

浙江大学 2019 - 2020 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)

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

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 A 4 C 5 D
 6 A/C 7 D 8 D 9 D 10 D

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

1. (1) The program may exit abnormally.

(2) void (*p)(int, int (*)[10]);

2. 332123321
- _____

3. 0->1->3->5->6->7->9->10->14->23->34->44->59->70
- _____

4. 100
- _____

5. Flash a circle drawn in the center of a window once every 500 milliseconds.
The ESCAPE key is used as a switch to toggle the blink.
- _____

6. W0T#W0U#W0V#X0T#X0U#X0V#Y0T#Y0U#Y0V#
- _____

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

(1)	char *, int	(2)	SwitchingCenter *, char *
(3)	int	(4)	SwitchingCenter
(5)	(Queue*)malloc(sizeof(Queue))	(6)	p=q->front; p!=q->rear; p=p->next, length++
(7)	NULL	(8)	q->rear=temp
(9)	q->front->next	(10)	stackADT
(11)	NewStack	(12)	char *
(13)	PushStack	(14)	TopStack
(15)	PopStack		

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

1.

```
void FractalTree(int n, double x, double y, double length, double
theta) {
    if (n > 0) {
        double radians = theta / 180.0 * PI;
        int dx = length * cos(radians);
        int dy = length * sin(radians);
        MovePen(x, y);
        DrawLine(dx, dy);
        FractalTree(n-1, x+dx, y+dy, length*0.75, theta + 15);
        FractalTree(n-1, x+dx, y+dy, length*0.75, theta - 15)
    }
}

void Main() {
    int n;
    double length;
    InitGraphics();
    OpenConsole();
    n = GetInteger();
    length = GetReal();
    CloseConsole();
    FractalTree(n, GetWindowWidth()/2.0, 0, length, 90);
    return;
}
```

2.

```
void RearrangeList(LinkList h){
    if(!h) return;
    ListNode *p = h;
    int even = 0;
    LinkList temp = CreateNode(0);
    ListNode *pt = temp;
    ListNode *r = NULL;
    while(p){
        if(even){
            pt->next = p;
            pt = p;
            r->next = p->next;
        }else{
            r = p;
        }
        p = p->next;
        even = 1 - even;
    }
    pt->next = NULL;
    r->next = temp->next;
}
```