

作業一

程式說明

程式要求:

1. 將符號排除留下英文字母（空格需保留），並將符號「*」後的小寫字母轉換成大寫字母印出
2. 所有「質數」數字加總印出

固定參數:

```
String input = "*i!t c$ou7l#d %ch(an)ge b8ut  
this+ feel<s li%^ke like t>h9e ca/lm be??fore  
th[e s=to=rm *not that |*i do''n't w3an/,na  
t~!ry but *i've &b;ee3n here6 be@.fore";
```

輸出結果:

```
It could change but this feels like like the calm before the storm Not that I dont wanna try but Ive been here before  
13
```

設計說明

1. 程式開始時，會先定義一個要解析的字串，並且初始化一些變數。接著，程式會逐一讀取字串中的每個字元。
2. 當讀取到字元時，程式會依照字元的種類進行不同的處理。
3. 如果是星號 '*'，則代表下一個英文字元需要轉成大寫，程式會設定一個變數(upperCase)表示這個狀態。
4. 如果是英文字母或空格，則程式會將它直接加入結果字串中，如果前一個字元是星號，程式會將這個英文字元轉成大寫，並取消星號變數(upperCase)的設定。
5. 如果字元是數字，程式會將這個數字加到一個暫存變數(tempNum)中，代表正在讀取一個數字。如果這個字元不是數字，代表讀完了一個數字，程式會將暫存變數中的數字解析成整數並判斷它是否為質數，如果是則將這個數字加到總和中，最後將暫存變數(tempNum)清空。
6. 當程式讀完整個字串後，會輸出處理後的字串和所有質數的總和。

自訂變數及和函式

函式:

```
● ● ●  
1 public static boolean checkPrime(int num)
```

▲ 檢查傳入的數字是否為質數

```
● ● ●  
1 public static void main(String[] args)
```

▲ 進行操作的主函式

變數:

```
● ● ●  
1 String result = "";
```

▲ 用來儲存結果字串

```
● ● ●  
1 int primeSum = 0;
```

▲ 儲存質數總和

```
● ● ●  
1 boolean findDigit = false;
```

▲ 紀錄是否已查看完數字

```
● ● ●  
1 String tempNum = "";
```

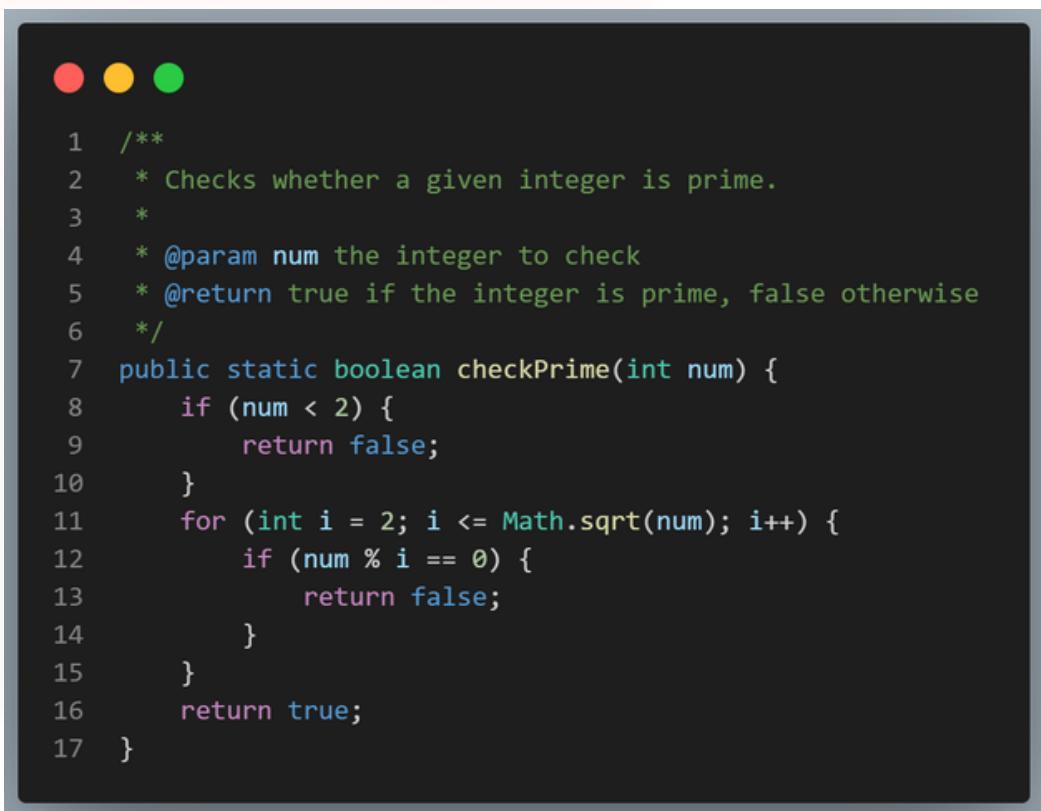
▲ 暫時儲存數字字串

```
● ● ●  
1 boolean upperCase = false;
```

