail->next;
    tail->data = carry + '0';
}
tail->next = ⑤;
p = head->next;
free(head);
return p;
}
① _____ ② _____ ③ _____
④ _____ ⑤ _____

```

附表 1 运算符的优先级和结合性

运 算 符	结 合 性
() [] -> .	左结合
! ~ ++ -- + - * & (类型) sizeof	右结合
* / %	左结合
+ -	左结合
<< >>	左结合
< <= > >=	左结合
== !=	左结合
&	左结合
^	左结合
	左结合
&&	左结合
	左结合
?:	右结合
= += -= *= /= %= &= ^= = <<= >>=	右结合
,	左结合

注：同一行上各运算符具有相同的优先级，从上往下优先级递降。

附表 2 相关库函数声明

函数原型	功 能
int strlen(const char *s);	求字符串 s 长度（不计 ‘\0’ ）
char *strstr(char *s, const char *t)	在字符串 s 中查找第一次出现字符串 t 的位置，返回在 s 中第一次出现 t 字符串的位置，如果未找到则返回 NULL。
int strcmp(const char *s1, const char *s2);	按字典序比较字符串 s1 和 s2,s1<s2 返回负数，s1>s2 返回正数，两串一样返回 0
char *strcpy(char *dest, const char *src);	把 src 所指向的字符串复制到 dest
void malloc(unsigned size);	分配 size 字节的存储区，返回指向该区的指针，分配不成功，返回 NULL
void free(void *p);	释放 p 所指的内存
FILE *fopen(const char * fname, const char *mode);	以 mode 方式打开文件 fname, 成功打开，返回文件指针，否则返回 NULL
int fclose(FILE *fp);	关闭文件，使 fp 不再与具体文件相关联，成功，返回 0，否则返回非 0
int fgetc(FILE *fp);	从文件指针 fp 指向的文件中读取一个字符
char *fgets(char *str, int n, FILE *stream);	从指定的流 stream 读取一行，并把它存储在 str 所指向的字符串内
int fputc (int c, File *fp)	将字符 c 写到文件指针 fp 所指向的文件
size_t fwrite(const void *ptr, size_t size, size_t n, FILE *fp);	将内存中从 ptr 地址开始的数据往 fp 所指向的文件里写，写入到文件里的字节数为 size*n

<code>exit(int status);</code>	关闭所有打开的文件并终止程序
--------------------------------	----------------