

যাদের প্যাটার্ন প্রিন্টিং প্রোগ্রাম লিখতে সমস্যা হয় তাদের জন্য আমি কীভাবে সমাধান করার চেষ্টা করি সেটা একটু তুলে ধরার চেষ্টা করলাম।

আজকে গ্রুপে ঘুরতে ঘুরতে নিচের দিকে একজনকে দেখলাম এই প্যাটার্নটার ছবি দিয়ে সমাধান চেয়েছেন তাই ডেমো হিসেবে এটাকেই বেছে নিলাম।

Given Pattern:

```

      10
     9 11
    8  12
   7   13
  6    14
 5     15
 4     16
 3     17
 2     18
 1     19
 2     18
 3     17
 4     16
 5     15
 6     14
 7     13
 8     12
 9    11
    10
  
```

Step-1: প্রথমে প্যাটার্নটিকে পর্যবেক্ষণ করে কয়েকটি ছোট ছোট অংশে ভাগ করার জন্য উল্লম্ব / আড়াআড়ি / আনুভূমিক প্রতিসমতা খুঁজতে হবে।

Step-2: এখন ছোট অংশগুলোকে দ্বিমাত্রিক গ্রিডে ফেলে সুবিধামতো ইন্ডেক্সিং করে নিতে হবে।

Step-3: গ্রিড এর ইন্ডেক্স এর সাথে গ্রিড এর উপাদানগুলোর জন্য শর্ত ও সম্পর্ক খুঁজে বের করতে হবে।

Step-4: এখন সেই শর্ত ও সম্পর্ক অনুযায়ী প্রত্যেকটি ছোট ছোট অংশের জন্য কোড ইমপ্লিমেন্ট করতে হবে।

নিচে উপরোক্ত সমস্যাটির জন্য এই ধাপ গুলো ইমপ্লিমেন্ট করে দেখানোর চেষ্টা করলাম

প্রতিসমতা :

```

      10
    part A 9 | 11 part B
      8 | 12
      7 | 13
      6 | 14
      5 | 15
      4 | 16
      3 | 17
      2 | 18
      1 | 19
-----
      2 | 18
      3 | 17
  
```

4		16
5		15
6		14
7		13
part C 8		12 part D
9		11
10		

About used variables, conditions, and assignments to this problem:

-> row (rows of the grid)
-> col (columns of the grid)
-> step(how many step to go) = 10 + 1
-> condition: row + col = step
-> For elements determination: col, n + row - 1

Part A:

	1	2	3	4	5	6	7	8	9	10
1										10
2									9	
3								8		
4							7			
5						6				
6					5					
7				4						
8			3							
9		2								
10	1									

Code:

```

for (row = 1; row <= 10; row++) {
    for (col = 1; col <= n; col++) {
        row + col == step ? printf("%d", col) : printf(" ");
    }
    printf("\n");
}

```

Part B:

9	8	7	6	5	4	3	2	1
---	---	---	---	---	---	---	---	---

[illegible]

```
|--|--|--|--|--|--|--|--|--|
```

Code:

```
for (row = 10 - 1; row > 0; row--) {
    for (col = 1; col <= n; col++) {
        row + col == step ? printf("%d", col) : printf(" ");
    }
    printf("\n");
}
```

Part D:

	9	8	7	6	5	4	3	2	1
9								18	
8							17		
7						16			
6				15					
5			14						
4		13							
3		12							
2	11								
1									

Code:

```
for (row = 10 - 1; row > 0; row--) {
    for (col = 10 - 1; col > 0; col--) {
        row + col == step ? printf("%d", n + row - 1) : printf(" ");
    }
    printf("\n");
}
```

Full Code:

```
#include <stdio.h>

int main(void) {
    int row, col, step, n;

    scanf("%d", &n);

    step = n + 1;
```

```

for (row = 1; row <= n; row++) {
    for (col = 1; col <= n; col++) {
        row + col == step ? printf("%d", col) : printf(" ");
    }

    for (col = n - 1; col > 0; col--) {
        row + col == step ? printf("%d", n + row - 1) : printf(" ");
    }
    printf("\n");
}

for (row = n - 1; row > 0; row--) {
    for (col = 1; col <= n; col++) {
        row + col == step ? printf("%d", col) : printf(" ");
    }

    for (col = n - 1; col > 0; col--) {
        row + col == step ? printf("%d", n + row - 1) : printf(" ");
    }
    printf("\n");
}

return 0;
}

```