

# 浙江大学 2015 - 2016 学年夏学期

## 《C 程序设计专题》课程期末考试答题卷

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: 2016 年 06 月 28 日, 考试时间: 120 分钟

试题号	一	二	三	四	总分	
满分	20	30	30	20		
得分					统分人 1	
阅卷人					统分人 2	

### Section 1: Single Choice(2 marks for each item, total 20 marks)

- 1 C                      2 D                      3 B                      4 C                      5 A  
 6 B                      7 A                      8 B                      9 A                      10 D

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

1. \_\_\_\_\_ 42 \_\_\_\_\_
2. (1) 'b'                      (2) 2                      (3) 3  
 \_\_\_\_\_
3. \_\_\_\_\_ c \_\_\_\_\_
4. \_\_\_\_\_ 3#4#5#6#7#8#9#10# \_\_\_\_\_
5. \_\_\_\_\_ 5 \_\_\_\_\_

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

- |                             |                                     |
|-----------------------------|-------------------------------------|
| (1) _____ h _____           | (2) _____ p->next _____             |
| (3) _____ p _____           | (4) _____ h _____                   |
| (5) _____ p=p->next _____   | (6) _____ key_pressed _____         |
| (7) _____ timer_touch _____ | (8) _____ startTimer(0, 5000) _____ |
| (9) _____ static _____      | (10) _____ cancelTimer(0) _____     |

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

1. (1)

```
void Push(StackP sp, char c)
{
    if (sp->top < MAXSIZE-1) sp->S[++sp->top] = c;
}

char Pop(StackP sp)
{
    if (sp->top >= 0) return sp->S[sp->top--];
}
```

..... 线 ..... 订 ..... 装 .....

1.(2)

```

int Check(char *BracketsStr)
{
    char *s=BracketsStr, c;
    StackP *sp;

    sp=CreatStack();
    while (*s) {
        switch (*s) {
            case '{' : case '[': case '(':
                Push(sp, *s);
                break;
            case '}':
                c=Pop(sp);
                if (c=='{') break;
                return 0;
            case ']':
                c=Pop(sp);
                if (c=='[') break;
                return 0;
            case ')':
                c=Pop(sp);
                if (c=='(') break;
                return 0;
            default:
                return 0;
        }
        s++;
    }
    return 1;
}
    
```

2.

(1) 算法设计

定义数组  $A[n+1]$  保存  $n$  次多项式系数，其中， $A[0]$  存储  $a_n$ ，...， $A[n]$  存储  $a_0$

递归出口：

当  $n=0$  时， $f_0(x)=A[0]$

递归式：

$f_n(x) = x * f_{n-1}(x) + A[n]$

(2) 递归实现函数

```
float fn(float x, float A[], int n)
{
    if(n==0)
        return A[0];
    else
        return x*fn(x,A,n-1)+A[n];
}
```

计算  $n$  次多项式的值所需乘法次数是  $n$