

2017-2018-II 《计算机程序设计基础 II》 参考答案

一、单项选择 ($20 \times 1.5 = 30$ 分)

DBADC BCDBA DCBDA CCBBD

二、填空 ($10 \times 1 = 10$ 分)

- 1、数组名
- 2、1、0
- 3、成员、箭头（或者指向）
- 4、point = &s; 、 s
- 5、(double *)
- 6、行优先
- 7、0 （或者 NULL）

三、程序分析 ($5 \times 5 = 25$ 分)

- 1、
2 4 5 6
- 2、
10101010
- 3、
ab
de
gh
e
- 4、
Return2
Return1
Return2
6
- 5、
A
E
G

H

四、程序设计($2 \times 10 + 15 = 35$ 分)

1、

```
#include <stdio.h>
#define N 20
int main( )
{
    int data[N] = {0}, i, j, temp;
    for (i = 0; i < N; i++)
        scanf("%d", data + i);
    for (i = 0; i < N - 1; i++)
    {   /* 递减冒泡排序 */
        for (j = 0; j < N - i - 1; j++)
        {
            if (data[j] < data[j + 1])
            {   /* 逆序则交换 */
                temp = data[j];
                data[j] = data[j + 1];
                data[j + 1] = temp;
            }
        }
    }
    for (i = 0; i < N; i++)
    {
        if (i && i % 5 == 0)
            printf("\n");
        printf("%d\t", data[i]);
    }
    return 0;
}
```

2、

```
#include <stdio.h>
int GCD(int m, int n);           /* 求最大公约数的函数原型 */
```

```
int LCM(int m, int n, int gcd);    /* 求最小公倍数的函数原型 */
```

```
int main( )
```

```
{
    int a, b, gcd, lcm;
    printf("请输入两个正整数，用空格分隔：");
    scanf("%d%d", &a, &b);
    if (a <= 0 || b <= 0)
    { /* 数据不正确，非正常退出 */
        printf("输入的数据必须是正整数!\n");
        return -1;
    }
    gcd = GCD(a, b);    /* 最大公约数 */
    lcm = LCM(a, b, gcd); /* 最小公倍数 */
    printf("最大公约数 %d，最小公倍数 %d\n", gcd, lcm);
    return 0;
}
```

```
int GCD(int m, int n)
```

```
{ /* 辗转相除求最大公约数 */
    int mid; /* 中间变量 */
    while (n != 0) /* 余数不为 0 时继续辗转相除 */
    {
        mid = m % n;
        m = n;
        n = mid;
    }
    return m;
}
```

```
int LCM(int m, int n, int gcd)
```

```
{ /* 利用最大公约数求最小公倍数 */
    return m / gcd * n;
}
```

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

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

```
struct node
```

```
{ /* 链表结点类型 */
    int data;
    struct node *next;
};
```

```
struct node *Create( ); /* 创建链表函数原型 */
void Output(struct node *head); /* 输出链表函数原型 */
int Count(struct node *head, int num); /* 查找函数原型 */
```

```
int main( )
```

```
{
    int n;
    struct node *head = NULL, *p;
    head = Create( );
    Output(head);
    scanf("%d", &n);
    printf("链表中共有 %d 个 %d\n", Count(head, n), n);
    while (head) /* 退出前释放所有结点空间 */
    {
        p = head;
        head = head->next;
        free(p);
    }
    return 0;
}
```

```
struct node *Create( )
```

```
{ /* 头插入法创建无头结点的链表 */
    struct node *head = NULL, *p;
    int num;
```

```
while (1)
{
    scanf("%d", &num);
    if (num < 0)
        break;
    p = (struct node *)malloc(sizeof(struct node));
    p->next = head;
    head = p;
}
return head;
}
```

```
int Count(struct node *head, int num)
{    /* 按次序计数 */
    int cnt = 0;
    while (head)
    {
        cnt += head->data == num;
        head = head->next;
    }
    return cnt;
}
```

```
void Output(struct node *head)
{    /* 输出链表结点信息 */
    while (head)
    {
        printf("%d  ", head->data);
        head = head->next;
    }
    printf("\n");
}
```