浙江大学 2019－2020 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： 2020 年 09 月 08 日, 考试时间： 120 分钟

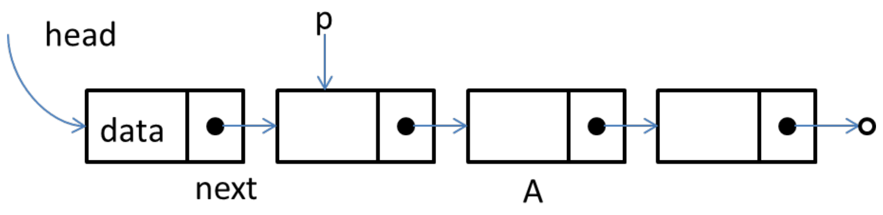
诚信考试，沉着应考，杜绝违纪.

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)

1. About function-calls, _______ is NOT correct among the following descriptions.
   A. Recursive functions, like normal functions, can use global variables.
   B．In recursive functions, static local variables are created only once.
   C．When calling functions in nest(嵌套调用函数), the function called first returns first, and the function called later returns later.
   D．The number of recursive-calls is not unlimited.
2. The algorithm time complexity of the following program fragment is _______.
```
for (i=1; i<=n; i++) x++;
for (i=1; i<=n; i++)
    for (j=1; j<=n; j++) x++;
```
   A. O(N)　　B. O(N^2)　　C. O(2N)　　D. O(N^3)
3. Given the following definition and declaration, which of the expressions does not refer to the data field of record t ? _______
```
struct node {
        unsigned id;
        int data;
} t, *p;
p=&t;
```
   A. p.data　　B. n.data　　C. p->data　　D. (*p).data
4. What is the correct statement to delete a node A from the single linked list illustrated below?



A. p->next=NULL; B. p.next=p.next.next;

C. p->next=p->next->next; D. p.next=NULL;

5. Which of the following is incorrect about the abstract data type queue?

   A. It can be operated on both head and tail.

   B. It is first-in, first-out.

   C. Its typical operations are CreateQueue, EnQueue, DeQueue, QueueLength and DeleteQueue.

   D. DeQueue operation always involves moving all remaining elements in the queue.

6. Suppose the following functions ***Larger()*** and ***Smaller()*** (which can be used to compare the size of two numbers) have been defined as shown below, and an integer array ***a***:

   int Large(int a, int b);
   int Less(int a, int b);
   static int a[10]={6,3,5,7,4,2,9,8,0,1};

   Now you want to define the function ***Bubble()***, by calling it, the integer array ***a*** can be sorted, then ________ is the correct prototype declaration of the function ***Bubble()***.

   A. void Bubble(int a[], int size, int (*cmd)(int,int));

   B. void Bubble(int a[], int size, int *cmd(int,int));

   C. void Bubble(int *a, int size, int cmd(int,int));

   D. void Bubble(int *a, int size, int *cmd(int,int));

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

   A.
   ```
   int g (int c, int (*f) (int)) {
       return f(c);
   }
   ```
   B.
   ```
   int g (int c, int (*f) (int)) {
       return (&*f)(c);
   }
   ```
   C.
   ```
   int g (int c, int f (int)) {
       return (*&f) (c) ;
   }
   ```
   D.
   ```
   int g (int c, int (*f) (int)) {
       return (&f)(c);
   }
   ```

8. For function definition, which one below is NOT correct? ______.

   A. The return type of a function shall be void or an object type other than array type.

   B. Each parameter of a function has automatic storage duration.

   C. After all parameters have been assigned, the compound statement that constitutes the body of the function definition is executed.

   D. For the special case of a parameter list consisting of a single parameter of type void, an identifier (name) shall be included.

9. For graphics library, which one below is NOT correct? ______.

A. Before using any of the other functions in the graphics library, you must first call ***InitGraphics*** to initialize the package.
B. To draw a line segment, you first call ***MovePen*** to position the pen at the starting point and then call ***DrawLine*** to draw the actual line.
C. You can extend the capabilities of the library be defining new functions, such as ***DrawBox*** and ***DrawCenteredCircle***.
D. The functions in a library are written by implementors and are called by clients. The point at which clients and implementors come together is called the package.

