

22级C语言期末考试A卷

编程题 (总分: 100.00)

1. 邮件收费 (分值: 20.00)

【问题描述】

假设邮局规定寄邮件时若每件重量在1公斤以内(含1公斤), 按2.5元计算邮费, 如果超过1公斤时, 其超出部分每公斤收1.8元。请编程序计算邮件收费。

【输入形式】

公斤数。

【输出形式】

邮件收费, 保留2位小数

【样例输入】

5

【样例输出】

9.70

【样例说明】

【评分标准】

参考答案:

```
#include<stdio.h>
int main()
{
    double a,s=0;
    scanf("%lf", &a);

    if(a<=1 && a > 0)
    {
        s=2.5;
    }
    else if(a>1){
        s= 2.5 + (a-1)*1.8;
    }
    printf("%.2f",s);
    return 0;
}
```

2. 编写程序，输入n,根据公式计算s的值(阶乘) (分值：25.00)**【问题描述】**

编写程序，输入n(n<10),根据公式计算s的值，要求输出5位小数。

$$s = \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{n!}$$

【输入形式】

n

【输出形式】

s

【样例输入】

4

【样例输出】

1.70833

【样例说明】**【评分标准】**

参考答案:

```
#include<stdio.h>
int main()
{
    int i,j,n;
    double s=0;
    int sum = 1;

    scanf("%d", &n);

    for(i=1;i<=n;i++)
    {
        // for(j=1;j<=i;j++)
        //{
        //     sum= sum*j;
        //}
        sum= sum*i;
        s=s+1.0/sum;
    }
    printf("%.5f",s);
    return 0;
}
```

3. 数组选择 (最小) (分值：25.00)**【问题描述】**

将任意 n ($n < 10$) 个正整数输入一维数组，最小数放到最后面，，其他顺序不变输出。

【输入形式】

n

数组

【输出形式】

数组

【样例输入】

6

5#1#2#7#8#4

【样例输出】

#5#2#7#8#4#1

#代表空格，输出时输出空格不要输出#。

【样例说明】

【评分标准】

参考答案:

```
#include<stdio.h>
int main()
{
    int max,num=0,i,n;
    int a[1000];
    max = 10000;
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
        if(max>a[i])
        {
            max=a[i];
            num = i;
        }
    }
    for(i=num;i<n;i++)
    {
        a[i] = a[i+1];
    }
    a[n-1] = max;

    for(i=0;i<n;i++)
    {
        printf(" %d",a[i]);
    }
    return 0;
}
```

4. 字符串查找 (分值: 15.00)**【问题描述】**

字符串查找。输入字符串A，输入某子字符串B。输出B在A中出现的次数。没有找到输出0. 可以使用strlen函数。

【输入形式】

字符串A

字符串B

【输出形式】

数量

【样例输入】

123abc456789abc67

abc

【样例输出】

2

【样例说明】**【评分标准】**

参考答案:

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

/*字符串寻找子串 -- 简单版*/

int find(char parents[], char son[])
{
    //最终返回子串第一次所在的索引位置
    int count = 0;
    int len1 = strlen(parents);
    int len2 = strlen(son);

    for (int i = 0; i < len1; i++)//最多进行len1次比较
    {
        int j,k = i;
        for (j = 0; j < len2; j++)
        {
            //子串一直比较到结尾如果都符合就返回比较第一次的值
            if (parents[k++] == son[j])
            {
                continue;
            }else{
                break;
            }
        }
        if (j == len2)
            count++;
    }

    return count;//找不到就返回 - 1
}

int main() {
    /*测试数据*/
    char a[100] ;
    char b[100];
    gets(a);
    gets(b);
    printf("%d", find(a, b));

    return 0;
}
```

5. 删除排序链表中的重复元素 (分值: 15.00)

【问题描述】

给定一个**已排序**的链表的头 head，删除所有重复的元素，使每个元素只出现一次。返回已排序的链表。说明：

部分代码已给出，只实现deleteDuplicates函数完成上述要求即可。注意：链表创建时，输入0结束。

以下代码可以直接使用，链表创建和输出已经实现。

```
#include <stdio.h>

#include <string.h>

#include <stdlib.h>

struct node

{

    int num;

    struct node* next;

};

//创建函数

struct node* createList();

//打印函数

void printList(struct node* head);

int main(){

    struct node* head=NULL;

    head = createList();

        // 这里调用deleteDuplicates函数

    printList(head);

    return 1;

}

struct node* createList(){

    //头指针

    struct node* head=NULL;

    struct node* p1,*p2;

    int num;

    scanf("%d",&num);

    while(num!=0)
```

```
{
//创建一个新的结点
p1 = (struct node* ) malloc(sizeof(struct node));
//把信息拷贝到新的结点
p1->num = num;
if(head==NULL){
    head = p1;
head->next=NULL;
}
else{
p2=head;
while(p2->next!=NULL)
{
p2=p2->next;
}
p2->next = p1;
p1->next = NULL;
}
scanf("%d",&num);
}
return head;
}
```

```
void printList(struct node* head){
struct node* p;
if(head==NULL)
{
printf("Empty List\n");
return;
}
else{
p=head;
while(p!=NULL)
{
printf("%d",p->num);
```

```
p=p->next;  
}  
}  
  
}
```

【输入形式】

已排序的链表

【输出形式】

无重复的链表

【样例输入】

1#1#2#0

【样例输出】

#1#2

ps:#代表空格，输出时输出空格而不是#。

参考答案:


```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
struct node
{
    int num;
    struct node* next;
};

//打印函数
void printList(struct node* head);
struct node* createList();
struct node* deleteDuplicates(struct node* head);
int main(){

    //头指针
    struct node* head=NULL;
    head = createList();
    head = deleteDuplicates(head);
    printList(head);
    return 1;
}

struct node* createList(){
    //头指针
    struct node* head=NULL;
    struct node* p1,*p2;
    int num;
    scanf("%d",&num);
    while(num!=0)
    {
        //创建一个新的结点
        p1 = (struct node* ) malloc(sizeof(struct node));
        //把信息拷贝到新的结点
        p1->num = num;
        if(head==NULL){
            head = p1;
            head->next=NULL;
        }
        else{
            p2=head;
            while(p2->next!=NULL)
            {
                p2=p2->next;
            }
            p2->next = p1;
            p1->next = NULL;
        }
        scanf("%d",&num);
    }
    return head;
}
```

```
void printList(struct node* head){
    struct node* p;
    if(head==NULL)
    {
        printf("Empty List\n");
        return;
    }
    else{
        p=head;
        while(p!=NULL)
        {
            printf(" %d",p->num);
            p=p->next;
        }
    }
}

struct node* deleteDuplicates(struct node* head){
    //空链表就直接返回空指针
    if(!head) // 注：在条件判断中，head == NULL 和 !head 所起的作用是一样的。
        return NULL;

    struct node* cur = head;
    while(cur->next && cur) // 注意循环条件
    {
        struct node* next = cur->next;
        //若相等，则删掉下一个节点
        if(cur->num == next->num)
        {
            cur->next = next->next;
            free(next);
        }
        //若不等，则将下一个节点变为当前节点
        else
            cur = cur->next;
    }

    return head;
}
```