

浙江大学 2018 - 2019 学年夏学期

《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)

考试形式: 闭、开卷 (请在选定项上打)，允许带 / 入场

考试日期: 2019 年 06 月 26 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 20 marks)

- If the queue "Q" contains "D E F" after the call sequence: Enqueue(Q, "E"), Dequeue(Q), Dequeue(Q), Enqueue(Q, "F"), which of the following is NOT the possible contents of "Q" before these calls _____.
A. "C D E" B. "B C D" C. "F E D" D. "E F D"
- Which one of the following statements is NOT true for a single linked list? _____.
A. Storage is proportional to the number of the elements.
B. Memory can be dynamically allocated.
C. Elements can be randomly accessed(随机访问)
D. Nodes don't need to be moved when insert a new element.
- For storage-class specifiers, which one below is NOT correct? _____.
A. At most, one storage-class specifier may be given in the declaration specifiers in a declaration.
B. A static global variable is not accessible by other compile units.
C. A static function is accessible by other static functions in different compile units.
D. "static int *p;" means that p is a static pointer to an integer.
- The output of the following program is _____.

```
#include <stdio.h>
f (int x)
{
    if(x<=1) return x;
    return f(x-1)+f(x-2);
}
int main( )
{ printf("%d", f(f(4))); }
```


A. 2 B. 3 C. 5 D. None of the above
- Inserting a node into a descending-order(降序) linked list with N nodes needs _____ comparisons at average.
A. O(1) B. O(N^2) C. O(NlogN) D. O(N)
- Given a data set of N(N=10^6) integers which is within the range of the whole integers, and unsorted, most of the data are duplicated except one. Given a good

sort function which has an complexity of $O(N\log N)$, to find out the single one, the complexity of the best algorithm is _____.

- A. $O(N)$ B. $O(N^2)$ C. $O(N\log N)$ D. $O(N\log N+N)$
7. The output of the following program will be _____.
 A. 8 B. 7 C. 6 D. 5

```
#include <stdio.h>
struct abc {
    int a,b,c;
};
int main()
{
    struct abc arr[2] = {{1,2,3},{4,5,6}}, *p = arr + 1;
    int t;
    t = p->b + (p-1)->a;
    printf("%d",t);
}
```

8. Given the following struct and variable definitions:

```
struct IDCard {
    char address[200];
    char name[20];
    int id;
}
```

```
card, *p = &card;
```

the correct expression that prints out the "address" member of the variable "card" is _____.

- A. `printf("%s\n", *p.address)` B. `printf("%s\n", (*p).address)`
 C. `printf("%s\n", *p->address)` D. `printf("%s\n", p->address)`

9. What is the output of the following program?

```
#include <stdio.h>
void print_integer(int *x) { printf("%d ", ++*x);}
int main()
{
    int a = 10;
    print_integer(&a);
    printf("%d", a);
}
```

- A. 10 10 B. 11 10 C. 10 11 D. 11 11

10. If we define function f like "int f(int a){return a;}". For statement "printf("%d", g(1, f));", which one below is NOT correct? _____.

A.	<code>int g (int c, int (*f) (int)) { return f(c); }</code>	B.	<code>int g (int c, int (*f) (int)) { return (*f)(c); }</code>
C.	<code>int g (int c, int f (int)) { return (&f)(c); }</code>	D.	<code>int g (int c, int f (int)) { return f(c); }</code>

Section 2: Read the following problems and answer questions (5 marks for each item, total 30 marks)

1. Write down your answers.

(1) Will the following program fragment cause any compile error? If so, briefly explain the cause of the error. _____

```
struct vector {
    double x, y, z;
};
```

vector.x = vector.y = vector.z = 0;

(2) Write down the definition of a complex type T in order to define a pointer p like T p, where p is a pointer to a function expecting (char *,double) and returning a pointer to an integer.

