2017-2018 学年第二学期期末考试 A 卷

一、单选题(每小题 3 分，共 30 分)

1、关于算法的时间空间复杂度，以下说法正确的是_____

A、时间复杂度和空间复杂度都是问题规模 n 的函数

B、计算量等于$8n^2+n$的算法是n^2-3n+1的算法复杂度的 8 倍

C、空间复杂度为$O(n^3)$，表明该算法的问题规模与n^3成正比

D、算法的时间复杂度越大空间复杂度也越大

2、下面程序的输出结果是_____

```
int main(){
    struct st{
        int n;
        struct st *next ;
    } a[3] = {5, &a[1], 7, &a[2], 9, &a[0]}, *p=a;
    int i;
    for( i=0; i<5; i++ ) p = p -> next;
    printf("%d\n",p -> n);
    return 0;
}
```

A、0　　B、5

C、7　　D、9

3、以下叙述正确的是_____.

A、任一个 C 语言源文件（.c 文件）必须包含 main 函数

B、一个 C 语言源文件（.c 文件）可以包含两个以上 main 函数

C、在一个可以正确执行的 C 语言程序中，一个函数的原型声明可以出现任意多次

D、C 语言头文件（.h 文件）和源文件(.c 文件）都要进行编译

4、如下关于图形交互操作的说法中不正确的是_____.

A、MovePen(x, y)把画笔移到（x,y）的位置，接下来绘图以这个位置开始

B、DrawLine(dx, dy)是绘制一条以（dx, dy）为终点的线段，绘制完成后画笔的位置是（dx,dy）

C、DrawArc(r,start,360)绘制半径为 r 的圆，绘制完成后不改变绘图的起点位置

D、DrawTextString("Hello")从当前位置开始输出文本“Hello”，输出完后当前位置改变。

5、程序中应避免“魔鬼数值”，魔鬼数值指程序中直接出现的文字量，如 100、3.1415，应该用符号常量来替代魔鬼数值。使用符号常量，下面说法中错误的是_____.

A、修改会变得更可靠　　B、修改会变得更容易

C、程序可读性提高　　D、程序中所有出现的数值都用符号常量替代包括 0 和 1

6、假设已定义如下所示的函数 Larger()和 Smaller()（可用于比较两数大小），以及整型数组 a：

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

现欲定义函数 Bubble()，通过调用它，可对整型数组 a 进行排序，则_____是函数 Bubble()的正确原型声明.

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、若有宏定义：

```
#define N    2
#define Y(n)((N+1)*n )
```

则执行语句 int z; z = 2*(N+Y(5));后，z 的值是_____.

A、语句有错　　B、不确定

C、34　　D、70

8、将 newnode 指向的新节点（newnode->next 为 NULL）插入到 head 单向链表的链头，正确的是_____

A、newnode->next = head;

C、
```
if(head==NULL)
    head = newnode;
else{
    head = newnode
    newnode->next = head->next;
};
```

B、head = newnode;

D、
```
if(head==NULL)
    head = newnode;
else{
    newnode->next = head;
    head = newnode;
};
```

9、根据下面的函数定义，调用 f(4)得到的结果是_____。

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

A、0　　B、4

C、10　　D、以上均不是

10、C 语言的编译系统对宏命令的处理是_____.

A、在程序运行时进行的

B、在程序连接时进行的

C、和 C 程序中的其它语句同时进行编译的

D、在对源程序中其它成分正式编译之前进行的

二、改错题(共 20 分)

1、在一个 C 源程序文件中，若要定义一个只允许本源文件中所有函数使用的全局变量，则该变量定义时要加关键字 extern 来限定。（2 分）

2、在最坏情况下，选择、插入、冒泡、归并和快速等排序算法的时间复杂度都一样，是$O(n^2)$。（2 分）

3、下面的递归函数欲求 m 和 n 的最大公约数，请指出其中的错误并修正。（2 分）

```
1  int gcd(int m, int n)
2  {
3     if (n)
4        return gcd(n,m/n);
5     else
6        return m;
7  }
```

