院系________ 学号________ 姓名________ 任课教师________

装 订 线

浙江大学 2016－2017 学年夏学期

《C 程序设计专题》课程期末考试参考答案

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： 2017 年 06 月 29 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 A　　5 C

6 D　　7 B　　8 C　　9 A　　10 C

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

1. (1) 10 (2) typedef void (*KeyboardEventCallback)(int, int);

2. Answer:3

3. (1)The input is in the ascending order. (2) O(n)

4. 1->3->2->2->4->5

5. 7#11#13#

6. Press mouse’s left button down and move mouse to move the circle.

   Release the mouse’s left button to stop moving the circle.

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

(1) sizeof(int)*num

(2) total = 0;

(3) i < num

(4) count[input_array[i]]++

(5) array,num

(6) -1

(7) MAX_STACK-1

(8) ++sp

(9) sp == -1

(10) Stack[sp--]

(11) A

(12) p->next != NULL

(13) p->data == B->data

(14) q->next = p->next

(15) B = B->next

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

1.【Version 1】

```
void ksort(int arr[], int n, int k) /*Bubble sort: descending order.*/
{
    int outloop, inloop, temp;

    for (outloop = 0; outloop < k; outloop++) {
        for (inloop = 0; inloop < n - 1 - outloop; inloop++) {
            if (arr[inloop] < arr[inloop+1]) {
                temp = arr[inloop];
                arr[inloop] = arr[inloop+1];
                arr[inloop+1] = temp;
            }
        }
    }
    printf("The %d'th smallest element is : %d\n", k, arr[n-k]);
    return;
}
```

【Version 2】

```
void ksort(int arr[], int n, int k) /*Selection sort: ascending order.*/
{
    int outloop, inloop, mindex, temp, i;
    for (outloop = 0; outloop < k; outloop++) {
        mindex = outloop;
        for (inloop = outloop+1; inloop < n; inloop++) {
            if (arr[mindex] > arr[inloop]) mindex = inloop;
        }
        if (mindex != outloop) {
            temp = arr[outloop];
            arr[outloop] = arr[mindex];
            arr[mindex] = temp;
        }
    }
    printf("The %d'th smallest element is : %d\n", k, arr[k-1]);
    return;
}
```

2.

```
void RecursiveSubsets(int set[], bool flag[], int start, int end)
{
    int i;
    if(start <= end) {
        flag[start] = FALSE; // pick the a[start]
        RecursiveSubsets(set, flag, start+1, end);
        flag[start] = TRUE; // not pick the a[start]
        RecursiveSubsets(set, flag, start+1, end);
    } else {
        printf("{ ");
        for(i = 0; i <= end; i++)
            if (flag[i]) printf("%d ", array[i]);
        printf("}\n");
    }
    return;
}
```