

Given two positive integers n and k, print all increasing sequences of length k such that the elements in every sequence are from first n natural numbers.

Examples:

Input: k = 2, n = 3

Output: 1 2
1 3
2 3

Input: k = 5, n = 5

Output: 1 2 3 4 5

Input: k = 3, n = 5

Output: 1 2 3
1 2 4
1 2 5
1 3 4
1 3 5
1 4 5
2 3 4
2 3 5
2 4 5
3 4 5

We strongly recommend to minimize the browser and try this yourself first.

It's a good recursion question. The idea is to create an array of length k. The array stores current sequence. For every position in array, we check the previous element and one by one put all elements greater than the previous element. If there is no previous element (first position), we put all numbers from 1 to n.

Following is C++ implementation of above idea.

```
// C++ program to print all increasing sequences of  
// length 'k' such that the elements in every sequence  
// are from first 'n' natural numbers.
```

```
#include<iostream>  
using namespace std;
```

```
// A utility function to print contents of arr[0..k-1]  
void printArr(int arr[], int k)  
{  
    for (int i=0; i<k; i++)  
        cout << arr[i] << " ";  
    cout << endl;  
}
```

```
// A recursive function to print all increasing sequences  
// of first n natural numbers. Every sequence should be  
// length k. The array arr[] is used to store current  
// sequence.  
void printSeqUtil(int n, int k, int &len, int arr[])  
{  
    // If length of current increasing sequence becomes k,  
    // print it  
    if (len == k)  
    {  
        printArr(arr, k);  
        return;  
    }  
}
```

```

// Decide the starting number to put at current position:
// If length is 0, then there are no previous elements
// in arr[]. So start putting new numbers with 1.
// If length is not 0, then start from value of
// previous element plus 1.
int i = (len == 0)? 1 : arr[len-1] + 1;

// Increase length
len++;

// Put all numbers (which are greater than the previous
// element) at new position.
while (i<=n)
{
    arr[len-1] = i;
    printSeqUtil(n, k, len, arr);
    i++;
}

// This is important. The variable 'len' is shared among
// all function calls in recursion tree. Its value must be
// brought back before next iteration of while loop
len--;
}

// This function prints all increasing sequences of
// first n natural numbers. The length of every sequence
// must be k. This function mainly uses printSeqUtil()
void printSeq(int n, int k)
{
    int arr[k]; // An array to store individual sequences
    int len = 0; // Initial length of current sequence
    printSeqUtil(n, k, len, arr);
}

// Driver program to test above functions
int main()
{
    int k = 3, n = 7;
    printSeq(n, k);
    return 0;
}

```

Output:

```

1 2 3
1 2 4
1 2 5
1 2 6
1 2 7
1 3 4
1 3 5
1 3 6
1 3 7
1 4 5
1 4 6
1 4 7
1 5 6
1 5 7
1 6 7
2 3 4
2 3 5

```

```
2 3 6
2 3 7
2 4 5
2 4 6
2 4 7
2 5 6
2 5 7
2 6 7
3 4 5
3 4 6
3 4 7
3 5 6
3 5 7
3 6 7
4 5 6
4 5 7
4 6 7
5 6 7
```