

浙江大学 2014 - 2015 学年夏季学期

《C 程序设计专题》课程期末考试试卷

课程号: 211Z0050, 开课学院: 计算机学院

考试试卷: A 卷、B 卷 (请在选定项上打 \checkmark)

考试形式: 闭、开卷 (请在选定项上打 \checkmark), 允许带 / 入场

考试日期: 2015 年 07 月 03 日, 考试时间: 120 分钟

诚信考试, 沉着应考, 杜绝违纪.

考生姓名: _____ 学号: _____ 所属院系: _____

(注意: 答题内容必须写在答题卷上, 写在本试题卷上无效)

Section 1: single Choice(2 marks for each item, total 14 marks)

- Suppose that five numbers are pushed into a stack in the order of **1, 2, 3, 4, 5**, and the first popped number is **4**. What is the possible last popped number? ____.
A. 1 B. 5 C. 1 or 5 D. 1 or 3
- In a single linked list with **N** nodes, which operation requires the time complexity of **$O(N)$** ? _____. (Given: the time complexity of the operation is linear. 操作时间复杂性是线性的)
A. to find the ***i*-th** node in the list ($1 \leq i \leq N$)
B. to insert a new node after the node pointed by ***p***
C. to arrange the nodes of the list in increasing order
D. to delete a node after the node pointed by ***p***
- To merge two sorted linked lists both of ***n*** nodes and return a new sorted list. What is the minimum number of compare times (最少比较次数) during the merge? _____.
A. 1 B. ***n*** C. **$2n$** D. **$n \log n$**
- Given the following definitions, which one is the correct reference (引用)? _____.

```
struct {  
    int a;  
    float b;  
} data, *p=&data;
```


A. **$(*p).data.a$** ; B. **$(*p).a$** ; C. **$p->data.a$** ; D. **$p.data.a$** ;
- For the following recursive function, the return value of function call **$f(4)$** is _____.

```
int f(int n)  
{  
    return f(n-1)+n;  
}
```


A. 10 B. 11 C. 0 D) None of the above

6. After executing the following code fragment, the value of variable **z** is ____.
- ```

static struct {
 int x, y[3];
} a[3]={{1,2},{3,4,5},{6,7,8,9}}, *p=a+2;
int z;
z=((int *)p-2);

```
- A. 5                      B. 1                      C. 7                      D. None of the above
7. Given the definitions:
- ```

void f1();
void (*pf)();
void f2(void (*p)(), int x);

```
- In the following statements, ____ is **WRONG**.
- A. pf = f1; (*pf)(); B. pf = f1; f2(pf, 2); C. pf = f2; (*pf)(); D. f2(f, 3);

Section 2: Read the following problems and answer questions (6 marks for each item, total 30 marks)

1. Write down the definitions:
- (1) Use typedef to define a new type name **STRPA** _____ and make it denote(表示) a pointer array of with **10** elements each of which points to a character(10 个元素的字符指针数组).
- (2) For the function: **int f(void *p, double a, double b)**;, define a variable **pf** _____ to which can be assigned with the value **f**.

2. The following program will output _____.

```

#include <stdio.h>

void printa(int n)
{
    if(n/10) printa(n/10);
    putchar(n%10+'0');
}

void printb(int n)
{
    putchar(n%10+'0');
    if(n/10) printb(n/10);
}

main()
{
    printa(123);
    printb(123);
}

```

3. The following program will output _____.

```

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

main()
{
    static char t[]="REDCAP";
    void fun(char *t);

    fun(t);
}

```

```

    printf(t);
}

void fun(char *t)
{
    char ch;
    int i, j, n;

    n = strlen(t);
    for (i = 1; i < n; i++) {
        ch = t[i];
        for (j = i-1; j >= 0 && ch < t[j]; j--) t[j+1] = t[j];
        t[j+1] = ch;
    }

    return;
}

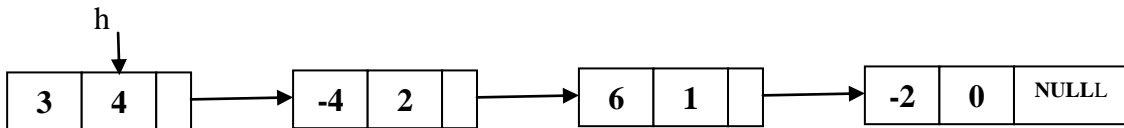
```

4. Given the definition of a linked list and a practical example (linked list **h**), please draw up (画出) the result of the linked list **h** after calling the function **h=f(h)**.

```

struct node{
    int coe;
    int exp;
    struct node *next;
};
typedef struct node ListNode;

