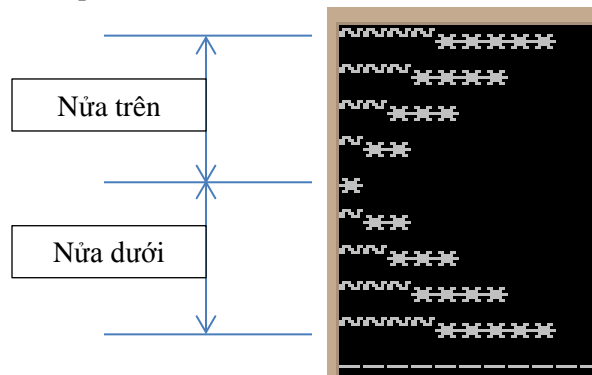


## 1/Đề:

Cho Input  $n = 5$ , in ra hình sau:



## 2/Phân tích ví dụ bài:

-Hình này có tính chất đối xứng (qua dấu “\*”) ở giữa. Ta chia thành 2 nửa: trên và nửa dưới như chú thích trên

-Số hàng của mỗi nửa lần lượt là 4, tương ứng với  $n = 5$  trong ví dụ trên, hay xét tổng quát hơn là  $n-1$

Tổng số hàng tổng cộng là  $2(n-1) + 1 = 2n-1$  hàng

-Luôn đảm bảo hàng của mỗi nửa có 2 patter là “~” và “\*”, trong đó số lượng “~” nhỏ hơn “\*” một đơn vị ở mỗi hàng

### -Xét nửa trên:

+ Số lượng patter ở hàng đầu (tương ứng trái qua phải là pattern “~” và “\*”) là:

$$4 + 5 = n - 1 + n = 2n - 1 = 9, \text{ với } n = 5$$

+ Số lượng patter ở hàng hai (tương ứng trái qua phải là pattern “~” và “\*”) là:

$$3 + 4 = n - 2 + n - 1 = 2n - 3 = 7, \text{ với } n = 5$$

$$\text{hay } 3 + 4 = 2n - 1 \text{ với } n = 4$$

Tương tự cho những hàng còn lại, cho tới hàng thứ  $n-1$  (trước điểm đối xứng vị trí  $n$ ) thì số lượng pattern là 3 ( $= 1+2$ )

### -Xét nửa dưới:

+ Số lượng patter ở hàng đầu (tương ứng trái qua phải là pattern “~” và “\*”) là:

$$1 + 2 = n + n + 1 = 2n + 1 = 3, \text{ với } n = 1$$

+ Số lượng patter ở hàng hai (tương ứng trái qua phải là pattern “~” và “\*”) là:

$$2 + 3 = n + 1 + n + 2 = 2n + 3 = 5, \text{ với } n = 1$$

$$\text{hay } 2 + 3 = 2n + 1 \text{ với } n = 2$$

Tương tự cho những hàng còn lại, cho tới hàng thứ  $2n-1$  (hàng cuối) thì số lượng pattern là  $2n+1$  ( $= n + n+1$ )

Lúc này càng nhận thấy rõ sự đối xứng của 2 nửa

Từ những phân tích trên ta có các ý sau:

+**Ở nửa trên**, số lượng các pattern **giảm dần** theo công thức:  $2n-1$ , với  $n$  bắt đầu từ giá trị đầu vào và giảm dần tới 1.

+**Ở nửa dưới**, số lượng các pattern **tăng dần** theo công thức:  $2n+1$ , với  $n$  bắt đầu từ 1 tới giá trị đầu vào.

Như thế ta có ý tưởng code sau:

### 3/Y tưởng:

-Biến i chạy số lượng hàng :  $1 \leq i \leq 2n - 1$

-Biến j chạy số lượng pattern mỗi hàng

-Điều kiện để biết đang in nửa nào?

$+1 \leq i \leq n - 1$ : nửa trên

$+i = n$ : vị trí đối xứng

$+n + 1 \leq i \leq 2n - 1$ : nửa dưới

-Làm sao xác định được số lượng mỗi pattern “~” và “\*” ở mỗi hàng của mỗi nửa?