10. Given the definition:

```
typedef int F (int a);
F g, h;
int (*p[])(int) ={g, h};
int q(F* array[]);
```

In the following statements, _____ is wrong.

A. q(p);   B. q(p+1);   C. q(&p[0])   D. q(&g);

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

1. Write down your answers.
   (1) When input ***Hello,World<ENTER>***, what's the result after executing the following code fragment?

   ____

```
char *str;
scanf("%s", str);
printf("%s", str);
```

   (2) Write down the declaration of a function pointer to a void procedure(过程) ***f*** with two formal parameters: the first is an integer variable, the other is a pointer to an array of 10 integers.

   ____

2. What is the output of the following program? ______________________

```
#include <stdio.h>
struct st {
    char c; char s[80];
};
char * f(struct st *t);
struct  st a[4] = {{'3',"123"}, {'3',"321"}, {'2',"123"}, {'4',"321"}};

int main( )
{
    int k;
    for(k = 0; k < 4; k++)  printf("%s", f(a+k));
    return 0;
}

char * f(struct st *t)
{
    int k = 0;
    while(t->s[k]!='\0'){
      if(t->s[k] == t->c) return t->s+k;
      k++;
    }
   return t->s;
}
```

3. Given the following program, what will be the returned linked list of the function process(h1,h2) when the two input lists are:
h1: 1->3->5->9->10->23->34->59->70
h2: 0->3->6->7->9->14->34->44

____

```
#include <stdio.h>
typedef struct _NodeT{
int data;
struct _NodeT *next; } ListNode ;
typedef ListNode *LinkList;

ListNode *create_node(int data){
ListNode* p = (ListNode*) malloc(sizeof(ListNode));
p->data = data;
p->next = NULL;
return p;
}

ListNode* extend(ListNode* a, int data){
    return a->next = create_node (data);
}

ListNode* process(ListNode* l1, ListNode* l2){
    ListNode *p1=l1, *p2=l2, *m=create_node(0), *mp = m;
while (p1&&p2){
    if (p1->data > p2->data){
            mp = extend(mp, p2->data);
            p2 = p2->next;
    }else if (p1->data < p2->data){
            mp = extend(mp, p1->data);
            p1 = p1->next;
    }else{
            mp = extend(mp, p2->data);
            p1 = p1->next;
            p2 = p2->next;
    }
}
while(p1){
            mp = extend(mp, p1->data);
            p1 = p1->next;
    }
    while(p2){
            mp = extend(mp, p2->data);
            p2 = p2->next;
    }
    return m->next;
}
```

4. The fllowing program will output ________.

```
#include <stdio.h>
typedef void* (*H)(int a);
void* h(int a)
{
    if(a) printf("%d", a);
    return h;
}

int main(void)
{
    ( (H) h(0) )(100);
    return 0;
}
```

5. The following program is based on the graphics library demonstrated in our course. What’s the functions of the program? ____

```
#include <windows.h>
#include "genlib.h"
#include "graphics.h"
#define TIMERB 1
const int mseconds = 500;
static double ccx = 1.0, ccy = 1.0;
static double radius = 1.0;
static bool isB = FALSE;
static bool isD = TRUE;
void DrawCenteredCircle(double x, double y, double r);
void KeyboardEventProcess(int key,int event);
void TimerEventProcess(int timerID);
void Main()
{
    InitGraphics();
    registerKeyboardEvent(KeyboardEventProcess);
    registerTimerEvent(TimerEventProcess);
    ccx = GetWindowWidth()/2; ccy = GetWindowHeight()/2;
    DrawCenteredCircle(ccx, ccy, radius);
    if(isB) startTimer(TIMERB, mseconds);
}
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) {
            isB = !isB;
            if (isB ) {
                    startTimer(TIMERB, mseconds);
            } else {
```

```
                        cancelTimer(TIMERB);
                        DrawCenteredCircle(ccx, ccy, radius);
                }
        }
}
void TimerEventProcess(int timerID) {
        if(timerID == TIMERB) {
                bool erasemode = GetEraseMode();
                SetEraseMode(isD);
                DrawCenteredCircle(ccx, ccy, radius);
                SetEraseMode(erasemode);
                isD=!isD;
        }
}
```

