浙江大学 2015–2016 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： 2016 年 06 月 28 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. Given that the pushing sequence of a stack is ***{1, 2, …, n}*** and the popping sequence is ***{ p1, p2, …, pn }***. If ***p2=n***, how many different possible popping sequences can we obtain? ___.
   A. 1　　B. 2　　C. n-1　　D. n
2. Let ***P*** stands for push and ***O*** for pop. When using a stack to calculate the value of the postfix expression ***1 2 3 + * 4 –***, the stack operation sequence is ___.
   A. PPPOOPOO　　B. PPOOPPOOPPOO
   C. PPPOOPOOPPOO　　D. PPPOOPOOPPOOPO
3. For the following declaration, which is the correct reference to ***a***?___.
   ```
   struct {
       int a;
       float b;
   } data, *p=&data;
   ```
   A. (*p).data.a　　B. (*p).a　　C. p->data.a　　D. p.data.a
4. If a function is declared as:
   ```
   int (*func(int) ) (double);
   ```
   The return type of this function is:___.
   A. An int;
   B. A pointer to an int;
   C. A pointer to a function that returns an int;
   D. A pointer to a double.
5. Given code fragment below:
   ```
   #define SQ(x)   x*x
   #define DD(x,y)  SQ(x)-SQ(y)
   printf("%d", DD(2*3, 2+3);
   ```
   The output will be ___.
   A. 43　　B. 11　　C. 25　　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},{9,10,11,12}}, *p=a;
   int z;
   ```

```
z=*((int *)(p+1)+2);
```

A. 3 B. 7 C. 10 D. None of the above

7. Which one of the following algorithms is ***NOT*** an ***O(n)*** algorithm?___.
   A. Finding someone in your telephone book;
   B. Linear Search;
   C. Deletion of a specific element in a double-linked List (unsorted);
   D. Comparing two strings.
8. Which one of the following algorithms is ***NOT*** an ***O(1)*** time complexity algorithm?___.
   A. Calculating the average value of the first three elements of a double-linked list;
   B. Searching in a stack;
   C. Accessing to the third element of a single-linked list;
   D. Accessing to the third element of an array.
9. Binary search uses at worst _①_, at average _②_ , and at best _③_comparisons.
   A. ①=O(log n),②=O(log n) and ③=O(1);
   B. ①=O(n), ②=O(log n) and ③=O(log log n);
   C. ①=O(n), ②=O(log n) and ③=O(1);
   D. ①=O(log n), ②=O(log n) and ③=O(log n).
10. When ***dsp("12")*** is called, the function prints out _____.(The ASCII value of '0' is 48.)

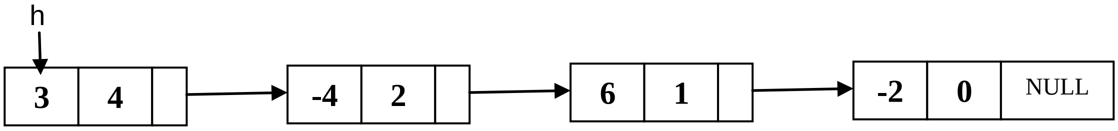
```
void dsp(char *s)
{
    if(*s)   dsp(s+1);
    printf("%d",*s);
}
```

A. 21 B. 12 C. 5049 D. 05049

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

1. Given the definition of a linked list and a practical example (linked list ***h***), Please give the value of variable ***res*** after calling the function ***res=f(h,2)***.

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



```
int f(ListNode *h, int n)
{
    ListNode *p=h;
    int res=0, last, cur, i;

    if (h==NULL) return res;
    last=h->exp;
    while (p!=NULL) {
        cur=p->exp;
        for (i=last; i>cur; i--)  res=res*n;
        res += p->coe;
        last=cur;
        p=p->next;
    }
    for (i=last; i>0; i--)  res=res*n;
    return res;
}
```

2. For the structure declaration below, please give the values of each following expression (Note: These expressions are INDEPENDENT.):

(1) *(++p->s)   (2) ++p->x   (3) (p+1)->x

```
struct {
    int x;
    char *s;
} A[2]={{1, “ab”}, {3, “cd”}}, *p=A;
```

3. Given two source code files below:

a.c:

```
#include <stdio.h>
void a() { printf(“a”);}
void b() { printf(“b”);}
void c() { printf(“c”);}
```

b.c:

```
#include <stdio.h>
void a();
void b();
void c();
int main()
{
    void (*CMDS[])() = {a, b, c};
    int k;
    scanf("%d", &k);
    if ( k >=0 && k < sizeof(CMDS)/sizeof(CMDS[0])) CMDS[k]();
}
```

Put them togther in one project and compile and build the executable program. When input: ***2<ENTER>***, the result to run the program is _________________.

4. When input: ***3 4 8 6 7 5 9 10 2 1 <ENTER>***, the following program will print out _________________.

