

提取()或[]的方法

法一：会读取空格

[具体题目：卖个萌](#)

前提：带有[]或()的一串字符已经由getline()读入s中

方法：设定两个pointer，一个j用来记录'['所在的位置，一个k记录']'所在的位置。当找到i之后的j时，将子串str.substr(j+1,k-j-1)赋给字符串型vector，然后break;然后j++，直到找到下一个 '['，k也从上一个位置进行k++，直到下一个']'，重复以上过程，最后将一个字符串都存入以字符串型vector为元素的vector，也就是二维动态数组。

什么字符都会读入，如果有空格也会读入。

下面的模板一个字都不要改

```
int j = 0, k = 0;
vector<string> row;
while (j < s.length()) {
    if (s[j] == '[') {
        while (k++ < s.length()) { //防治越界
            if (s[k] == ']') {
                row.push_back(s.substr(j + 1, k - j - 1));
                break;
            }
        }
    }
    j++;
}
```

法二：可以指定读入

```
for(int j=0;j<s.size();j++){
    string temp;
    while(s[j]!=' '){
        if(s[j]>='a'&&s[j]<='z') temp+=s[j]; //可以指定读入
        j++;
    }
}
```

法三：

[具体题目：选择题](#)

```
for(int i = 0; i < n; i++) {
    int score = 0;
    scanf("\n"); //因为上面和下面有scanf("%c")所以要吃掉回车
    for(int j = 0; j < m; j++) {
        if(j != 0) scanf(" "); //读取括号之间的空格，即(a...) (b...)
        scanf("%d", &k);
        set<char> st;
        char c;
        for(int l = 0; l < k; l++) {
            scanf(" %c", &c); //读取字母之间的空格
            st.insert(c);
        }
    }
}
```

```

    }
    scanf("%c");//读取 ) ，不将其放入元素，只是读一下
}
}

```

去掉字符串中所有空格

不返回字符串：

```

void trim(string &s){
    int index = 0;
    if( !s.empty()){
        while( (index = s.find(' ',index)) != string::npos){
            s.erase(index,1);
        }
    }
}

```

返回字符串：

```

string trim(string s){
    int index = 0;
    if(!s.empty()){
        while( (index = s.find(' ',index)) != string::npos){
            s.erase(index,1);
        }
    }
    return s;
}

```

二维vector的遍历方法

```

for (int i = 0; i < vect.size(); i++){
    for (int j = 0; j < vect[i].size(); j++){
        cout << vect[i][j] << " ";
    }
    cout << endl;
}

```