4、以下程序对输入的字符串字符进行简单处理后输出。请指出其中的错误并修正。（4 分）

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #define LINELENGH 81
4
5  int main( )
6  {
7      char *p,*q;
8      p =malloc(LINELENGH);
9      gets(p);
10     while (*p) {
11         *p=*p+1;
12         putchar(*p);
13         p++;
14     }
15     free(p);
16     return 0;
17 }
```

5、假设工作目录下有文件 a1.h 和 a1.c。请找出代码中的错误并修正。要求保留函数中的各条语句。（5 分）

```
0  /*文件 a1.h*/
1  #include <stdio.h>
2  #define a1.h
3  #ifndef _A1_H_
4
5  void Print();
6  int SayHello(char, double);
7
8  #ENDIF
```

```
0  /*文件 a1.c*/
1  #include <stdio.h>
2  #include "a1.h "
3
4  void Print(){
5     puts("Morning World! ");
6  }
7  int SayHello(char ch, double x){
8     printf("%c,%lf ", ch, x);
9  }
```

6、指出下面程序的错误，不需要改正。程序输入一个字符串和一个字符，如果该字符在字符串中，就从该字符首次出现的位置开始输出字符串中的字符。例如，输入字符串 This is the input string 和字符 t，那么将输出 the input string。（2 分）

```
1   #include <stdio.h>
2   char *getone()
3   { char ch, str[80], *s = str;
4      gets(str); /*  输入一个字符串*/
5      ch = getchar(); /*  读入一个字符*/
6      while ( *s )
7           if ( *s == ch ) return s; /*  返回字符在字符串中的地址*/
8           else s++;
9        return (NULL);
10  }
11
12  int main( )
13  { char *p = NULL;
14     if( p = getone( ) ) /*  调用函数 getone( ) */
15          printf("%s\n", p);
16     else
17           printf("Not Found\n");
18  return 0;
    }
```

7、以下程序对输入的 n 个学生的成绩信息，计算并输出每个学生的个人平均成绩。找出其中错误并修正。（3 分）

```
1   #include <stdio.h>
2   double count_average(struct student s)
3   {
4       return (s.math + s.english) / 2.0;
5   }
6   int main( )
7   {
8           struct student{
9           char name[10];
10          int english, math;
11          double average;
12      };
13      int i, n;
14      struct student s1;
15      scanf("%d", &n);
16      for(i = 1; i <= n; i++){
17              scanf("%s%d%d%d",&s1.name,&s1.math,&s1.english);
18              s1.average = count_average(s1);
19              printf("name:%s, average:%.2f\n", s1.name, s1.average);
20      }
21      return 0;
22  }
23
```

三、问答题（共 10 分）

1、结构体变量作为函数参数的优缺点是什么？结构体指针作为函数参数与结构体变量相比优势是什么？

2、假定下面代码中相关变量都已正确定义，且规定循环变量初始化、循环终止条件判断、循环变量自增、赋值均为基本操作，每种基本操作执行时间为常数 1。

```
for(column = 0; column < 10000; column++) {
    for(row =0; row < 5; row++) {
        sum += table[row][column];
    }
}
```

接下来请完成下述任务：

1) 分析上面代码的执行时间；

2) 改写代码，使之优化；

3) 分析优化后代码的执行时间，与优化前代码比，是否能得出什么结论。

四、程序填空题（每空 2 分，共 26 分）

1、如下是链表结点的结构类型定义：

```
typedef struct node {
    int data;
    struct node *next;
} NodeT;
```

请完成编写下列函数，它将不带头结点的单链表中的逆序结点删除，形成从小到大的有序序列。例如：若当前链表 h 为 3->3->2->6->1->4->7，将删除逆序结点 2、1、4，形成链表 3->3->6->7。

```
NodeT *order( NodeT *h )
{
    NodeT *p, *t;
    p = h;
    while (___1___) {
        t = p->next;
        if ( !t ) break;
        else
            if (___2___)
                p = p->next;
            else {
                p->next = t->next;
                ___3___;
            }
        }
        return h;
    }
```

2、以下程序用“比较计数”法对结构数组 a 按字段 num 进行升序排列。“比较计数”法的基本思想是：通过另一字段 con 记录 a 中小于字段 num 的元素的个数。待算法结束，a[i].con 就是 a[i].num 在 a 中的排序位置。请根据题意完成程序填空。

