

## 综合测试题二参考答案

### 一、选择题（本大题共 10 小题，每小题 1 分，共 10 分）

1. C 语言中用于结构化程序设计的三种基本结构是\_\_\_\_\_ **A** \_\_\_\_\_。

- A. 顺序结构、选择结构、循环结构      B. if、switch、break  
C. for、while、do-while                  D. if、for、continue

2. 下列关于 C 语言用户标识符的叙述中正确的是\_\_\_\_\_ **B** \_\_\_\_\_。

- A. 用户标识符中可以出现下划线和中划线  
B. 用户标识符中不可以出现中划线，但可以出现下划线  
C. 用户标识符中可以出现下划线，但不可以放在用户标识符的开头  
D. 用户标识符中可以出现下划线和数字，他们都可以放在用户标识符的开头

3. 有以下程序： main() {

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

编译时出现错误，以下叙述中正确的是\_\_\_\_\_ **D** \_\_\_\_\_。

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

4. 有以下程序段： int m=0, n=0; char c='a' ;

```
scanf("%d%c%d", &m, &c, &n); printf("%d, %c, %d\n", m, c, n);
```

若从键盘上输入：12A10<回车>，输出结果是\_\_\_\_\_ **A** \_\_\_\_\_。

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

5. 设有定义：double x;，则以下正确的输入语句是\_\_\_\_\_ **C** \_\_\_\_\_。

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

6. 有以下程序： void main() {

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

执行后的结果是\_\_\_\_\_ **B** \_\_\_\_\_。

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

7. 设已有定义: `char a[10], *p=a;` 下面的赋值语句中, 正确的是 C。
- A. `a[10]="Turbo C";`                      B. `a="Turbo C";`  
 C. `p="Turbo C";`                              D. `*p="Turbo C";`
8. 设有已定义: `int n=100, *p=&n, *q=p;` 则以下不正确的赋值语句是 C。
- A. `p=q;`                      B. `*p=*q;`                      C. `p=n;`                      D. `n=*q;`
9. 主调函数中已有定义: `int a[3][4], n;` 当使用函数调用语句: `fun(a, n);` 时, 被调函数 `fun()` 的形参说明表列正确的为 B。
- A. `fun(int a[3][], int n){...}`                      B. `fun(int a[][4], int n){...}`  
 C. `fun(int a[][] , int n){...}`                      D. `fun(int *p[4], int n){...}`
10. C 程序由函数组成, 关于 C 语言函数的说法错误的是 D。
- A. 除整型函数外, 其它类型函数必须在定义时给以类型说明。  
 B. 函数原型说明语句必须给出函数类型的说明。  
 C. 函数类型决定返回值的类型。  
 D. 函数返回值的类型就是 `return` 语句中的表达式值的类型。

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

1. 设 `a, x` 已都定义为 `double` 类型变量, 试将数学公式  $\left| \cos \frac{\pi}{2} \right|$  写成 C 表达式为:  
`fabs(cos(3.14159/2))` 或 `a=3.14159, fabs(cos(a/2))`
2. 设 `x, n` 已定义为 `double` 类型变量, 试将数学公式  $\frac{3x^n}{2x-1}$  写成 C 表达式为:  
`3*pow(x,n)/(2*x-1)`
3. 若有定义: `int a=7; float x=2.5, y=4.7;`  
 执行 `printf("%f\n", x+a%3*(int)(x+y)%2/4);` 的结果为: 2.5
4. 若有定义: `int a=3, b=4;`  
 执行 `printf("%d\n", (a++b, a+5, a/5));` 的结果为: 1
5. 若有定义: `int a=3, b=4, c=5;`  
 执行 `printf("%d\n", a*=b+c);` 的结果为: 27
6. 若有定义: `int p, a=5;`  
 执行 `if(p=a!=0)printf("%d\n", p);`  
     `else printf("%d\n", p+2);` 的结果为: 1
7. 若有定义: `int a[3][4]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}, (*p)[4]=a;`  
 执行 `printf("%d\n", *(*(p+1)+3));` 的结果为: 8