2. Given the definition of a student information record, running the following program on a 32-bit system will output _____.

```
#include <stdio.h>
#include <malloc.h>
typedef struct{
    char id[12];
    char name[20];
    void* photo;
} StudentInfo, *PtrStudentInfo;
int main()
{
    PtrStudentInfo pStudent = (PtrStudentInfo)malloc(sizeof(StudentInfo));
    pStudent->photo = malloc(100*100*3);
    printf("%d %d %d %d", sizeof(StudentInfo), sizeof(PtrStudentInfo),
        sizeof(pStudent->name), sizeof(pStudent->photo));
    .....
    free(pStudent->photo);
    free(pStudent);
}
```

3. If a linked list A is : 6->5->8->2->3->7->4, what will be the node returned by the following function fun(A)? _____

```
typedef struct _ListNode{
    int val;
    struct _ListNode *next;
} ListNode;
ListNode* fun(ListNode* head)
{
    ListNode *p1, *p2;
    if(head == NULL) return NULL;
    p1 = p2 = head;
    while(p1!=NULL && p1->next!=NULL) {
        p1=p1->next->next; p2=p2->next;
    }
    return p2;
}
```

4. Given the following code fragment and two simple linked lists below:

L1: 1->3->5->7->9

L2: 2->4->6

After calling L1=opTwoLists(L1, L2), the list L1 is _____.

```
typedef struct _ListNode ListNode;
struct _ListNode {
    int val;
    ListNode *next;
};
ListNode *opTwoLists(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;
}

```

5. What is the output of the following program? _____

```

#define MaxSize 10
struct DStack {
    int Data[MaxSize];
    int Top1;
    int Top2;
} S;
void Push( struct DStack *PtrS, int item, int Tag )
{ if ( PtrS->Top2 - PtrS->Top1 == 1) { printf("FULL"); return ; }
  if ( Tag == 1 ) PtrS->Data[++(PtrS->Top1)] = item;
  else PtrS->Data[--(PtrS->Top2)] = item;
}
int Pop( struct DStack *PtrS, int Tag )
{ if ( Tag == 1 ) {
    if ( PtrS->Top1 == -1 ) { printf("Empty1!"); return 0; }
    else return PtrS->Data[(PtrS->Top1)--];
  } else {
    if ( PtrS->Top2 == MaxSize ) { printf("Empty2!"); return 0; }
    else return PtrS->Data[(PtrS->Top2)++];
  }
}
int main()
{
    S.Top1 = -1;
    S.Top2 = MaxSize;
    Push(&S,1,1);
    Push(&S,2,1);
    Push(&S,3,2);
    printf("%d#",Pop(&S,2));
    printf("%d#",Pop(&S,2));
    printf("%d#",Pop(&S,1));
    printf("%d#",Pop(&S,1));
}

```

6. The following program will output _____.

```

#include <stdio.h>
void swap(int a[], int i, int j)
{
    int t = a[i]; a[i] = a[j]; a[j] = t;
}
int fun(int a[], int n, int k)
{
    int i,j,l,m,x;
    l=0 ; m=n-1 ;
    while (l<m) {
        x=a[k] ;
        i=l ;
        j=m ;
        do {
            while (a[i]<x) i++ ;
            while (x<a[j]) j-- ;
            if (i<=j) {
                swap(a, i, j);
            }
        } while (i<j);
    }
}

```

```

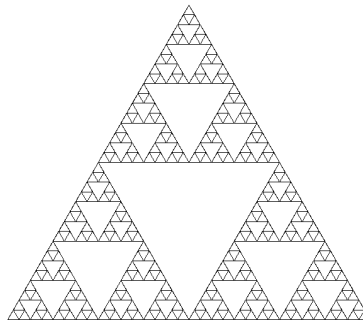
        i++; j--;
    }
    } while (i<=j);
    if (j<k) l=i;
    if (k<i) m=j;
    }
    return a[k];
}

int main()
{
    int arr[] = {1, 9, 8, 4, 11, 21, 2, 7, 30, 6, 23, 9};
    int n = sizeof(arr) / sizeof(arr[0]);
    printf("%d", fun(arr,n,4));
    return 0;
}

```

Section 3: According to the specification, complete each program (2 marks for each blank, total 30 marks)