```



```

ListNode *f(ListNode *head)
{
    ListNode HEAD, *cur, *pre;
    HEAD.next = head;
    cur = head;
    pre = &HEAD;
    if (!head) return NULL;
    while (cur) {
        if (cur->coe < 0) {
            pre->next = cur->next;
            free(cur);
            cur = pre->next;
        } else {
            pre = cur;
            cur = cur->next;
        }
    }
    return HEAD->next;
}

```

5. Merge two sorted linked lists and return it as a new sorted list. The new list should be made by splicing (拼接) together the nodes of the original two lists. The sorted list is in ascending order (升序). For the sorted linked list **I1** and **I2**
- I1: 1->3->5->7->9**
I2: 2->4->6

- (1) Following is a wrong program. According to this program, how is the merged linked list look like after calling **mergeTwoLists(l1,l2)**?
- (2) Why is the program wrong? (Explain the reason. No need the correct program.)

```

ListNode *mergeTwoLists(ListNode *l1, ListNode *l2)
{
    ListNode HEAD;
    ListNode *cur = &HEAD;

    while (l1 && l2) {
        if (l1->val > l2->val) {
            cur->next = l1;
            l1 = l1->next;
        } else {
            cur->next = l2;
            l2 = l2->next;
        }
        cur = cur->next;
    }
    return HEAD.next;
}

```

Section 3: According to the specification, complete each program (3 marks for each blank, total 30 marks)

1. Given a sorted linked list in ascending order (升序), to insert a node with data **x** to this linked list and keep the order ascending.

```

ListNode *Insert( ListNode *head, int x)
/*insert a node of x to an ascending order list head*/
{
    ListNode *p, *p1=head, *p2;

    p=(ListNode *)malloc(__(1)__);
    p->data=x;
    p->next=NULL;
    while (p1 && p->data>p1->data) {
        p2=p1;
        p1=__(2)__;
    }
    if (p1==head) head=__(3)__;
    else p2->next=p;
    p->next=p1;
    return __(4)__;
}

```

2. Given the following functions which are related with the timer(定时器) in the graphics library of our course:

```

typedef void (*TimerEventCallback) (int timerID);
void registerTimerEvent(TimerEventCallback callback);
void startTimer(int timerID, int timeinterval);
void cancelTimer(int timerID);

```

The code fragment(代码片段) below will implement the timing (定时) task with 5 seconds interval. The string "**hello**" will be displayed ONLY ONCE at the first 5 seconds interval after starting the timer, then no more timer events happen.

Please fill in the blanks.

```
void timerEventCallback(int timerID)
{
    printf("hello\n");
    if ( timerID == 0 ) {
        _____(5)_____;
    }
}

void SetUp()
{
    .....
    registerTimerEvent(_____(6)_____);
    _____(7)_____;
    .....
}
```

3. The following code fragment will implement the task that executes the function **a**, **b** and **c** according to the number (from **0** to **2**) input by the user, with **0** for **a**, **1** for **b**, and **2** for **c**.

Please fill in the blanks to complete the code fragment.

NOTE: only **ONE** statement or expression can be written down in a blank.

```
void a(void);
void b(void);
void c(void);

void cmd()
{
    void (*CMDS[])(void) = {
        _____(8)_____
    };
    int k;

    scanf("%d", &k);
    if ( k >=0 && k < _____(9)_____ )
    {
        _____(10)_____;
    }
}
```

Section 4: Algorithms design (13 marks for each item, total 26 marks)

1. The recursive function **void ListPermutations(int a[],int n)** will implement the tasks that list all the permutations(排列) with the first **n** objects of array **a**, and print out these permutations.

For example, for definition **int a[5]={11,22,33,44,55}**, the function call

ListPermutations(a, 3);

will output:

```
11 22 33
11 33 22
22 11 33
22 33 11
33 22 11
33 11 22
```

Please write down the recursive function **ListPermutations**.

2. A rational number(有理数) is one that can be expressed as the quotient of two integers. Thus, the number 1.25 is a rational number because it is equal to 5 divided by 4. A rational number $r=num/den$ can be represented as a structure of these two integers, thus we can define a rational type *rationalT*. We are now going to write a program to calculate the average (平均数) of a serial rational numbers $r1, r2, r3, \dots$. For example the average of $1/2, 1/6, 3/6, -5/10$ is $1/6$. Here, $10/60$ should be simplified to $1/6$.

(1) Give the declaration of rational type *rationalT*;

(2) Give the function *rationalT simplifyRational(rationalT r)*, to simplified a rational number to its lowest term, such as simplified $10/60$ to $1/6$. (有理数化简函数)

(3) Give the function *rationalT rationalsAverage(rationalT r[], int n)*, to calculate the average of n rational numbers stored in an array r .