

## LESSON\_11 参考试题

### 一、 选择题

1. (样题) 以下哪个递推关系式表示斐波那契数列( )。

A.  $F(n)=F(n-1) + F(n-2) + F(n-3)$   
B.  $F(n) = F(n-1) + F(n-2)$   
C.  $F(n) = F(n-1) * F(n-2)$   
D.  $F(n) = F(n-1) / F(n-2)$

【答案】 B

2. (2023年9月) 执行以下C++语言程序后, 输出结果是( )。

```
#include <iostream>
using namespace std;
int main(){
    int fib[10];
    fib[0]=0;
    fib[1]=1;
    for(int i=2; i<10; i++)
        fib[i]=fib[i-1]+fib[i-2];
    cout<<fib[10]<<endl;
    return 0;
}
```

A. 0  
B. 5  
C. 55  
D. 无法确定

【答案】 D

3. (样题) 在下列代码的横线处填写( ), 可以使得输出是“21”。

```
#include <iostream>
using namespace std;
int main(){
    int a[5];
    a[0]=1;
    for(int i=1; i<5; i++)
        a[i]=a[i-1]*2;
    int sum = 0;
    for(int i=0; i<5; _____)
        sum+=a[i];
    cout<<sum<<endl;
```

```

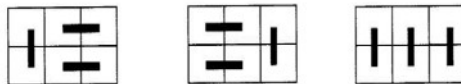
    return 0;
}

```

- A.  $i++$
- B.  $i+=2$
- C.  $i+=3$
- D.  $i/=2$

【答案】B

4. 有  $2*n$  的一个长方形方格，用一个  $1*2$  的骨牌铺满方格，当  $n=5$  时，铺法总数为多少( )。



- A. 5
- B. 6
- C. 7
- D. 8

【答案】D

5. 设有一个共有  $n$  级台阶的楼梯，某人每步可走 1 级，也可以走 2 级，用递推的方式可以计算出某人从底层开始走完全部台阶的走法。例如，当  $n=3$  时，共有 3 种走法，即  $1+1+1$ 、 $1+2$ 、 $2+1$ 。当  $n=6$  时，从底层开始走完全部台阶的走法共有多少种？( )

- A. 12
- B. 13
- C. 14
- D. 15

【答案】B

6. 已知 Pell 数列定义如下,第 5 项的值是( )。

$$a_1=1$$

$$a_2=2$$

...

$$a_n=2*a_{n-1}+a_{n-2}$$

- A. 29
- B. 30
- C. 12
- D. 32

【答案】A

## 二、 判断题

1. (样题) 递推算法通常有初始值。

【答案】正确

2. (2023年6月) 数列 1, 1, 2, 3, 5, 8 ... 是以意大利数学家列昂纳多·斐波那契命名的数列, 从第三个数开始, 每个数是前面两项之和。如果计算该数列的第  $n$  项 (其中  $n > 3$ )  $\text{fib}(n)$ , 我们采用如下方法: ① 令  $\text{fib}(1)=\text{fib}(2)=1$  ②用循环 for  $i=3$  to  $n$  分别计算  $f(i)$  ③输出  $\text{fib}(n)$ 。这体现了递推的编程思想。

【答案】正确

### 三、编程题

#### 1. 矩阵行走

【问题描述】

给定一个  $n \times m$  格的矩阵, 问从**左上角**走到**右下角**有多少条不同的路径?

每步只能**向下走**或**向右走**, 且只能沿着矩阵的边线行走。

【输入描述】

一行两个正整数  $n, m$ ,  $1 \leq n \leq 10$   $1 \leq m \leq 4$ 。

【输出描述】

路径数目  $t$

【样例输入 1】

1 1

【样例输出 1】

2

【样例输入 2】

2 2

【样例输出 2】

6

【样例输入 3】

6 4

【样例输出 3】

210

【参考代码】

```

#include <iostream>
using namespace std;
int a[15][15];
int main(){
    int n,m;
    cin>>n>>m;
    // 第一行设置为 1
    for(int i=0;i<=n;i++)a[i][0]=1;
    // 第一列设置为 1
    for(int j=0;j<=m;j++)a[0][j]=1;
    // 根据递推公式计算路径
    for(int i=1;i<=n;i++){
        for(int j=1;j<=m;j++){
            a[i][j]=a[i-1][j]+a[i][j-1];
        }
    }
    cout<<a[n][m];
    return 0;
}

```