

1. 【NOIP2008】

```

01 #include<iostream>
02 using namespace std;
03 void foo(int a,int b,int c)
04 {
05     if (a>b)
06         foo(c,a,b);
07     else
08         cout<<a<<' '<<b<<' '<<c<<endl;
09 }
10 int main()
11 {
12     int a,b,c;
13     cin>>a>>b>>c;
14     foo(a,b,c);
15     return 0;
16 }

```

●判断题

- (1)该程序只输出一行。 ()
- (2)如果输入的三个数都相同,程序会运行错误。 ()
- (3)如果输入 3 1 2,输出 2,3,1。 ()
- (4)如果输入 3 2 1,程序会超时。 ()

●选择题

- (5)输入 9 10 8,输出为()。
 A. 9,10,8 B. 9,8,10 C. 10,8,9 D. 8,9,10
- (6)令 n 代表输入变量 a 、 b 、 c 的次数,则 $n=3$ 该程序的时间复杂度为()。
 A. $O(n)$ B. $O(\log n)$ C. $O(\sqrt{\log n})$ D. $O(\log(n\sqrt{n}))$

2. 【NOIP2008】

```

01 #include<iostream>
02 using namespace std;
03 void f(int a, int b, int c)
04 {
05     cout<<a<<b<<c<<' / ';
06     if (b<c)
07         f(a, c, b);
08     else if (a<b)
09     {
10         if (a<c)
11             f(c, a, b);
12         else
13             f(b, c, a);
14     }
15 }
16 int main()
17 {
18     int a, b, c;
19     cin>>a>>b>>c;
20     f(a, b, c);
21     cout<<endl;
22     return 0;
23 }

```

●判断题

- (1)程序可能没有输出。 ()
- (2)程序可能会死循环。 ()
- (3)输入 10 10 10 程序会运行错误。 ()
- (4)输入 258 114514 1919810 程序会运行错误。 ()

●选择题

- (5)如果 $n=3$ 代表输入变量个数 $n=3$, 那么该程序时间复杂度是()。
 A. $O(2^n)$ B. $O(n!)$ C. $O(n^2)$ D. $O(n^n)$
- (6)如果输入 1 3 2, 则输出是()。
 A. 132/213/231/321/312/ B. 132/213/231/312/321/
 C. 213/132/231/321/312/ D. 213/132/231/312/321/

8. 【NOIP2014】

```
01 #include<iostream>
02 using namespace std;
03 int fun(int n){
04     if (n==1) return 1;
05     if (n==2) return 2;
06     return fun(n-2) - fun(n-1);
07 }
08 int main(){
09     int n;
10     cin>>n;
11     cout<<fun(n)<<endl;
12     return 0;
13 }
```

●判断题

- (1)输入 114514 时在普通计算机上程序运行时间不会超过 1s。 ()
- (2)输入 0 程序不会出现运行错误。 ()
- (3)该程序开启 O2 不会出现错误。 ()
- (4)输入 6,输出 7。 ()

●选择题

- (5)时间复杂度为()。
- A. $O(2^n)$ B. $O(n^n)$ C. $O(n^{\log n})$ D. $O(n)$
- (6)输入 7 时输出()。
- A. -11 B. 11 C. -10 D. 10

4. 【NOIP2010】

```
01 #include<iostream>
02 using namespace std;
03 const int NUM=5;
04 int r(int n)
05 {
06     int i;
07     if (n<=NUM)
08         return n;
09     for (i=1;i<=NUM;i++)
10         if (r(n-i)<0)
11             return i;
12     return -1;
13 }
14 int main()
15 {
16     int n;
17     cin>>n;
18     cout<<r(n)<<endl;
19     return 0;
20 }
```

●判断题

- (1)将第 4 行的 int 改为 unsigned,答案不会错误。
- (2)程序开启 O2 优化不会返回错误。
- (3)如果输入 -1,程序会输出 -1。
- (4)该问题 $r(n)$ 的值没有规律。

()
()
()
()

●选择题

- (5)如果输入 7,程序会输出()。

A. -1 B. 5 C. 1 D. 3

- (6)如果输入 16,程序会输出()。

A. 16 B. -1 C. 4 D. 1

5. 【NOIP2011】

```
01 #include<iostream>
02 using namespace std;
03 int solve(int n,intm) {
04     int i,sum;
05     if (m==1) return 1;
06     sum=0;
07     for (i=1;i<n;i++)
08         sum+=solve(i,m-1);
09     return sum;
10 }
11 int main() {
12     int n,m;
13     scanf('%d%d",&n,&m);
14     cout<<solve(n,m)<<endl;
15     return 0;
16 }
```

●判断题

- (1)将第 7 行 <改成<=程序会出现运行错误。 ()
- (2)本题使用 C++98 编译不会出现编译错误。 ()
- (3)本题输入 0 0 不会出现运行错误。 ()
- (4)本题不可能输出 0。 ()

●选择题

- (5)如果输入 7 4,输出()。
- A. 11 B. 20 C. 21 D. 12
- (6)如果输入 10 0,输出()。
- A. 0 B. 1 C. 2 D. 3