6. The following program will output____.

```
/*Note: This program uses the text-library of our course.*/
#include <stdio.h>
#include "genlib.h"
#include "strlib.h"
#include "simpio.h"
static void ListMnemonics(string str);
static string dialpad[] = {
    "0", "1", "ABC", "DEF", "GHI", "JKL", "MNO", "PRS", "TUV", "WXY"
};
static char Mnemonics[100] = {'\0'};
static int index = 0;

int main()
{
   ListMnemonics("908");
}

static void ListMnemonics(string str)
{
    string ptr;

   if (*str == '\0') {
      Mnemonics[index] = 0;
      printf("%s#", Mnemonics);
      return;
   }
    ptr = dialpad[*str-'0'];
    while (*ptr != '\0') {
            Mnemonics[index++] = *ptr;
            ListMnemonics(str+1);
            index--;
            ptr++;
    }
    return;
}
```

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

1. Assuming that we have a ***network message library*** (***nml***) which contains type definitions and functions. If we already know they will be used as follows.

```
…
int MsgHandle(char * msg, int size)
{
   …
}

int main(int argc, char** argv)
{
   SwitchingCenter center = { … };
   InitNetMsg(&center, *argv);
   RegisterNetMsgEven(MsgHandle)
   return MsgLoop(center);
}
```

Please complete the following ***nml*** header file with the most likely code.

```
typedef void (*NetMsgEventCall) (___(1)___);

/* Initialize network message mechanism. */
void InitNetMsg(___(2)___);

/* Register the callback function of network message. */
void RegisterNetMsgEvent(NetMsgEventCall callback);

/* Message loop. */
__(3)__ MsgLoop(__(4)__)
```

2. The following program is an implementation of a Queue using single linked list. It implements basic operations including CreateQueue, EnQueue, DeQueue and QueueLength. Please fill in the blanks to complete the implementation.

```
typedef struct _NodeT{  int data;  struct _NodeT *next; } ListNode;
typedef ListNode *LinkList;
// The queue, front stores the front node of LL and rear stores the last node of LL
typedef struct{ ListNode *front, *rear;} Queue;

// A utility function to create a new linked list node.
ListNode* newNode(int k) {
   ListNode *temp = (ListNode*)malloc(sizeof(ListNode));
   temp->data = k;
   temp->next = NULL;
   return temp;
}

// A utility function to create an empty queue
Queue CreateQueue() {
   Queue *q =__________(5)________;
   q->front = q->rear = NULL;
   return q;
}

int QueueLength(Queue* q){
   ListNode *p;
   int length=0;
   for(_________(6)________);
   return q->front==NULL? 0: length+1;
}

// The function to add a key k to q
void EnQueue(Queue* q, int k){
   ListNode* temp = newNode(k);
   // If queue is empty, then new node is front and rear both
   if (q->rear == ___(7)___) {
       q->front = q->rear = temp;
       return;
   }
   q->rear->next = temp;
   _________(8)________;
}

// The function to remove a key from a given queue q
void DeQueue(Queue* q){
   if (q->front == NULL) return;
   // Store previous front and move front one node ahead
   ListNode *temp = q->front;
   q->front =___________(9)________;
   if (q->front == NULL) q->rear = NULL;
   free(temp);
}
```