- The following program will draw the graphics below using recursive(递归) method. Please fill in the blanks to complete the program. **Note:** the program is based on our course's graphics library.



```

#include "graphics.h"
#include <math.h>
#define LEN 6.0
#define PI 3.14159
#define EPS 0.05
typedef struct {
    double x, y;
} VERTEX;
VERTEX MidPoint(VERTEX A, VERTEX B);
void DrawTriangle(VERTEX A, VERTEX B, VERTEX C);
void FraTriangle(VERTEX A, VERTEX B, VERTEX C);
void Main()
{
    VERTEX A, B, C;
    double cx, cy;
    InitGraphics();
    cx = GetWindowWidth()/2;
    cy = GetWindowHeight()/2;
    A.x = cx;  A.y = cy + LEN/2*sin(PI/3);
    B.x = cx - LEN/2;  B.y = cy - LEN/2*sin(PI/3);
    C.x = cx + LEN/2;  C.y = B.y;
    FraTriangle(A, B, C);
}
void DrawTriangle(VERTEX A, VERTEX B, VERTEX C)/*Draw △ABC*/

```

```

{
    MovePen(A.x, A.y);
    DrawLine(B.x-A.x, B.y-A.y);
    DrawLine(C.x-B.x, C.y-B.y);
    DrawLine(A.x-C.x, A.y-C.y);
}
VERTEX MidPoint(VERTEX A, VERTEX B)
{
    VERTEX mAB;
    mAB.x = (A.x + B.x) / 2;
    mAB.y = (A.y + B.y) / 2;
    return _____(1)_____;
}
void FraTriangle(VERTEX A, VERTEX B, VERTEX C)
{
    VERTEX mAB, mBC, mCA;
    if (fabs(A.x-B.x) < EPS) return;
    _____(2)_____;
    mAB = MidPoint(A, B); mBC = MidPoint(B, C); mCA = MidPoint(C, A);
    _____(3)_____;
    _____(4)_____;
    _____(5)_____;
}

```

2. A ring-buffered queue (循环队列) maintains the indices of the queue front and rear and “wraps around” to the beginning if the last entry of the buffer is occupied when adding a new entry. Complete the following implementation of a ring-buffered queue.

```

#include <stdio.h>
#include <malloc.h>
#include <stdlib.h>
typedef struct _queue {
    int *pBase;
    int front;
    int rear;
    int maxsize;
} QUEUE, *PQUEUE;
void CreateQueue(PQUEUE Q,int maxsize);
int FullQueue(PQUEUE Q);
int EmptyQueue(PQUEUE Q);
int Enqueue(PQUEUE Q, int val);
int Dequeue(PQUEUE Q, int *val);
void CreateQueue(PQUEUE Q,int maxsize)
{
    Q->pBase = (int *)malloc(sizeof(int)*maxsize);
    if(NULL == Q->pBase) {printf("Memory allocation failure"); exit(-1);}
    Q->front = 0; Q->rear = 0; Q->maxsize = maxsize;
}
int FullQueue(PQUEUE Q) /*Check whether Q is FULL,1--TRUE, 0--FALSE */
{
    return _____(6)_____;
}
int EmptyQueue(PQUEUE Q) /*Check whether Q is EMPTY,1--TRUE, 0--FALSE */
{
    return _____(7)_____;
}
int Enqueue(PQUEUE Q, int val)
{
    if(FullQueue(Q)) return 0;
}

```

```

    _____ (8) _____ = val;
    Q->rear = _____ (9) _____;
    return 1;
}
int Dequeue(PQUEUE Q, int *val)
{
    if(EmptyQueue(Q)) return 0;
    _____ (10) _____ = Q->pBase[Q->front];
    Q->front = (Q->front+1)%(Q->maxsize);
    return 1;
}