6. 【NOIP2011】

```
01 #include<iostream>
02 using namespace std;
03 int n;
04 void f2(int x,int y);
05 void f1(int x,int y)
06 {
07     if (x<n)
08         f2(y,x+y);
09 }
10 void f2(int x,int y)
11 {
12     cout<<x<<' ';
13     f1(y,x+y);
14 }
15 int main()
16 {
17     cin>>n;
18     f1(0,1);
19     return 0;
20 }
```

●判断题

- (1)把 04 行删除,程序不会出现编译错误。 ()
- (2)输入为 10,输出为 125。 ()
- (3)把 07 行的“ $x < n$ ”改为“ $x \leq n$ ”,程序运行结果会出现改变。 ()
- (4)该程序的时间复杂度为 $O(n)$ 。 ()

●选择题

- (5)输入为 35 时,输出为()。

A. 1 2 5 13 34

B. 1 2 5 13

C. 1 2 13

D. 2 5 13

- (6)输出为 1 2 5 时, n 的值可以为()。

A. 3

B. 5

C. 9

D. 11

7. 【NOIP2015】

```
01 #include<iostream>
02 using namespace std;
03 int fun(int n,int fromPos,int toPos){
04     int t,tot;
05     if (n==0) return 0;
06     for (t=1;t<=3;t++)
07         if (t!=fromPos&& t!=toPos) break;
08     tot=0;
09     tot+=fun(n-1,fromPos,t);
10     tot++;
11     tot+=fun(n-1,t,toPos);
12     return tot;
13 }
14 int main(){
15     int n;
16     cin>>n;
17     cout<<fun(n,1,3)<<endl;
18     return 0;
19 }
```

●判断题

- (1)当 n 为小于 1000 的正整数时,将第 9 行和第 11 行一起去掉,程序输出结果为 1。 ()
- (2)当 n 为小于 1000 的正整数时,将第 9 行或第 11 行中其中一行去掉,程序输出 n 。 ()
- (3)函数中的 $fromPos$ 与 $toPos$ 与答案无关。 ()
- (4)该程序的时间复杂度为 $O(2^n)$ 。 ()

●选择题

(5) $fun(5,1,3)$ 的值为()。

- A. 31 B. 23 C. 66 D. 9

(6) $fun(n,1,3)$ 的通项公式为()。

- A. $fun(n-1,1,3) * 2 + 1$ B. $2 * n$
- C. $2^n - 1$ D. $\left(\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right) / \sqrt{5}$

8. 【NOIP2017】

```
01 #include<iostream>
02 using namespace std;
03 int g(int m,int n,int x){
04     int ans=0;
05     int i;
06     if (n==1)
07         return 1;
08     for (i=x;i<=m/n;i++)
09         ans+=g(m-i,n-1,i);
10     return ans;
11 }
12 int main(){
13     int t,m,n;
14     cin>>m>>n;
15     cout<<g(m,n,0)<<endl;
16     return 0;
17 }
```

●判断题

- (1)把第 6 行去掉,程序总会输出 1。 ()
- (2)将第 4 行的内容接在第 2 行的后面,程序输出与原样不同。 ()
- (3)把第 5 行去掉,程序会编译错误。 ()
- (4)此程序的功能是求将 m 个无序物品无序地分成 n 份的方案数。 ()

●选择题

- (5)输入 7 3,则输出()。
- A. 2 B. 4 C. 6 D. 8
- (6)输入 5 5,则输出()。
- A. 1 B. 3 C. 5 D. 7

9. 【NOIP2018】

```
01 #include<iostream>
02 using namespace std;
03 int n,m;
04 int findans(int n,int m){
05     if (n==0) return m;
06     if (m==0) return n%3;
07     return findans(n-1,m)-findans(n,m-1)+findans(n-1,m-1);
08 }
09 int main() {
10     cin>>n>>m;
11     cout<<findans(n,m)<<endl;
12     return 0;
13 }
```

●判断题

- (1)将第 5 行和第 6 行一起去掉,程序会出现死循环。 ()
- (2)当输入的 n, m 的绝对值在 1000 以内时,程序一定会正常运行。 ()
- (3)若将该递归程序执行记忆化,则程序的时间复杂度为 $O(nm)$ 。 ()
- (4)将第 3 行接在第 9 行后,则程序会编译错误。 ()

●选择题

(5)输入 5 6,则输出为()。

A. 5

B. 6

C. 7

D. 8

(6)输入 2 4,则输出为()。

A. 5

B. 6

C. 7

D. 8

10.【NOIP2014】

```
01 #include<iostream>
02 using namespace std;
03 int fun(int n,int minNum,int maxNum) {
04     int tot,i;
05     if (n==0) return 1;
06     tot=0;
07     for (i=minNum;i<=maxNum;i++)
08         tot+=fun(n-1,i+1,maxNum);
09     return tot;
10 }
11 int main() {
12     int n;
13     cin>>n;
14     cout<<fun(n,1,6)<<endl;
15     return 0;
16 }
```

●判断题

(1)将第 5 行删掉,程序编译错误。

()

(2)当输入的 n 的绝对值在 1000 以内时,程序不一定能正常运行。

()

(3)将第 4 行的内容接在第 2 行后,程序输出与原样不同。

()

(4)将第 4 行的内容去掉,程序将会运行错误。

()

●选择题

(5)fun(2,1,6)的值为()。

A. 15

B. 10

C. 6

D. 3

(6)fun(3,1,6)的值为()。

A. 20

B. 10

C. 4

D. 1