3. Bracket matching: Enter a string of characters (no more than 50 characters), which may include brackets, numbers, letters, punctuation marks, and spaces. The following program checks whether the (), [], {} in this string of characters match; if match, output "yes", and if not match, output "no".
For example: input ***sin(20+10)***, then output ***yes***; input ***{[}]***, output ***no***.
Given the abstract stack functions library(抽象堆栈函数库) as below:
typedef struct stackCDT *stackADT; /*Abstract Data Type of stack*/
stackADT NewStack(void); /* allocates and returns an empty stack.*/
void FreeStack(stackADT stack); /* frees the storage associated with stack.*/
void PushStack(stackADT stack, void *obj); /*adds obj to the top of the stack.*/
void *PopStack(stackADT stack); /* removes the data at the top of the stack and returns it to the client.*/
bool IsemptyStack(stackADT stack); /* Whether the stack is empty.*/
void *TopStack(stackADT stack); /* returns the top element of the stack */
Please fill in the blanks to complete the program.

```
#include <stdio.h>
#include "genlib.h"
#include "simpio.h"
#include "strlib.h"
#include "stack.h"

int main()
{
   ____(10)____ stack;
   string line;
   char *item, ch;
   int i, len;

   printf("Input a line with some (, ), [, ], {, } characters\n");
   stack = ____(11)____();
   while (TRUE) {
      printf(":");
      line = GetLine();
      if (StringEqual(line, "quit")) break;
      len = StringLength(line);
      if (len == 0) continue;
      for (i = 0; i < len; i++) {
            if (line[i] == '(' || line[i] == '[' || line[i] == '{') {
                     item = New(____(12)____);
                     *item = line[i];
                     ______(13)______(stack, item);
            } else if (line[i] == ')' || line[i] == ']' || line[i] == '}') {
                     ch = *(char *)_____(14)_____(stack);
                     if (line[i] == ')' && ch == '(' || line[i] == ']' && ch == '[' ||
                        line[i] == '}' && ch == '{') FreeBlock(____(15)____(stack));
            }
      }

      if (IsemptyStack(stack)_(stack)) {
            printf("Yes\n");
      } else {
            printf("No!\n");
      }
   }
   FreeStack(stack);
   return 0;
}
```

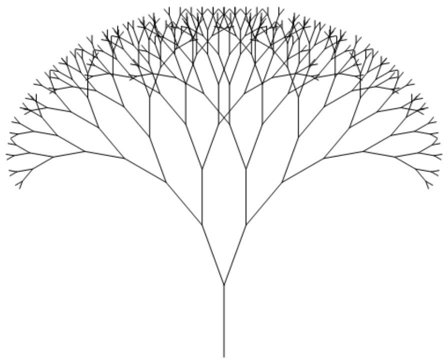
Section 4: Algorithms design (10 marks for each item, total 20 marks)

1. Fractal Tree

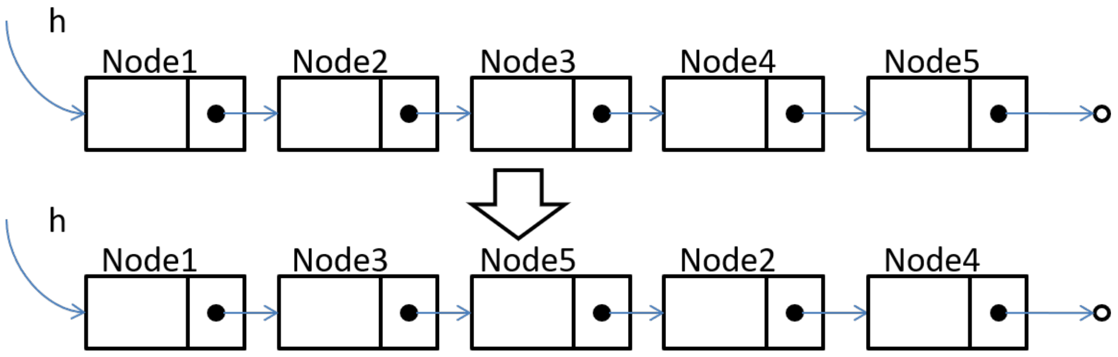
   Please write a c program to draw a fractal tree by using the graphics library demonstrated in our course.

Rules:

1) the shape of the fractal tree should be the same as the following picture.
2) Use scanf function to input initial length of the branch and the order of the fractal.
3) Each branch should decrease by 25% in width for each time it splits.
4) The angle between two sibling branches is 30 degree.



2. Write a function that processes an input single linked list h and moves all nodes at even positions（偶数位置）after the nodes of odd positions (奇数位置), as illustrated below, the original order within the same group has to remain unchanged. The function should not allocate more than one additional list node, i.e. its space complexity should be ***O(1)***. You may use the ***CreateNode*** function below.



```
typedef struct _NodeT{  int data;  struct _NodeT *next; } ListNode ;
typedef ListNode *LinkList;

ListNode *CreateNode(int data){
    ListNode* p = (ListNode*) malloc(sizeof(ListNode));
    p->data = data;
    p->next = NULL;
    return p;
}
```