

计算机程序设计 20 -20 学年第一学期考试试卷 (A 卷) 参考答案:

一、单选题 (共 26 分, 前 20 题每题 1 分, 后四题每题 1.5 分)

(第 1-20 题是单选题, 每题 1 分)

1. B
2. D
3. D
4. B
5. A
6. D
7. C
8. B
9. C
10. D
11. C
12. C
13. D
14. B
15. B
16. C
17. C
18. B
19. A
20. A

(第 21-24 题为单选题, 每题 1.5 分)

21. A
22. B
23. C
24. B

二、多选题 (第 1-6 题为多选题, 每题 1.5 分)

1. ACD
2. ABCD
3. BCD
4. ABD
5. ACD
6. AB

三、单项填空（共 10 分，每空 1 分）

1. (1) 2
2. (2) 7
3. (3) a=(int) (a\*100+0.5)/100.0
4. (4) cdef
5. (5) 结构体（或结构）
6. (6) 200    (7) 100
7. (8) 5050
8. (9) 4    (10) 4

四、程序填空（共 30 分，每空 1.5 分）

1.
  - (1) `j>=m`
  - (2) `a[m]=x`
  - (3) `flag=1`
  - (4) `a[i]>x`
  - (5) `insert(a,N,i,x)`
2.
  - (6) `c=c-'A'+'a'` 或 `c=c+32`
  - (7) `c=c-'a'+'A'` 或 `c=c-32`
3.
  - (8) `struct student t`（或其它变量名）
  - (9) `j=0; j<n-i-1`
  - (10) `r[j]<r[j+1]` 或反过来
  - (11) `t=r[j],r[j]=r[j+1],r[j+1]=t;`（分号逗号都可以）
4.
  - (12) `n % 10`
  - (13) `n /= 10` 或 `(n= n/10)!=0` 或 `n=n / 10` 或 `(n /= 10)!=0`
5.
  - (14) `char **p, int n` 或 `char *p[5], int n`
  - (15) `i`
  - (16) `i+1`
  - (17) `name[i]` 或 `*(name+i)`
6.
  - (18) (1):`void` 替换为 `double`
  - (19) (2):`double s=0;`
  - (20) (6):`c+=credit[i++];`

## 五、函数编程（共 25 分）

编程 1 参考答案：（此题解法不唯一，定义变量和输入 1 分，处理过程 3 分，输出 1 分）

```
int main()    {
    int price; //存放商品价格的变量
    int num_1=0,num_5=0,num_20=0,num_50=0,num=0;
    scanf("%d", &price);
    while(price>=50 && num_50<=3) { price-=50;num_50++; }
    while(price>=20 && num_20<=4) { price-=20;num_20++; }
    while(price>=5 && num_5<=2) { price-=5; num_5++; }
    while(price>=1 && num_1<=2) { price-=1; num_1++; }
    num=num_1+num_5+num_20+num_50;
    if(price==0) {
        printf("最少需要%d 张，其中 50 元%d 张，20 元%d 张，5 元%d 张，1 元%d 张\n", num, num_50, num_20, num_5, num_1);    }
    else printf("无法打印!\n");
    return 0; }
```

可能的问题：程序中 N20 之类的宏定义换成对应的数字也可以，不会影响结果。

编程 2 参考答案：（共 6 分，每错或少一条语句扣 0.5 分）

```
void replace(char *src, char dest[], char *sub, char *word){
    int i,j,k,len_src,len_sub,len_word;
    char tmp[100]; //用于存放从 src 的下标 i 开始，与 word 等长子串
    i=0; //用 i 扫 src 串
    k=0; //用 k 扫 dest 串
    len_src=strlen(src);
    len_sub=strlen(sub);
    len_word=strlen(word);
    while(i<len_src) {
        for(j=0;j<len_word;j++) { //从 src 串的下标 i 开始，取与 word 等长子串
            tmp[j]=src[i+j];
        }
        tmp[j]='\0';
        if(strcmp(tmp,word)==0) { //匹配上，直接将 sub 复制到 dest
            strcpy(dest+k,sub);
            i+=len_word;
            k+=len_sub;
        }
        else{ //没匹配上，将 src[i]复制到 dest
            dest[k++]=src[i++];
        }
    }
}
```

3-1 struct student \*create() { //共 4 分, 每错或少一条语句扣 0.5 分

```
int i=0;    struct student *p,*head,*rear;
head=NULL;
while (i<20) {
    p=(struct student *)malloc(sizeof(struct student));
    scanf("%d",&p->Num);
    scanf("%f",&p->score);
    if(head==NULL) head=p;
    else rear->next=p;
    rear=p;
    i++;    }
if(rear!=NULL)    rear->next=NULL;
return head;
}
```

3-2 float average(struct student \*head) { //共 2 分, 每错或少一条语句扣 0.5 分

```
int num=0; struct student *p = head;
float ave = 0;
while (p) {
    ave += p->score;
    p = p->next;
    num++;}
return ave / num;
}
```

3-3

struct student \*excellent(struct student \*head, float ave) { //8 分  
//head 为链表头, ave 为学生平均成绩

//补充本函数代码, 实现功能: 从以上链表中找出所有超过平均成绩的优秀学生, 按成绩从高到低的顺序创建一个新链表, 链表头指向成绩最高的学生, 并返回新链表头。

```
float swapscore;
int count=0,i,j,swapnum;
struct student *p = head, *q, *newhead, *q1, *q2;
newhead = NULL;
while(p) { //头插法建立一个无序的链表
    if ( ave < p->score) {
        q = (struct student *)malloc(sizeof(struct student));
        q->score = p->score;
        q->Num = p->Num;
        q->next = newhead;
        newhead = q;
        count++;
    }
    p = p->next;
}
```

```

for(i=1;i<count;i++) { //冒泡排序链表
    q1=newhead;
    for(j=0;j<count-i;j++) {
        q2=q1->next;
        if(q1->score < q2->score) {
            swapnum=q1->Num;
            q1->Num=q2->Num;
            q2->Num=swapnum;
            swapscore=q1->score;
            q1->score=q2->score;
            q2->score=swapscore;
        }
        q1=q1->next;
    }
}
return newhead;
}

```

另一种实现:

```

struct student *excellent(struct student *head, float ave) {
//共 8 分，每错或少一条语句扣 0.5 分
    struct student *p = head, *q,*pre, *newnode,*newhead;
    newhead = NULL;    pre = NULL;
    while (p) {
        if ( ave < p->score) { //筛选出高于均分的学生
            newnode = (struct student *)malloc(sizeof(struct student));
            newnode->score = p->score;
            newnode->Num = p->Num;
            newnode->next = NULL; //这一句也可以放到尾插部分
            //newnode 节点继承该学生信息，有序插入新链表
            q = newhead;
            while (q) { //用 q 遍历新链表，找插入点
                if ( newnode->score < q->score) {
                    pre = q; //pre 为 q 前一节点
                    q = q->next;
                }
                else break;
            }
            if (q == newhead) { //新链表为空或插入在第一个结点前
                newnode->next = newhead;
                newhead = newnode;
            }
        }
        p = p->next;
    }
    return newhead;
}

```

```
        else if (!q){ //插入在最后一个结点后
            pre->next = newnode;
        }
        else{ //插入位置在中间
            newnode->next = q;
            pre->next = newnode;
        }
    }
    p = p->next;
}
return newhead;
}
```

可能出现问题：尾插时没有将插入节点的 next 置为 NULL