**+Nửa trên:**  $1 \leq j \leq 2n - 1, j += 1; n -= 1$  từ giá trị input

- Tổng pattern luôn là:  $2n - 1$
- Số lượng pattern “~” luôn nhỏ hơn “\*” một đơn vị
  - ⇒ Lượng “~” :  $n - 1$ , tương ứng điều kiện  $1 \leq j \leq n - 1$
  - ⇒ Lượng “\*”:  $n$ , tương ứng điều kiện  $n \leq j \leq 2n - 1$ , hay else khi chạy chung với điều kiện chạy “~”

**+Nửa dưới:**  $1 \leq j \leq 2n + 1, n -= 1$  từ giá trị  $n = 1$

- Tổng pattern luôn là:  $2n + 1$
- Số lượng pattern “~” luôn nhỏ hơn “\*” một đơn vị
  - ⇒ Lượng “~” :  $n$ , tương ứng điều kiện  $1 \leq j \leq n$
  - ⇒ Lượng “\*”:  $n + 1$ , tương ứng điều kiện  $n + 1 \leq j \leq 2n + 1$ , hay else khi chạy chung với điều kiện chạy “~”

Đến đây còn một vướng mắc duy nhất, là n luôn thay đổi ở nửa dưới và nửa trên như vậy, thì sau khi chạy như vậy nó sẽ ảnh hưởng đồng thời đến vòng lặp i bên ngoài, và nửa dưới. Như vậy phải làm sao?

-Tạo biến copy = n để giữ giá trị đầu vào của n

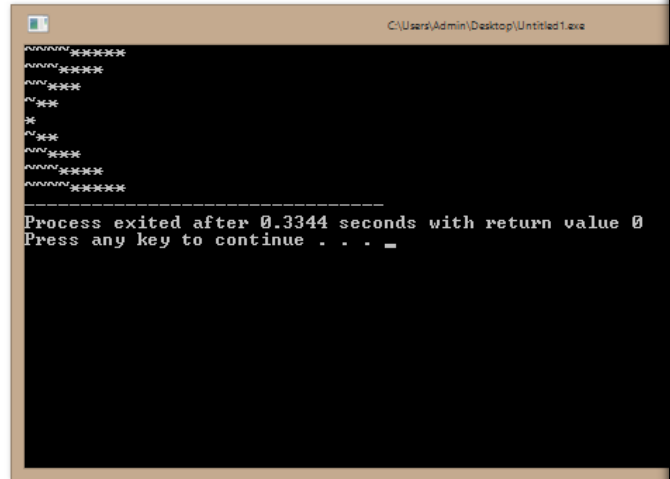
-Thay thế điều kiện của vòng ngoài  $1 \leq i \leq 2n - 1$  thành  $1 \leq i \leq 2 * copy - 1$

-Sau khi chạy xong nửa trên thì lúc này  $n = 2$ , nếu để nguyên mà in nửa dưới, do khác công thức so với nửa dưới và không thể để điều kiện ngoài if và trước else nên ta có cấu trúc tổng quát sau:

```
for(i = 1; i <= 2*copy - 1; i++){
    if(i < n){
        //vẽ nửa trên
    }
    else if(i == n){
        printf("*"); //vẽ vị trí đối xứng
        n = 1; // gán n = 1 để vẽ nửa còn lại
    }
    else{
        //vẽ nửa sau
    }
}
```

#### 4/ Code tham khảo:

```
1  #include <stdio.h>
2
3  void Print_K_Word(int n){
4      int i,j;
5      int copy = n;
6      for(i = 1; i <= 2*copy-1; i++){
7          if(i < copy){
8              for(j = 1; j <= 2*n-1; j++){
9                  if(j <= n-1) printf("~");
10                 else printf("*");
11             }
12             printf("\n");
13             n--;
14         }
15         else if(i == copy){
16             printf("*\n");
17             n = 1;
18         }
19         else{
20             for(j = 1; j <= 2*n+1; j++){
21                 if(j <= n) printf("~");
22                 else printf("*");
23             }
24             if(i < 2*copy-1) printf("\n");
25             n++;
26         }
27     }
28 }
29
30 int main() {
31     Print_K_Word(5);
32     return 0;
33 }
```



```
C:\Users\Admin\Desktop\Untitled1.exe
~~~~~*****
~~~~~*****
~~~~~*****
~~~~~*****
*****
*****
*****
*****
~~~~~*****
~~~~~*****
~~~~~*****

Process exited after 0.3344 seconds with return value 0
Press any key to continue . . .
```