8. 某地刑侦大队对涉及 6 个嫌疑人的一桩疑案进行分析：

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

为解这道题，需要将案情的每一条写成逻辑表达式，如变量 A 表示作案，!A 表示不作案，从第 1 条到第 6 条依次用 cc1, ……，cc6 表示，请参照以下已经给出的单序号的赋值语句形式，填写其余 3 个双序号赋值语句中的逻辑表达式：

```
cc1 = (A||B); // A, B 至少有一人作案;  
cc2 = (A&&E)|| (A&&F)|| (E&&F); // A, E, F3 人中至少有 2 人参与作案;  
cc3 = !(A&&B) // A, D 不可能是同案犯;  
cc4 = (B&&C)|| (!B&&!C); // B, C 或同时作案，或与本案无关;  
cc5 = (C&&!D)|| (D&&!C); // C, D 中有且仅有一人作案;  
cc6 = D|| !E; //如果 D 没有参与作案，则 E 也不可能参与作案。
```

### 三、阅读程序题：请仔细阅读分析下列程序，写出运行结果

(本大题共 5 小题，每题 6 分，共 30 分)

1. #include <stdio.h>

```
void main() {  
    int a[4][4]={ {1, 2, -3, -4}, {0, -12, -13, 14}, {-21, 23, 0, -24},  
                  {-31, 32, -33, 0}};  
  
    int i, j, s=0;  
    for(i=0; i<4; i++) {  
        for(j=0; j<4; j++) {  
            if(a[i][j]<0) continue;  
            if(a[i][j]==0) break;  
            s += a[i][j];  
        }  
        printf("%d\n", s);  
    }  
}
```

**运行结果：**

```
3  
3  
26  
58
```

2. #include <stdio.h>

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

**运行结果：**

```
3 2  
2 3
```

```

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

```

运行结果:

```

1
2 4
3 6 9

```

```

4. #include <stdio.h>
void main() {
    int i=0, CountX=0, CountY=0;
    char str[80]="The Programming Language";
    while(str[i]!='\0') {
        if(str[i]>='a' && str[i]<='z')
            str[i]-=32;
        switch(str[i]) {
            case 'A': case 'E':
            case 'I': case 'O':
            case 'U': CountX++;break;
            default : CountY++;break;
        }
        i++;
    }
    printf("CountX=%d\n", CountX); printf("CountY=%d\n", CountY);
}

```

运行结果:

```

CountX=8
CountY=15

```

```

5. #include <stdio.h>
#include <malloc.h>
char *fun1(char *a, int start, int len) {
    char *ans; int i;
    ans=(char *)malloc(strlen(a)+1);
    for(i=start; i<start+len; i++)
        ans[i-start]=a[i];
    ans[len]='\0';
    return(ans);
}
void fun2(char *s, char *t) {
    while(*s!='\0') s++;
    while((*s++=*t++)!='\0');
}
void main() {
    char a[80]="ABCDEFGHJIJ", b[80]="12345678910", *p;
}

```

运行结果:

```

AB
AB12
BCD
BCD234

```

```

int i, j;
for(i=0, j=1; i<2; i++) {
    p=fun1(a, i, ++j);    printf("%s\n", p);
    fun2(p, fun1(b, i, j)); printf("%s\n", p);
}
}

```

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

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

```

#include <stdio.h>
#include <process.h>
void main() {
    FILE *fp; char ch;
    if((fp=fopen("file1.dat", "w"或"w+"或"wb"或"wb+"))==NULL)
        exit(0);
    while((ch=getchar())!=EOF)
        fputc(ch,fp) 或 fprintf(fp,"%c",ch); ;
    fclose(fp);
}

```

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

```

#include <stdio.h>
void main() {
    int a[ 11 或比 11 大的数 ]={1, 10, 20, 30, 40, 50, 60, 70, 80, 90};

    int x, i, p;
    scanf("%d", &x);
    for(i=0, p=10; i<10; i++)
        if(x<a[i]) {
            p=i;
            break ;
        }
    for(i=9; i>=p; i--)
        a[i+1]=a[i];
    a[i+1]=x 或 a[p]=x ;
    for(i=0; i<=10; i++)
        printf("%5d", a[i]);
    printf("\n");
}