```
#include <stdio.h>
#define N 10
struct c{
    int num;
    int con;
} a[N];
int main()
{
    int i,j;
    for( i=0; i<N; i++) {
        scanf("%d",&a[i].num);
                ___4___;
    }
    for(i=N-1; i>=1; i--)
```

```
        for(___5___; j>=0; j--)
            if(___6___) a[j].con++;
            else a[i].con++;
        for(i=0;i<N;i++)
            printf("%d,%d\n",a[i].num,a[i].con);
        return 0;
    }
```

3、以下程序实现随笔画功能，即当鼠标左键按下拖动时，在窗口中随着鼠标位置画出轨迹，当鼠标左键抬起时则不画。请根据题意完成程序填空。

```
#include "graphics.h"
#include <winuser.h>
/*其他文件包含因篇幅所限未一一列出*/
void Painter(int x, int y, int button,___7___)
{
    double cx = ScaleXInches(x);
    double cy = ScaleYInches(y);
    double dx,dy;
    static int isLeftButtonDown = 0;
    static double lx = 0.0, ly = 0.0;

    switch(event){
        case BUTTON_DOWN:
            if(button==LEFT_BUTTON) isLeftButtonDown = 1;
            lx = cx; ly = cy;
            break;
        case BUTTON_UP:
            if(button==LEFT_BUTTON) isLeftButtonDown = 0;
            break;
        case___8___:
            if(isLeftButtonDown) {
                MovePen(lx,ly);
                dx = cx-lx;
                dy = cy-ly;
                DrawLine(dx,dy);
                lx = cx;
                ly = cy;
            }
            break;
        }
    }
    void Main(){
        ___9___;
        SetPenSize(1);
        SetPenColor("Black");
```

```
        ___10___
    }
```

4、快速排序（简称快排）是一种分而治之的排序方法。首先任意选取一个元素，作为 pivot，然后设法将数组剖分为两部分：前半部元素<= pivot，后半部元素> pivot； 再对两个子数组进行递归排序，直到子数组元素个数不超过 1 个。阅读以下的快速排序程序，并将其填写完整。

