智能测试2334答案

学年：2018-2019　　学期：第1学期　　科目：编程语言　　满分：100分

1、单选题

1. D (2分)
2. B (2分)
3. C (2分)
4. A (2分)
5. C (2分)

2、多选题

1. BC (4分)
2. ACD (4分)
3. CD (4分)
4. ABC (4分)
5. ACFG (4分)

3、填空题

1. 88 (4分)
2. text-align:center (4分)
3. 20 (4分)
4. position: static ; 则上边距为(20)px 静态定位 top值无效 (4分)
   position: relative ; 则上边距为(30) px
   移动的时候会包括margin
   position: absolute ; 则上边距为(30) px
   移动的时候会包括margin
   position: fixed ; 则上边距为(30) px 固定定位的margin也会生效
   移动的时候也会包括margin
   position: sticky ; 则上边距为(20)
   px，页面滚动起来为（10）px，margin会无效；页面没滚动的
   时候是静态定位
5. 123 (4分)

4、解答题

(10分)

1.
```
#include <bits/stdc++.h>
using namespace std;//,
int main() {
int n;
cin >> n;
vector<int> x(n, 0);
vector<int> l(n, 0);
vector<int> r(n, 0);
for(int i = 0; i < n; i++) scanf("%d", &x[i]);
for(int i = 1; i < n; i++) {
if(x[i] > x[i - 1]) {
l[i] = l[i - 1] + 1;
}
}
for(int i = n - 2; i >= 0; i--) {
if(x[i] > x[i + 1]) {
r[i] = r[i + 1] + 1;
}
}
int mx = 0, ll = -1, rr = -1;
for(int i = 0; i < n; i++) {
if(l[i] > 0 && r[i] > 0 && l[i] + r[i] > mx) {
mx = l[i] + r[i];
ll = i - l[i];
rr = i + r[i];
}
}
cout << ll << " " << rr << endl;
return 0;
}
```

2. 1. sleep是Thread类的方法，是线程用来
控制自身流程的。wait是Object类的方法，用来线程间的通
信，这个方法会使当前拥有该对象锁的进程等待知道其他线程，主要是用于不同线程之间

2.
每个对象都有一个锁来控制同步访问。Synchronized关键字可以和对象的锁交互，来实现
步。sleep方法没有释放锁，而wait方法释放了锁，使得其他线程可以使用同步控制块或者

3.
wait只能在同步控制方法或者同步控制块里面使用，而sleep可以在任何地方使用

4. sleep必须捕获异常，而wait不需要捕获异常

(10分)

(10分)

3.

```
#include <bits/stdc++.h>
using namespace std;//set,setscanf
int main() {
int n, m;
int x;
bool first = true;
set<int> a;
scanf("%d", &n);
for (int i = 0; i < n; ++i) scanf("%d", &x),
a.insert(x);
scanf("%d", &m);
for (int i = 0; i < m; ++i) {
scanf("%d", &x);
if (a.find(x) != a.end()) {
if (first) first = false; else printf(" ");
printf("%d", x);
}
}
return 0;
}
```

4. 此题应该避免使用递归的方法，因为当count较大时，递归的方法耗时较长。
故考虑使用迭代法，可以使用数组记录每一项。
但此题只需要用到前面两项，从节约空间的角度讲不需要开辟数组。

```
function getNthFibonacci(count) {
if(count<0) return 0;
if(count<=1) return 1;
var first = 1;
var second = 1;
var third = 0;
for(var i = 2; i <= count; i++) {
third = first + second;
first = second;
second = third;
}
return third;
}
```

(10分)

(10分)

5. HTML:

```
<input type="text" />
```

JavaScript:

```
document.getElementsByTagName('input')[0].onkeyup =
function(){
if(this.value && !/^d+$/.test(this.value)){
this.style.color = 'red';
}
else{
this.style.color = 'black';
}
 }
```