

高级语言程序设计 2025.12 第二次期中考试 试题目和答案

题目 1

参考题面：实现函数 findMinMax，传入数组 b、变量 maxIndex 的指针（用于存储最大值下标），找到数组 b 中的最大、最小值，将所传入数组的最大值下标赋值给 maxIndex，并返回最小值下标。

```
#include <stdio.h>

int findMinMax(int b[],int *maxIndex); /*最大值的下标通过指针带回，最小值的下标通过return返回 */

int main()
{
    int a[8];
    int min,max,i; //min是最小值的下标，max是最大值的下标

    for(i=0;i<8;i++)
        scanf("%d",&a[i]); /*数不会重复，即没有相同的数*/

    /* 调用函数 findMinMax */

    min = findMinMax(a, &max);
    printf("%d %d",min,max);

    return 0;
}

/* 完成函数 findMinMax 的实现，最大值的下标通过指针带回，最小值的下标通过return返回 */
int findMinMax(int b[],int *maxIndex)
{
    int minIndex, min = 12345678, max = -87654321;
    for (int i = 0; i < 8; i++) {
        if (b[i] > max) {
            max = b[i];
            *maxIndex = i;
        }
        if (b[i] < min) {
            min =b[i];
            minIndex =i;
        }
    }
}
```



```
    return minIndex;
}
```

题目 2

输入数字 n ，并依次输入三个长度为 n 、 $n+1$ 、 $n+2$ 的字符串，动态开辟内存空间来存储这三个字符串，并依次输出存入的字符串、及其第一个字母。

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

int main()
{
    char *p[3];
    /* 定义需要的变量，完成程序 */
    int n;
    scanf("%d", &n);
    for (int i = 0; i < 3; i++) {
        p[i] = (char*) malloc(sizeof(char) * (n + 1));
        scanf("%s", p[i]);
        n++;
    }
    for (int i = 0; i < 3; i++) {
        printf("%s %c\n", p[i], *p[i]);
    }
    return 0;
}
```

题目 3

输入电影数量 n ，及 n 个电影的 $name_i$ 、 $income_i$ ，按照电影票房 $income$ 从大到小排序，并依次输出，格式为 `%d %20s %4d`。

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

typedef struct
{
    char name[20];
    int ticketIncome;
}Film;

void inputValue(Film f[],int n);
void outputValue(Film f[],int n);
```



```

void sort(Film f[],int n);

int main( )
{
    int n;
    Film film[10];
    scanf("%d",&n);
    // 请在此编写你的程序
    inputValue(film, n);
    sort(film,n);
    outputValue(film, n);
    return 1;
}
void inputValue(Film f[],int n){
    for (int i = 0; i < n; i++) {
        scanf("%s %d", &f[i].name, &f[i].ticketIncome);
    }
    return;
}
void outputValue(Film f[],int n){
    for (int i = 0; i < n; i++) {
        printf("%d %20s %4d\n", i+1, f[i].name, f[i].ticketIncome);
    }
}
void sort(Film f[],int n){
    for (int i = 0; i < n; i++) {
        for (int j = i + 1; j < n; j++) {
            if (f[i].ticketIncome < f[j].ticketIncome) {
                Film temp = f[i];
                f[i] = f[j];
                f[j] = temp;
            }
        }
    }
    return;
}
}

```

题目 4

实现 create、print 函数。create 函数要求从输入使用尾插法创建链表，print 函数要求输出所传入的链表。

```

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

typedef struct node
{
    int data;
    struct node *next;
}NODE;

```



```

NODE *create();
void print(NODE *h);

int main( )
{
    NODE *h1,*h2=NULL;
    /*定义需要的变量*/

    h1 = create();
    print(h1);

    /*拆分链表，第五关在此完成，可以自己封装函数，也可以直接在这里写*/
    //第五关请将下面两行代码解除注释

    //print(h1);
    //print(h2);

    return 0;
}

NODE * create() /*用尾插法建链表，输入-1结束 */
{
    /*把程序补充完整 */
    NODE *h = NULL,*p,*r; // h = head, p = newNode, r = last
    int x;
    h=NULL;
    while(1){
        scanf("%d",&x);
        if (x == -1) return h;
        p = (NODE*) malloc(sizeof(NODE));
        p->data = x;
        p->next = NULL;
        if (h == NULL) {
            h = p;
            r = p;
        } else {
            r->next = p;
            r = p;
        }
    }
    return h;
}

void print(NODE *h) /* 若链表为空，打印NULL，不空输出格式用%3d，链表输出完后换行*/
{
    if (h == NULL) {
        printf("NULL\n");
        return;
    }
    for (; h != NULL; h = h->next) {

```



```

        printf("%3d", h->data);
    }
    printf("\n");
    return;
}

```

题目 5

在题目 4 的基础上，在指定位置完成拆分链表的操作。要求将 h1 链表中值为奇数的项目移动到 h2 链表中，使用头插法。

例如：h1 = 1 2 3 4 5，则处理后 h1 = 2 4、h2 = 5 3 1。

```

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

typedef struct node
{
    int data;
    struct node *next;
}NODE;

NODE *create();
void print(NODE *h);

int main( )
{
    NODE *h1,*h2=NULL;
    /*定义需要的变量*/

    h1 = create();
    print(h1);

    /*拆分链表，第五关在此完成，可以自己封装函数，也可以直接在这里写*/
    NODE *pre = NULL;
    for (NODE *cur = h1; cur != NULL;) {
        if (cur->data % 2 == 1) {
            if (pre != NULL) {
                pre->next = cur->next;
            } else {
                h1 = cur->next;
            }
            NODE *next = cur->next;
            cur->next = h2;
            h2 = cur;
            cur = next;
        } else {
            pre = cur;
            cur = cur->next;
        }
    }
}

```



```

    }
}

//第五关请将下面两行代码解除注释

print(h1);
print(h2);

return 0;
}

NODE * create() /*用尾插法建链表, 输入-1结束 */
{
    /*把程序补充完整 */
    NODE *h = NULL,*p,*r; // h = head, p = newNode, r = last
    int x;
    h=NULL;
    while(1){
        scanf("%d",&x);
        if (x == -1) return h;
        p = (NODE*) malloc(sizeof(NODE));
        p->data = x;
        p->next = NULL;
        if (h == NULL) {
            h = p; r = p;
        } else {
            r->next = p;
            r = p;
        }
    }
    return h;
}

void print(NODE *h) /* 若链表为空, 打印NULL, 不空输出格式用%3d, 链表输出完后换行*/
{
    if (h == NULL) {
        printf("NULL\n");
        return;
    }
    for (; h != NULL; h = h->next) {
        printf("%3d", h->data);
    }
    printf("\n");
    return;
}

```