**Q1 斐波那契数列（基础题）**

#include <stdio.h>

int fib(int n);

int main()

{

int n;

scanf("%d", &n);

printf("%d\n", fib(n));

return 0;

}

int fib(int n)

{

int f1 = 1;

int f2 = 1;

int s;

if (n <= 0)

return -1;

if (n == 1 || n == 2)

return 1;

else

{

for (int i = 3; i <= n; i++)

{

s = f1 + f2;

f1 = f2;

f2 = s;

}

return s;

}

}

【判断题】

1.当输入2时，fib函数会返回2。 （×）

2.如果将fib函数的循环中将i<=n改成i<n对返回结果无影响。（×）

3.如果将fib函数中的else去掉，返回结果无影响。（√）

4.如果输入负整数，程序会报错。（×）

【选择题】

1.当输入n=10时，fib函数返回的值是多少？

A.55 B. 89 C. 34 D. 144

答案：A

2.简述斐波那契数列的特点

A．第一个数为1，第二个数为2

B. 每一项都是前两项之和

C. 每一项都是前一项的两倍

D. 每一项都是前一项加上2

答案：B

**Q2 哥德巴赫猜想（基础题）**

#include <iostream>

#include <math.h>

using namespace std;

// 声明函数

int prime(int);

int main()

{

int x;

cin >> x;

for (int i = 2; i < x; i++)

{

// 判断第一个数是否为素数

if (prime(i) == 1)

{

// 判断第二个数是否为素数 且 满足两数之和为偶数x

if ((prime(x - i) == 1))

{

// 满足，输出两个素数

cout << i << "\t" << x - i << endl;

}

}

}

return 0;

}

// 判断素数

int prime(int n)

{

for (int i = 2; i < n; i++)

{

if (n % i == 0) // n可以整出除1和本身的数（不为素数）

return 0;

}

return 1; // n为素数

}

【判断题】

1.如果将prime函数的类型换成bool类型，对程序没有影响。（√）

2.如果将prime函数中for循环里的i<=sqrt(n)替换成i<n，对程序结果无影响，但会影响程序的运行效率。（√）

3.当输入2时，prime函数会返回0。（×）

【选择题】

1.输入4，程序返回（）

A. 2 2

B. 1 3

C. 0 4

D. 2 2和1 3

答案A

2.如果在主函数for循环中在嵌套一个循环for (int j = 2; j < x; j++)，且将后面的if判断替换成if ((prime(j) == 1) && (i + j == x))，输入8，程序的输出是什么？

A. 3 5

B. 4 4

C. 3 5和5 3

D. 3 5和5 3和4 4

答案C

**Q3 组合数（搜索）**

#include<bits/stdc++.h>

using namespace std;

// n个数字里取m个

int n, m;

int a[1000], b[1000];

void dfs(int k){

for(int i = a[k-1]; i <= n; i++){

if(b[i] == 0){

a[k] = i;

b[i] = 1;

if (k == m){

for(int i = 1; i <= m; i++){

printf("%d ", a[i]);

}

printf("\n");

}

else

dfs(k+1);

b[i] = 0;

}

}

}

int main(){

scanf("%d%d", &n, &m);

a[0] = 1;

dfs(1);

}

【判断题】

1. 当 k 等于 m 时，表示已经选取了m个数字，此时会打印出这m个数字。（√）

2. 上述代码的时间复杂度为 O(n \* m!)。（√）

3. 每次递归调用 dfs 函数时，都会尝试选取一个未被选取的数字。（√）

【选择题】

1. 以下哪个数组用来标记某个数字是否被选取？

A. a[] B. b[] C. Both A and B D. Neither A nor B

答案 B

2. 以下哪个数组用来存储最终选取的m个数字？

A. a[] B. b[] C. Both A and B D. Neither A nor B

答案 A

**Q4 国王的金矿（动态规划）**

#include <bits/stdc++.h>

using namespace std;

// 最优子结构，子问题重叠，边界，子问题独立 转移方程

int f[300][300], w[300], c[300];

// f[i][j]---第i个金矿被人挖走后可获得的最大价值

// w[i]---第i个金矿需要的人数

// c[i]---第i个金矿的价值

int main()

{

int m, n;

// m个金矿，n个人

cin >> m >> n;

for (int i = 1; i <= m; i++)

{

// cin>>w[i]>>c[i];

scanf("%d %d", &w[i], &c[i]);

