数据结构作业

第一章

1.8

1. $n-1$
2. n
3. $n-1$
4. $\frac{1}{2}n(n+1)$
5. $\frac{1}{6}n(n+1)(n+2)$
6. 如果@是指`if(i>j)`，n；如果是指`j++;`，$\lceil \frac{n}{2} \rceil$
7. $\lfloor \sqrt{n} \rfloor$
8. 如果@是指`if(x>100)`，1100；如果是指`x-=10;y=--;`，100

1.9

$o(\log_2 n)$

`count` = $\log_2 n - 2$

1.12

1. 对
2. 错
3. 错
4. 对
5. 错

1.16

```c
#include <stdio.h>

void swap(int *a, int *b);
void max(int *array, int len);
void array_print(int *array, int len);

void main(){
    int n = 3;
    int array[n];
    //输入
    printf("Please input %d integers, press enter after each one:\n", n);
    for (int i=0; i< n; i++){
        scanf("%d", &array[i]);
    }
    //排序
    max(array, n);
    //输出
    printf("rank result:\n");
    array_print(array, n);
    return;
}

void swap(int *a, int *b){
    //交换a,b
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
    return;
}

void max(int *array, int len){
    //输入长为n的int型数组，进行从大到小排序
    for (int i = 0; i < len-1; i++){
        for (int j = i+1; j < len; j++){
            if (array[i] < array[j]){
                swap(&array[i], &array[j]);
            }
        }
    }
    return;
}

void array_print(int *array, int len){
    //打印长度为len的整型数组
    for (int i=0; i < len; i++){
        printf("%d ", array[i]);
    }
    printf("\n");
    return;
}
```

1.19

```c
#include <stdio.h>
#include <stdlib.h>
#include <limits.h>         //获取整型数上界INT_MAX

void array_print(int *array, int len);

void main(){
    int n = 10;             //要求个数的上界
    int ARRAYSIZE = 20;     //数组长度，算了一下大概十几就爆掉了
    int array[ARRAYSIZE];

    //生成数组
    for (int i=1; i <= n; i++){
        //数组越界判断
        if (i > ARRAYSIZE){
            printf("Out of bounds when i = %d,\n", i);
            printf("The result before that:\n");
            array_print(array, i-1);
            exit(-1);
        }
        else if (i == 1){
            array[i-1] = 1 * 2;
        }
        else{
            //整值溢出判断
            if (array[i-2] <= INT_MAX/2/i){
                array[i-1] = array[i-2] * 2 * i;
            }
            else{
                printf("Overflow when i = %d,\n", i);
                printf("The result before that:\n");
                array_print(array, i-1);
                exit(-2);
            }
        }
    }

    //输出数组
    array_print(array, n);

    return;
}

void array_print(int *array, int len){
    //打印整型数组
    for (int i=0; i < len; i++){
        printf("%d ", array[i]);
    }
    printf("\n");
    return;
}
```

1.20

```c
#include <stdio.h>

double poly(double* a, int n, double x);

void main(){
    //输入多项式阶数
    int n;
    printf("Please input the order of polynomial:\n");
    scanf("%d", &n);

    //输入多项式系数
    double a[n];
    printf("Please input coefficients of polynomial, press enter after each one:\n");
    for (int i=0; i < n; i++){
        printf("a[%d]: ", i);
        scanf("%lf", &a[i]);
    }
    //展示多项式
    printf("The polynomail is:\n");
    for (int i=0; i < n; i++){
        if (i == 0){
            printf("%lf", a[0]);
        }
        else{
            printf(" + %lf*x^%d", a[i], i);
        }
    }
    printf("\n");

    //输入x
    double x;
    printf("Please input the x:\n");
    scanf("%lf", &x);

    //计算并输出多项式的结果
    printf("The result:\n%lf", poly(a, n, x));
    return;
}

double poly(double* a, int n, double x){
    //内部定义函数, 进行递归
    double __poly(double* a, int n, double x, int i){
        if (i == 0){
            return a[0];
        }
        else if (i < n){
            return a[i] + __poly(a, n, x, i - 1) * x;
        }
        else{
            printf("Out of bound\n");
        }
    }

    //调用内部函数__poly并返回值
```

```
    return __poly(a, n, x, n-1);
}
```