```
#include <stdio.h>
#define NMAX 8
int getIntArray(int a[], int nmax, int sentinel)
{
    int n = 0, temp;
    do {
        scanf("%d", &temp);
        if(temp==sentinel||n==nmax)break;
        a[n++] = temp;
    }while(1);
    return n;
}

void bs(int a[], int n)
{
    int lcv,temp,lastChange,limit = n-1;
    while (limit){
        lastChange = 0;
        for (lcv=0;lcv<limit;lcv++)
            if (a[lcv]>a[lcv+1]) {
                temp = a[lcv];
                a[lcv] = a[lcv+1];
                a[lcv+1] = temp;
                lastChange = lcv;
            }
        limit = lastChange;
    }
}

int main(void)
{
    int x[NMAX],i;
    int num = getIntArray(x, NMAX, 0);
    bs(x+num/4,num/2);
    for(i=0;i<num;i++)  printf("%d#", x[i]);
    return 0;
}
```

5. When input: ***2 1<ENTER>***, the output of the following program is _____.

```
#include<stdio.h>
int ack(int m,int n)
{
      int num;
      if(m == 0) return n+1;
      if(m>0 && n==0) {
             num = ack(m-1, 1);
             return num;
      }
      num = ack(m-1, ack(m, n-1));
      return num;
}
int main(void)
{
      int m, n;
      scanf("%d %d", &m, &n);
      printf("%d", ack(m, n));
      return 0;
}
```

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

1. For linked list ***h***, function ***struct node *process(struct node *h, int n, int m)*** deletes all of the nodes which data value is in the range ***[n, m]***; and function ***void printList(struct node *h)*** print all nodes' data in the linked list ***h***.Please complete the following code fragment.

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

struct node *process(struct node *h, int n, int m)
{
         struct node *p, *q;
        p=__(1)__;
        while (p!=NULL)  {
               if (p->data >=n && p->data <=m) {
                    if (p==h) { p=p->next;  free(h);  h=p; }
```

```
                else {
                    q->next=___(2)___;
                    free(p);
                    p=q->next;
                }
            } else  {
                q=__(3)__;
                p=p->next;
            }
        }
        return ____(4)____;
    }

    void printList(struct node *h)
    {
        struct node *p=h;

        while (p!=NULL) {
            printf("%d " , p->data);
            ____(5)_____;
        }
    }
```

2. The timer and char functions in the course graphics library are:

typedef void (*CharEventCallback) (char c);
typedef void (*TimerEventCallback) (int timerID);
void registerCharEvent(CharEventCallback callback);
void registerTimerEvent(TimerEventCallback callback);
void startTimer(int id,int timeinterval);
void cancelTimer(int id);

The code fragment below is to display ***"hello"*** every ***five seconds*** for ***three times*** when the space bar is pressed. Please fill in the blanks below.

```
void key_pressed(char c);
void timer_touch(int id);

void SetUp()
{
        …...
        registerCharEvent(____(6)_____);
        registerTimerEvent(____(7)____ );
        …...
}

void key_pressed(char ch)
{
        if ( ch == ' ' ) {
                ____(8)____;
        }
}

void timer_touch(int id)
{
        ___(9)___ int count = 0;
```

```
            if ( id == 0 ) {
                    printf("hello\n");
                    if ( ++count ==3 ) {
                            _______(10)_______;
                    }
            }
    }
```

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

1. A string consists of brackets (括号，含{,},[,],(,)). We can use stack to check whether these brackets are matching. For examples, “{[]()}” is a matching string, but “{[()}]” is not. Please:

(1) According to the following declarations, complete the stack’s operation functions ***Push()*** and ***Pop()***.

```
#define MAXSIZE 100
 struct Stack {
     char S[MAXSIZE];
     int top;
 };
 typedef struct Stack *StackP;

   StackP CreatStack()
   {
      StackP *sp;
     sp=(StackP)malloc(sizeof(struct Stack));
     sp->top= -1;
     return sp;
   }

 void Push(StackP sp, char c)
 {…….}

 char Pop(StackP sp)
 {…. }
```

(2) Complete the function ***int Check(char *BracketsStr)***, to check whether the brackets in the string ***BracketsStr*** are matching. If the brackets are matching, return ***1***, else return ***0***.

2. Given a polynomial $f_n(x)=a_0+a_1x+a_2x^2+\ldots+a_nx^n$, for a given ***x***, if calculate the value separately for each item, ***1+2+...+n=n(n+1)/2*** times multiplications will be needed and the efficiency is very low. If rewrite the formula as:

$$f_n(x)=((\ldots(((a_n)x + a_{n-1})x + a_{n-2})x +\ldots)x + a_1)x + a_0$$

a recursive algorithm can be used to calculate the value of polynomial function $f_n(x)$ more efficient.

(1) Please design the recursive algorithm of calculating the n-order polynomial $f_n(x)$, including the data structure and the two recursion-conditions.

(2) Write down the function of implementing the recursive algorithm above, and analyze the required number of multiplications.