}

for (int i = 1; i <= m; i++)

{

for (int j = 1; j <= n; j++)

{

if (w[i] <= j)

{

// 人数够时

f[i][j] = max(f[i - 1][j], f[i - 1][j - w[i]] + c[i]);

}

else

{

f[i][j] = f[i - 1][j];

}

}

}

cout << f[m][n];

return 0;

}

【判断题】

1. 当w[i]（第i个金矿需要的人数）小于等于j（当前可用的人数）时，会更新f[i][j]的值。（√）

2. 如果当前金矿需要的人数w[i]大于当前可用的人数j，则f[i][j]的值不会改变。（√）

3. c[i]表示的是第i个金矿的价值。（√）

4.我[i]表示的是i个金矿需要的人数（√）

【选择题】

1. 那个选项正确地描述了f[i][j]的含义？

A. 第i个人挖第j个金矿可以获得的最大价值

B. 第i个金矿被挖走后可获得的最大价值

C. 第i个人挖第j个金矿需要的人数

D. 第i个金矿的价值

答案 B

2. 在状态转移方程f[i][j] = max(f[i - 1][j], f[i - 1][j - w[i]] + c[i])中，f[i - 1][j]代表的是？

A. 前i-1个金矿在j个人中的最大价值

B. 前i-1个金矿在j-w[i]个人中的最大价值

C. 第i个金矿在j个人中的最大价值

D. 第i个金矿在j-w[i]个人中的最大价值

答案 A

**Q5 括号匹配（数据结构-栈）**

#include <bits/stdc++.h>

using namespace std;

// [(])

int main()

{

stack<char> st;

string str;

int flag = 1;

cin >> str;

for (int i = 0; i <= str.length(); i++)

{

if (str[i] == '(' || str[i] == '[' || str[i] == '{')

{

【①】

}

else

{

if (str[i] == ')')

{

if (st.top() == '(')

{

st.pop();

}

else

{

flag = 0;

break;

}

}

if (str[i] == ']')

{

if (st.top() == '[')

{

st.pop();

}

else

{

flag = 0;

break;

}

}

if (str[i] == '}')

{

if (st.top() == '{')

{

st.pop();

}

else

{

flag = 0;

break;

}

}

}

}

if (【②】 || (flag == 1 && !st.empty()))

cout << "Wrong";

else

{

cout << "OK";

}

return 0;

}

【判断题】

1. 当输入的字符串为"{}“时，程序会输出"Wrong”。（×）

2. 当输入的字符串为"([{}])]“时，程序会输出"Wrong”。（√）

3. 如果最后栈为空，则说明所有的括号都匹配。（×）

4. 当遇到右括号时，我们会检查栈顶元素是否与该右括号匹配，如果匹配则弹出栈顶元素。（√）

【选择题】

1. ①处的代码应填

A. st.push(str[i]);  
B. st.push(str[i])

C. flag = 1;

D. flag = 0;

答案 A

2. ②处的代码应填

A. st.push(str[i]);  
B.flag == 0

C.flag = 0

D.flag == 1

答案 B

**Q6 节约用电（贪心）**

#include<bits/stdc++.h>

using namespace std;

int a[100005];

int main(){

int n, m;

cin >> n >> m;

for(int i = 0; i < n; ++i){

cin >> a[i];

}

//先进行排序

sort(a, a + n);

int ans = 0;

int last = a[0];

for(int i = 1; i < n - 1; ++i){

if(a[i + 1] - last <= m){

//说明可以关掉

ans++;

}else{

last = a[i];

}

}

cout << ans;

return 0;

}

【判断题】

1. 在这个代码中，我们使用数组a来存储输入的灯的状态。（√）

2. 如果a[i+1] - last的值大于m，则我们将last更新为a[i]。（√）

3. 在for循环中，我们每次检查a[i+1] - last的值，如果小于等于m，则说明可以关掉。（√）

4. 当输入为：5 3，1 2 3 4 5时，程序的输出是3。（√）

【选择题】

1. 如果a[i+1] - last的值小于等于m，我们会进行的操作是？

A. 将ans加1

B. 将last更新为a[i]

C. 将ans更新为i+1

D. 将last更新为a[i+1]

答案 A

2当输入为：6 1，1 2 3 4 5 6时，程序的输出是？

A. 1 B. 2 C. 3 D. 4

答案 A