```
/*a - 被排序的数组*/
/* low - 被排序的首元素下标*/
/* high - 被排序的末尾元素下标*/
void Qsort(int a[], int low, int high)
{
    int s = low, t = high, pivot;
    if (low >= high) return;
    pivot = a[ (s+t)/2 ];
    a[ (s+t)/2 ] = a[s];
    while ( s < t ) {
        while (s < t && a[ t ] >= pivot )___11_____ ;
        a[ s ] = a[ t ];
        while ( s < t && a[ s ] <= pivot)___12_____;
        a[ t ] = a[ s ];
    }
    a[ s ] = pivot;
    Qsort(a, low, s - 1);
    ___13___;
}
```

五、算法设计（共 14 分）

1、考虑用递归的方法对整型数组 a 前 n 个元素进行求和，请完成以下任务：

1）写出递归算法的递归式和递归出口；

2）写出实现该算法的函数定义（函数原型：int SumArr(int a[], int n)；）

2、定义 n 个元素的集合 S = {ai, 0≤i≤n-1}的“中位数”m 如下：如果把这 n 个数存入一个数 A[n]并从小到大排序，那么元素 A [(n-1)/2] 就是中位数 m。现给定两个整数数组 A 和 B，它们都是 n（n > 0）个元素并且已经从小到大排序。现要找出 A 和 B 合并后的集合 S 的中位 m。

请完成：

1)写出一个时间复杂性尽可能好的算法思想，并分析你的算法的时间复杂性，

2)写出实现该算法的函数定义(函数原型:int Median(int A[ ],int B[ ],int n);),要求加入足够注释。

2017-2018 学年第二学期期末考试 A 卷参考答案

一、单选题(每小题 3 分，共 30 分)

1、**【正解】**A

【解析】B 计算量不是严格意义上的 8 倍，可举反例，如 n=3；

C 问题规模为 n；

D 时间复杂度和空间复杂度没有绝对的关系。

【考点延伸】《考试宝典》时间复杂度和空间复杂度分析

2、**【正解】**D

【解析】一开始 P 指向 a[0]经过 5 次循环后，指向 a[3],a[3].n=9

【考点延伸】《考试宝典》专题九 9.7 用指针处理链表

3、**【正解】**C

【解析】A，B 一个程序必须有且仅有一个 main 函数。但可以有多个.c 文件。

D .c 文件包含.h 文件

【考点延伸】《考试宝典》专题十 10.1 文件的基础知识

4、**【正解】**B

【解析】Drawline(dx,dy)是从当前位置(x,y)画线段，终点为(x+dx,y+dy)。

【考点延伸】《考试宝典》Drawline 函数的使用

5、**【正解】**D

【解析】0 和 1 没有必要用字符常量，如此反而会导致代码更难以理解。

【考点延伸】《考试宝典》专题二 2.1 常量和变量

6、**【正解】**A

【解析】最后一个形参为函数指针，函数指针的写法为：

函数返回值类型 (* 指针变量名) (函数参数列表)

故选 A。

【考点延伸】《考试宝典》专题八 8.2 函数的调用

7、**【正解】**C

【解析】Y(5)=((N+1)*5)=15，z=2*(N+15)=34；

【考点延伸】《考试宝典》专题八 8.4 预处理命令

8、**【正解】**D

【解析】按照链表添加的程序，判断首指针是否为空，若为空则建立新首指针，若不为空，则将当前的头指针给该指针的 next，再将该指针设为头指针。

【考点延伸】《考试宝典》专题九 9.7 用指针处理链表

9、**【正解】**C

【解析】f(4)=f(3)+4；f(3)=f(2)+3；f(2)=f(1)+2；f(1)=f(0)+1；f(0)=0；所以 f(4)=4+3+2+1=10；

【考点延伸】《考试宝典》专题八 8.2 函数的调用

10、**【正解】**D

【解析】编译系统对宏的预处理是纯文本替换，在正式编译前就已经完成。

【考点延伸】《考试宝典》专题八 8.4 预处理命令

二、改错题（共 20 分）

1、**【正解】**在一个 C 源程序文件中，若要定义一个只允许本源文件中所有函数使用的全局变量，则该变量定义时要加关键字 static 来限定。

【考点延伸】《考试宝典》专题七 7.2 变量的存储类型

2、**【正解】**归并排序是严格$n\log n$的，在最坏情况下选择、插入、冒泡、快排复杂度一样，都是$O(n^2)$

【考点延伸】《考试宝典》算法的时间复杂度分析

3、【正解】int gcd(int m, int n){

```
    if(n)
        return gcd(n,m%n);
    else
        return m;
}
```

【解析】应取余数，而不是商。

【考点延伸】《考试宝典》专题八 8.2 函数的调用

4、【正解】p = malloc(LINELENGH); 改为 p=(char*)malloc(sizeof(char)* LINELENGH);

【解析】动态内存分配 malloc 函数的规范用法。

【考点延伸】《考试宝典》专题五 5.4 字符数组

5、【正解】左面改成

```
#ifndef _A1_H_
#define _A1_H_
#include <stdio.h>
void Print();
void SayHello(char,double);
#endif
```

右面改成

