

出卷人：虞小江
审卷人：王晨晨

Q1【基础题】

```
1  #include <stdio.h>
2  bool isPrime(int x){//判断一个正整数是否为素数
3      if(x==0 || x==1){
4          return false;
5      }
6      for(int i=2;i*i<=x;i++){
7          if(x%i==0){
8              return false;
9          }
10     }
11     return true;
12 }
13
14 int main(){
15     int n;
16     int sum=0,count=0;
17     scanf("%d",&n);
18     for(int i=1;i<=n;i++){
19         if(isPrime(i)){
20             if(count==1){//计算第偶数个素数的和
21                 sum+=i;
22             }
23             count=1-count;
24         }
25     }
26     printf("%d",sum);
27     return 0;
28 }
```

【判断题】

- 1.输入整数 10，打印结果为 10。（√）
- 2.如果输入负整数，程序会报错。（×）
- 3.若将第 6 行 $i*i \leq x$ 改为 $i*i < x$ ，输入大于 5 的整数，输出结果不会发生改变。（×）
- 4.若将第 18 行 $i=1$ 改为 $i=2$ ，程序输出结果不会发生改变。（√）

【选择题】

- 5.若将 20 行 $count==1$ 改为 $count==0$ ，输入整数 15，输出结果为（B）
A.23 B.18 C.41 D.20
- 6.若将程序改为实现输出第 3 的倍数个的素数之和，在其他程序不变的情况下，
可以将第 20 行和第 23 行的标注的代码改为（D）
A. $count==2;count=2-count\%2;$ B. $count\%3==0;count++;$
C. $count==2;count=2-count\%3;$ D. $count\%3==2;count++;$

Q2 【基础题】

```
1  #include <stdio.h>
2  #include <string.h>
3  int main(){
4      char a[100];
5      scanf("%s",a);
6      int len=strlen(a);
7      for(int i=0;i<len;i++){
8          if(a[i]>='A'&& a[i]<='Z'){//判断字符是否为大写字母
9              if(i==0){
10                 a[i+1]='*';
11                 continue;
12             }
13             if(i==len-1){
14                 a[i-1]='*';
15                 continue;
16             }
17             a[i-1]='*';
18             a[i+1]='*';
19         }
20     }
21     printf("%s",a);
22     return 0;
23 }
```

【判断题】

- 1.若输入为“abc\n123”，那么\n后面的 123 将不会被输入进数组中。（×）
- 2.若输入为“aAZz”，那么输出结果将会是“*****”。（×）
- 3.若将第 8 行代码中判断条件改为 a[i]>='a'&&a[i]<='z'，那么将会使得小写字母的左边和右边的字符变为'*'。（√）
- 4.若将第 17 行代码与第 18 行交换，代码输出结果不变。（√）

【选择题】

- 5.如果输入中含有 3 个大写字母，则输出最少有（A）个'*'。
A.1 B.2 C.3 D.4
- 6.如果输入含有 2 个小写字母，5 个大写字母，则输出最多有（B）个'*'。
A.3 B.4 C.5 D.6

Q3 【搜索算法题】

```
1  #include<iostream>
2  using namespace std;
3  int a[1000];
4  int count=0;
5  int n,k;
6  void f(int x,int flag,int tsum){
7  if(x==n){
8      if(tsum+a[x]*flag==k){
9          count++;
10         }
11         return;
12     }
13     f(x+1,0,tsum+a[x]*flag);
14     f(x+1,1,tsum+a[x]*flag);
15 }
16 int main(){
17     scanf("%d %d",&n,&k);
18     for(int i=1;i<=n;i++){
19         scanf("%d",&a[i]);
20     }
21     f(0,0,0);
22     printf("%d",count);
23     return 0;
24 }
```

本题规定输入的 n 为正整数。

【判断题】

- 1.若将第 7 行的判断条件改为 $x==n-1$ ，运行结果不变。（×）
- 2.该搜索算法的时间复杂度为 $O(2^n)$ 。（√）
- 3.若将第 13 行以及第 21 行的第二个输入变量改为-1，则程序功能将改变。（√）

【选择题】

- 4.输入一个数 n ，再输入 $n+1$ 个 0，输出结果为（A）
A. 2^n B. 2^{n-1} C. n^2 D. n^2-1
- 5.若将第 8 行的判断条件改为 $(tsum+a[x]*flag)\%k==0$ ，则先输入 3 3，再输入 2 6 7，运行结果为（C）
A.0 B.1 C.2 D.3