▲ 紀錄是否需要轉換大寫

程式分析

1. 輔助函式 - 判斷質數



The screenshot shows a code editor window with three colored circular icons at the top left (red, yellow, green). The main area contains the following Java code:

```
1  /**
2   * Checks whether a given integer is prime.
3   *
4   * @param num the integer to check
5   * @return true if the integer is prime, false otherwise
6   */
7  public static boolean checkPrime(int num) {
8      if (num < 2) {
9          return false;
10     }
11     for (int i = 2; i <= Math.sqrt(num); i++) {
12         if (num % i == 0) {
13             return false;
14         }
15     }
16     return true;
17 }
```

1. 首先判斷傳入的數是否小於2，如果小於2，直接返回 **false**。
2. 然後使用**for**循環，從2開始到數字平方根之間的所有數字進行遍歷。
3. 在循環中，如果傳入的數能夠被當前遍歷的數整除，就返回 **false**。
4. 如果循環結束後都沒有返回 **false**，就返回 **true**，表示這個數是質數。

2. 主函式

```
● ● ●  
1 // The input string to parse  
2 String input = "*it c$ou7l# %ch(an)ge b8ut this+ feel<s li%^ke like t>h9e ca/lm be??fore th[e s=to=rm "  
3     + "*not that |*i do''n't w3an/,na t~!ry but *i've &b;ee3n here6 be@.fore";  
4  
5 // The result string after parsing  
6 String result = "";  
7  
8 // A temporary string to hold a number being parsed from the input string  
9 String tempNum = "";  
10  
11 // The sum of all prime numbers found in the input string  
12 int primeSum = 0;  
13  
14 // A flag to indicate whether the next alphabetic character should be uppercase  
15 boolean upperCase = false;  
16  
17 // A flag to indicate whether the current character is a number  
18 boolean findDigit = false;
```

▲ 先宣告需要用到的變數

```
● ● ●  
1 // Loop through each character in the input string  
2 for (char c : input.toCharArray())
```

▲ 接下來將字串轉換成CharArray方便遍歷

```
● ● ●  
1 // If the character is "*", set the uppercase flag and clear the digit flag  
2 if (c == '*') {  
3     upperCase = true;  
4     findDigit = false;  
5 }
```

▲ 首先第一點考慮的是遇到 '*' 的情況，這時候下一個字母要大寫，所以將 upperCase 設為 true，而因應到假如 '*' 前出現的是數字的情況，就要將 findDigit 設為 false，表示沒有(連續的數字)繼續接下去針對數字操作的必要。

```
1 // If the character is alphabetic or whitespace
2 if (Character.isAlphabetic(c) || Character.isWhitespace(c)) {
3     // If the uppercase flag is set, append the uppercase version of the character
4     // and clear the uppercase flag
5     if (upperCase) {
6         result += Character.toUpperCase(c);
7         upperCase = false;
8     }
9     // Otherwise, simply append the character
10    else {
11        result += c;
12    }
13    // Clear the digit flag
14    findDigit = false;
15 }
```

▲ 接下來判斷字元是否為 "字母" 或是 "空格"，檢查是否需要轉換為大寫，若是則將該字元轉換為大寫並連接在 result 後方，然後將 upperCase 設為 false 表示操作完成。如果不需要轉換為大寫，則直接將該字元加入 result 後方。同時將 findDigit 設為 false 表示沒有連續的數字出現。

```
1 // If the character is a digit
2 if (Character.isDigit(c)) {
3     // Append the digit to the temporary string and set the digit flag
4     tempNum += c;
5     findDigit = true;
6 }
```

▲ 接下來判斷字元是否為 "數字"，如果遇到數字，將該數字加入 tempNum 中，表示目前已經找到一個數字。同時將 findDigit 設為 true。

```
1 // If the digit flag is not set
2 if (!findDigit) {
3     // If the temporary string contains a number, parse it and check if it's prime,
4     // adding it to the prime sum if it is
5     if (tempNum.length() > 0) {
6         int num = Integer.parseInt(tempNum);
7         if (checkPrime(num)) {
8             primeSum += num;
9         }
10    }
11    // Clear the temporary string
12    tempNum = "";
13 }
```

▲ 最後檢查當前是否有找到數字，如果沒有找到數字，表示目前已經找完一個數字的所有位數，假如 tempNum 內是非空的話就將 tempNum 轉換為整數 num，並檢查 num 是否為質數。若是質數則將 num 加入 primeSum 中。最後將 tempNum 清空。

```
1 // Print the result
2 System.out.println(result);
3 System.out.println(primeSum);
```

▲ 輸出最終字串及質數總和

執行結果

```
PS I:\Java\NCU-MIS-Java\Homework_01> i:; cd 'i:\Java\NCU-MIS-Java\H  
'I:\Java\NCU-MIS-Java\Homework_01\bin' 'HW1_111403538'  
It could change but this feels like like the calm before the storm N  
13
```

```
a\Homework_01'; & 'C:\Program Files\Java\jdk-12.0.1\bin\java.exe' '-cp'  
m Not that I dont wanna try but Ive been here before
```

▲ 輸出結果因過長裁剪為兩份，但輸出與題目要求一致