```
#include <stdio.h>
#include “a1.h”
void Print()
{ puts("Morning World! "); }
void SayHello(char ch, double x)
{ printf("%c,%lf ", ch, x); }
```

【解析】左侧：调整语句顺序，先#ifndef 语句，再#define 语句；右侧：函数无返回值，类型为 void。

【考点延伸】《考试宝典》专题八 8.4 预处理命令

6、【解析】str[80]应定义为全局变量，否则 getone 函数调用完毕，str 数组便不存在了。或者将 str[80]数组在主函数中定义和读取。

【考点延伸】《考试宝典》专题七 7.2 变量的存储类型

7、【正解】scanf("%s%d%d%d",&s1.name,&s1.math,&s1.english);

改为 scanf("%s%d%d",s1.name,&s1.math,&s1.english);

【解析】读取字符串时 scanf 函数的正确写法。

【考点延伸】《考试宝典》专题一 1.3 格式化输出/输入函数

三、问答题（共 10 分）

1、【解析】结构体变量作为函数参数的优点：容易理解，简单方便。

缺点：采用值传递的方式，如果在函数被执行期间改变了形参的值，该值不能反映到主调函数中的对应的实参。

而指针传递通过改变指针指向的结构体变量的值，可以间接改变实参的值。并且，在调用函数期间，仅仅建立了一个指针变量，大大的减小了系统的开销。

【考点延伸】《考试宝典》专题九 9.3 结构在函数间的传递

2、【正解】

（1）代码执行时间约为 190000。

（2）改为：

```
for(row =0; row < 5; row++)
  { for(column = 0; column < 10000; column++)
      { sum += table[row][column]; }
  }
```

（3）优化后代码执行时间约为 150000，在交换循环变量情况不变的情况下，循环次数少的循环变量可以放到外面。

【解析】

（1）循环变量初始化 10000 次，条件判断 70000 次，循环变量自增 60000 次，赋值 50000 次。

（2）通过交换循环变量层级，达到降低除赋值以外的运行时间。

（3）循环次数少的循环变量放到外面，减少程序运行次数。交换后，循环变量初始化 6 次，条件判断 50000 次，循环变量自增 50000 次，赋值 50000 次。

【考点延伸】《考试宝典》时间复杂度分析

四、程序填空题（每空 2 分，共 26 分）

1、**【正解】**（1）p!=NULL　（2）p->data <= t->data　（3）free(t)

【解析】先用 p!=NULL 限制不越界。

p->data<=t->data 保证不降序。

free(t)将 t 的空间释放。

【考点延伸】《考试宝典》专题九 9.7 用指针处理链表

2、**【正解】**（4）a[i].con=0;

（5）j=i-1;

（6）a[i].num<a[j].num;

【解析】先对 a[i].con 初始化，再对任意两个数据进行比较。

【考点延伸】《考试宝典》专题九 9.2 结构数组与结构指针

3、**【正解】**（7）int event

（8）MOUSEMOVE

（9）InitGraphics()

（10）registerMouseEvent(MouseEventProcess)

【解析】（7）事件

（8）鼠标移动的情况

（9）初始化图形窗口

（10）注册鼠标消息回调函数

【考点延伸】《考试宝典》图形程序设计基础

4、**【正解】**（11）t--　（12）s++　（13）Qsort(a, s+1, high)

【解析】快速排序法：

（11）右端下标递减，直到找到比基数小的数字；

（12）左端下标递增，直到找到比基数大的数字；

（13）递归调用 Qsort 函数，对右半段进行快速排序。

【考点延伸】《考试宝典》专题四 4.1 while 语句

五、算法设计（共 14 分）

1、【解析】1）递归式：a[n]+jia(a,n-1); 出口：if(!n)return a[0];

2）
```
int SumArr(int a[], int n){
    if(!n)return a[0];
    return a[n]+jia(a,n-1);
}
```

【考点延伸】《考试宝典》专题八 8.2 函数的调用

2、【解析】（1）将两数组分别分为左右两部分，左侧的两部分即为包含中位数的左侧，右侧即为可能包含中位数的右侧。由于一个数组的划分位置确定，另一个数组划分位置即可确定，所以只需使用二分查找在一个数组中查找到正确的划分位置就行了。这个算法的时间复杂度为 $O(\log_2 n)$。

（2）
```
int Median(int A[],int B[],int n){
    int left,right,mid1,mid2;
    left=0;right=n-1;
```

```
    while(left<=right){
        mid1=(left+right)/2;
        mid2=n-mid1-3;
        if(A[mid1+1]<B[mid2])
            left = mid1+1;
        else if(A[mid1]>B[mid2+1])
            right = mid1-1;
        else return A[mid1]>B[mid2]?A[mid1]:B[mid2];
    }
}
```

【考点延伸】《考试宝典》专题六 6.3 指针与数组