Q4【模拟题】 [NOIP2003 普及组] 乒乓球

题目背景

国际乒联现在主席沙拉拉自从上任以来就立志于推行一系列改革，以推动乒乓球运动在全球的普及。其中 11 分制改革引起了很大的争议，有一部分球员因为无法适应新规则只能选择退役。华华就是其中一位，他退役之后走上了乒乓球研究工作，意图弄明白 11 分制和 21 分制对选手的不同影响。在开展他的研究之前，他首先需要对他多年比赛的统计数据进行一些分析，所以需要你的帮忙。

题目描述

华华通过以下方式进行分析，首先将比赛每个球的胜负列成一张表，然后分别计算在 11 分制和 21 分制下，双方的比赛结果（截至记录末尾）。

比如现在有这么一份记录，（其中 W 表示华华获得一分，L 表示华华对手获得一分）：

WWWWWWWWWWWWWWWWWWWWWWWWLW

在 11 分制下，此时比赛的结果是华华第一局 11 比 0 获胜，第二局 11 比 0 获胜，正在进行第三局，当前比分 1 比 1。而在 21 分制下，此时比赛结果是华华第一局 21 比 0 获胜，正在进行第二局，比分 2 比 1。如果一局比赛刚开始，则此时比分为 0 比 0。直到分差大于或者等于 2，才一局结束。

你的程序就是要对于一系列比赛信息的输入（WL 形式），输出正确的结果。

输入格式

每个输入文件包含若干行字符串，字符串有大写的 W、L 和 E 组成。其中 E 表示比赛信息结束，程序应该忽略 E 之后的所有内容。

输出格式

输出由两部分组成，每部分有若干行，每一行对应一局比赛的比分（按比赛信息输入顺序）。其中第一部分是 11 分制下的结果，第二部分是 21 分制下的结果，两部分之间由一个空行分隔。

```
1  #include<iostream>
2  #include<math.h>
3  using namespace std;
4  char str[10000];
5  int count=0;
6  void show(int n){
7      int a=0,b=0;
8      for(int i=0;i<cnt;i++){
9          if(str[i]=='W') a++;
10         if(str[i]=='L') b++;
11         if((a>=n || b>=n)&&fabs(a-b)>=2){
12             cout<<a<<":"<<b<<endl;
13             a=0;
14             b=0;
15         }
```

```

16     }
17     cout<<a<<":"<<b<<endl;
18 }
19
20 int main(){
21     char ch;
22     while(cin>>ch&&ch!='E'){
23         if(ch=='W' || ch=='L'){
24             str[cnt++]=ch;
25         }
26     }
27     show(11);
28     cout<<endl;
29     show(21);
30     return 0;
31 }

```

【判断题】

- 1.show 函数实现了输出几分制对应的比赛结果。（√）
- 2.若直接输入一个 E，将会没有输出结果。（×）
- 3.若将第 13 行与第 14 行代码放在 if 语句的外面，程序输出结果将不变。（×）

【选择题】

- 4.依次输入 10 个 W 和 10 个 L，再输入 2 个 W 和 1 个 E，则输出结果分别为（D）
 A.11:10 1:0;12:10 B.10:10 2:0;12:10
 C.12:10;12:10 D.12:10 0:0;12:10
- 5.若输入字符依次为 WL 循环并且以 W 开始，WE 结束（如 WLWLWLWLWE），那么两次输出结果的行数分别为（A）
 A.1;1 B.2;1 C.2;2 D.1;2

Q5【模拟题】 [NOIP2015 普及组] 扫雷游戏

题目描述

扫雷游戏是一款十分经典的单机小游戏。在 n 行 m 列的雷区中有一些格子含有地雷（称之为地雷格），其他格子不含地雷（称之为非地雷格）。玩家翻开一个非地雷格时，该格将会出现一个数字——提示周围格子中有多少个是地雷格。游戏的目标是在不翻出任何地雷格的条件下，找出所有的非地雷格。

现在给出 n 行 m 列的雷区中的地雷分布，要求计算出每个非地雷格周围的地雷格数。

注：一个格子的周围格子包括其上、下、左、右、左上、右上、左下、右下八个方向上与之直接相邻的格子。

输入格式

第一行是用一个空格隔开的两个整数 n 和 m ，分别表示雷区的行数和列数。

接下来 n 行，每行 m 个字符，描述了雷区中的地雷分布情况。字符 '*' 表示相应格子是地雷格，字符 '?' 表示相应格子是非地雷格。相邻字符之间无分隔符。

输出格式

输出文件包含 n 行，每行 m 个字符，描述整个雷区。用 '*' 表示地雷格，用周围的地雷个数表示非地雷格。相邻字符之间无分隔符。

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  char a[101][101];
4  int b[101][101];
5  int dx[8]={1,0,-1,0,-1,1,1,-1};
6  int dy[8]={0,1,0,-1,-1,-1,1,1};
7  int n,m;
8
9  void dfs(int x,int y)//调用函数将地雷周围的数加起来
10 {
11     int nx,ny,k;
12     for(k=0;k<8;k++){
13         nx=x+dx[k];
14         ny=y+dy[k];
15         if(①){
16             ②;
17         }
18     }
19 }
20 int main(){
21     cin>>n>>m;
22     for(int i=0;i<101;i++){
23         for(int j=0;j<101;j++){
```