```

3. The following program uses the graphics library demonstrated in our course to flash a circle drawn in the center of an window once every 100 milliseconds. The ESCAPE key is used as a switch to toggle the blink.

```

#include <windows.h>
#include "genlib.h"
#include "graphics.h"
#define TIMER_BLINK100 1
const int mseconds100 = 100;
static double ccx = 1.0, ccy = 1.0;
static double radius = 1.0;
static bool isBlink = FALSE;
static bool isDisplayCircle = TRUE;
void DrawCenteredCircle(double x, double y, double r);
void KeyboardEventProcess(int key,int event);
void TimerEventProcess(int timerID);
void Main()
{
    _____ (11) _____;
    registerKeyboardEvent(_____ (12) _____);
    registerTimerEvent(_____ (13) _____);
    ccx = GetWindowWidth()/2; ccy = GetWindowHeight()/2;
    DrawCenteredCircle(ccx, ccy, radius);
    if(isBlink) startTimer(TIMER_BLINK100, mseconds100);
}
void DrawCenteredCircle(double x, double y, double r) {
    MovePen(x+r, y);
    DrawArc(r, 0.0, 360.0);
}
void KeyboardEventProcess(int key,int event) {
    if(event == KEY_DOWN && key == VK_ESCAPE) {
        isBlink = !isBlink;
        if (isBlink ) {
            startTimer(TIMER_BLINK100, mseconds100);
        } else {
            cancelTimer(_____ (14) _____);
            DrawCenteredCircle(ccx, ccy, radius);
        }
    }
}
void TimerEventProcess(int timerID) {
    if(timerID == TIMER_BLINK100) {

```

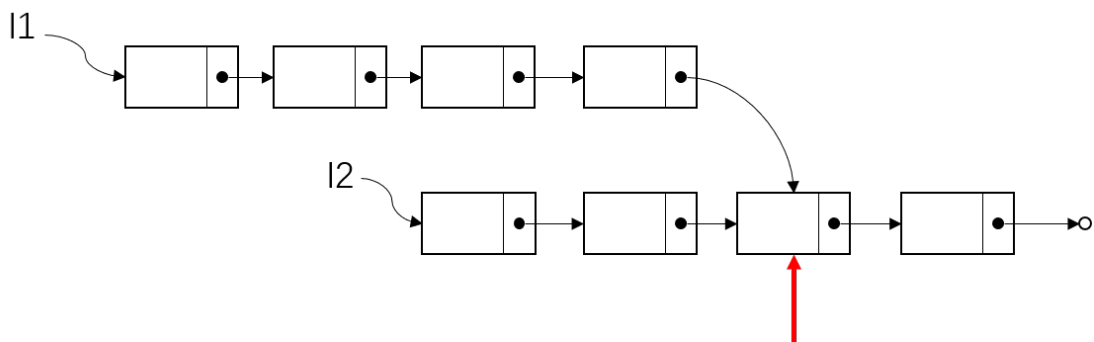
```

        bool erasemode = GetEraseMode();
        SetEraseMode(isDisplayCircle);
        DrawCenteredCircle(ccx, ccy, radius);
        SetEraseMode(erasemode);
        _____;
    }
}

```

Section 4: Algorithms design (10 marks for each item, total 20 marks)

- Write a function "FindFirstCommonNode" to find the first common node (a node that exists in both lists, e.g. the node pointed by the bold arrow in the example below) of two single linked lists, assuming no cycle in both. The function should return the pointer to the first common node of the two input lists, or NULL if no common node is found. You can call the ListLength(ListNode* l) function to obtain the length of l.



```

typedef struct _ListNode {
    void* data;
    struct _ListNode *next;
} ListNode;
int ListLength(ListNode* l)
{
    int count = 0;
    while (l){
        l=l->next;
        count++;
    }
    return count;
}
static ListNode* FindFirstCommonNode(ListNode* l1, ListNode* l2)
{
    .....
}

```

- Binary Selection Sort.
Selection sort is a simple sort algorithm which selects the max or min element in the array and swaps it with the last or first element in each round. An improvement can be introduced to find both the max and the min elements, and swap them to the position in one round. Now, given the prototype of the sort function below, finish the function body. The sorted data should be in ascending order.

```

void binSelection(int array[], int n);

void binSelection(int array[], int n)
{
    .....
}

```