```

## 五、程序设计题（共两个程序设计题，第1题25分，第二题15分，共40分）

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

```
#include <stdio.h>
#include <malloc.h>
#include <string.h>
#include <process.h>
struct student{           //结构体类型定义
    char sno[16], cno[12];
    float score;
    struct student *next; };
```

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

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

或数组形式编写。

- ② 编写一个字符串比较 str\_cmp() 函数，可采用数组或指针方法编写；设有字符串 s 与 t，若 s 大于 t 则返回正值，s 等于 t 则返回 0，s 小于 t 则返回负值。（5分）

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

或

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

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

```
struct student *create() { //建立链表函数，表尾插入算法
    struct student *head, *p1, *p2;
    head=NULL;
```

```

p1=p2=(struct student *)malloc(sizeof(struct student));
scanf("%s%s%f", &p1->sno, &p1->cno, &p1->score);
while(str_len(p1->sno)>1) {
    if(head==NULL) head=p1;
    else p2->next=p1;
    p2=p1;
    p1=(struct student *)malloc(sizeof(STRUCT STUDENT));
    scanf("%s%s%f", &p1->sno, &p1->cno, &p1->score);
}
p2->next=NULL; free(p1);
return(head);
}

```

或 采用表头插入方式、有序插入方式等编写

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

```

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

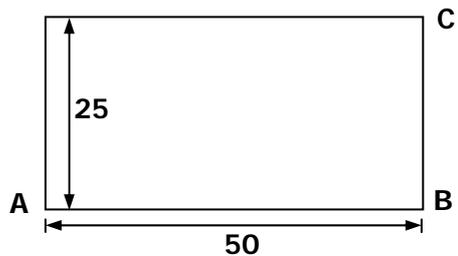
```

```

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

```

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



- ① 给出求解该问题的算法描述；（7分）
- ② 根据算法编写完整的求解程序（8分）

说明：距离的长度精度控制在 0.1 米或以下。

解：① 求解问题的算法描述：

本题采用枚举法（亦称穷举法）求解

分析：

1) 假设在 AB 边上行进  $x_1$  后的 D1 点下水，历时  $t_1=x_1/1.2$ ;

2) 计算从 D1 点到 C 点的距离和 D1 点到 C 点历时

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

3) 则从 A 点到 B 点间的 D1 点下水游到 C 点时间的为  $t=t_1+t_2$ ，显然，从 A 点到 B 点间的 D1 点下水游到 C 点最短计算如存在的话，肯定在 0 到 50 之间，而这之间  $D_i$  点在一定精度的步长内个数是有限的，所以可以设计一个基于枚举法的算法，设用  $\text{time}$  存储其最小值， $\text{dx}$  存储距 A 点的长度，其最短时间算法描述如下：

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

```
while(x<=50.0){
    temp=x/1.2
        +sqrt(WIDTH*WIDTH+(LENGTH-x)*(LENGTH-x))/0.8;
    if(若所求时间小于等于前次所求最短时间){
        更新保存最短时间和相应距离两变量的值;
    }
    x=x+0.1;          //距离增步进长 0.1，这里步进长精度为 0.1
}
```

② 根据算法编写完整的求解程序如下：

```
#include<stdio.h>
#include<math.h>
#define LENGTH 50.0
#define WIDTH 25.0
#define V1 0.8
#define V2 1.2
void main(){          //枚举法求解 1
    float x=0.0,time,temp,ax=0;
    time =32768;
    while(x<=50.0){
        temp=x/V2+sqrt(WIDTH*WIDTH+(LENGTH-x)*(LENGTH-x))/V1;
        if(temp<=time){
            time=temp;
            ax=x;
        }
        x=x+0.1;
    }
    printf("time min=%f, x=%f\n",time,ax);
    printf("(50+25)/1.2 =%f\n", (LENGTH+WIDTH)/V2);
}
```

注：此题解法甚多，视算法描述进行分析判分。