```

24         ③ ;
25     }
26 }
27 for(int i=1;i<=n;i++){
28     for(int j=1;j<=m;j++){
29         cin>>a[i][j];
30         if ( ④ ){
31             ⑤ ;
32         }
33     }
34 }
35 for(int i=1;i<=n;i++){
36     for(int j=1;j<=m;j++){
37         if(a[i][j]=='*'){//判断是地雷就直接输出
38             cout<<a[i][j];
39         }
40         else{
41             cout<<b[i][j];//不是地雷输出答案
42         }
43     }
44     cout<<endl;
45 }
46 return 0;
47 }

```

【选择题】

根据程序所实现功能，填写标注位置所缺代码。

1.①处代码应填写为 (B)

- A. $nx \geq 0 \&\& nx \leq n \&\& ny \geq 0 \&\& ny \leq m$ B. $nx \geq 1 \&\& nx \leq n \&\& ny \geq 1 \&\& ny \leq m$
 C. $nx \geq 0 \&\& nx \leq m \&\& ny \geq 0 \&\& ny \leq n$ D. $nx \geq 1 \&\& nx \leq m \&\& ny \geq 1 \&\& ny \leq n$

2.②处代码应填写为 (A)

- A. $b[nx][ny]++$ B. $b[nx][ny]=1$
 C. $a[nx][ny]++$ D. $a[nx][ny]='*'$

3.③处代码应填写为 (C)

- A. $a[i][j]='*'$ B. $a[i][j]='?'$
 C. $b[i][j]=0$ D. $b[i][j]=1$

4.④处代码应填写为 (A)

- A. $a[i][j]=='*'$ B. $a[i][j]=='?'$
 C. $b[i][j]==1$ D. $b[i][j]==0$

5.⑤处代码应填写为 (C)

- A. $dfs(i-1,j-1)$ B. $dfs(i+1,j+1)$
 C. $dfs(i,j)$ D. $dfs(b[i],b[j])$

Q6 【数据结构-栈】

检查一段 C 语言代码的小括号()、中括号[]和大括号{}是否匹配。

若括号是匹配的，在第二行打印 YES，否则打印 NO。

在一行中输入一段 C 语言代码，本题保证长度不超过 1000 个字符（行末以换行符结束）。

```
1  #include<bits/stdc++.h>
2  #include<stack>
3  using namespace std;
4  int main(){
5  string s;
6  stack<char> st;
7  getline(cin,s);//输入字符串
8  int l=0,r=0;
9  bool flag=1;
10     for(int i=0;i<s.length();i++) {
11         if( ① ) {
12             l++;
13             st.push(s[i]);
14         }
15         if(s[i]=='{' || s[i]=='[' || s[i]=='}') {
16             r++;
17             if( ② ) {
18                 flag=0;
19                 continue;
20             }
21             char ch= ③ ;
22             if( ④ ){
23                 st.pop();
24             }else{
25                 flag=0;
26             }
27         }
28     }
29     if( ⑤ ) cout<<"YES\n";
30     else cout<<"NO\n";
31     return 0;
32 }
```

【选择题】

根据程序所实现功能，填写标注位置所缺代码。

1.①处代码应填写为 (A)

A.s[i]=='{' || s[i]=='[' || s[i]=='{' B.s[i]=='{' || s[i]=='[' || s[i]=='{'

C.s[i+1]=='(' || s[i+1]=='[' || s[i+1]=='{' D.(s[i]=='(' || s[i]=='[' || s[i]=='{')&&flag

2.②处代码应填写为 (C)

A.st.empty()&&flag B.flag

C.st.empty() D.!st.empty()

3.③处代码应填写为 (B)

A.s[i+1] B.st.top()

C.s[i] D.st.top()-1

4.④处代码应填写为 (A)

A.flag&&((ch=='('&&s[i]=='') || (ch=='['&&s[i]=='') || (ch=='{'&&s[i]==''))

B.(ch=='('&&s[i]=='') || (ch=='['&&s[i]=='') || (ch=='{'&&s[i]==''))

C.flag&&((ch=='(' || s[i]=='') || (ch=='[' || s[i]=='') || (ch=='{' || s[i]==''))

D.flag

5.⑤处代码应填写为 (D)

A.flag B.st.empty()

C.!st.empty() D.flag&&st.empty()