

```

#include <stdio.h>

#include <stdbool.h>

#define MAX_N 10

int p[MAX_N];

int pi[MAX_N];

int n;

void initPermutation() {
    for (int i = 0; i < n; i++) {
        p[i] = i + 1;
        pi[i] = -1;    }
}

void printPermutation() {
    for (int i = 0; i < n; i++) {
        printf("%d ", p[i]);    }
    printf("\n");    }

int getLargestMobile() {
    int mobile = 0;

    for (int i = 0; i < n; i++) {
        if (pi[p[i] - 1] == -1 && i > 0 && p[i] > p[i - 1]) {
            if (p[i] > p[i - 1] && p[i] > mobile) {
                mobile = p[i];    }
        }

        if (pi[p[i] - 1] == 1 && i < n - 1 && p[i] > p[i + 1]) {
            if (p[i] > p[i + 1] && p[i] > mobile) {
                mobile = p[i];    }
        }
    }    return mobile;    }

int getLargestMobilePos(int mobile) {
    for (int i = 0; i < n; i++) {
        if (p[i] == mobile) {

```

```

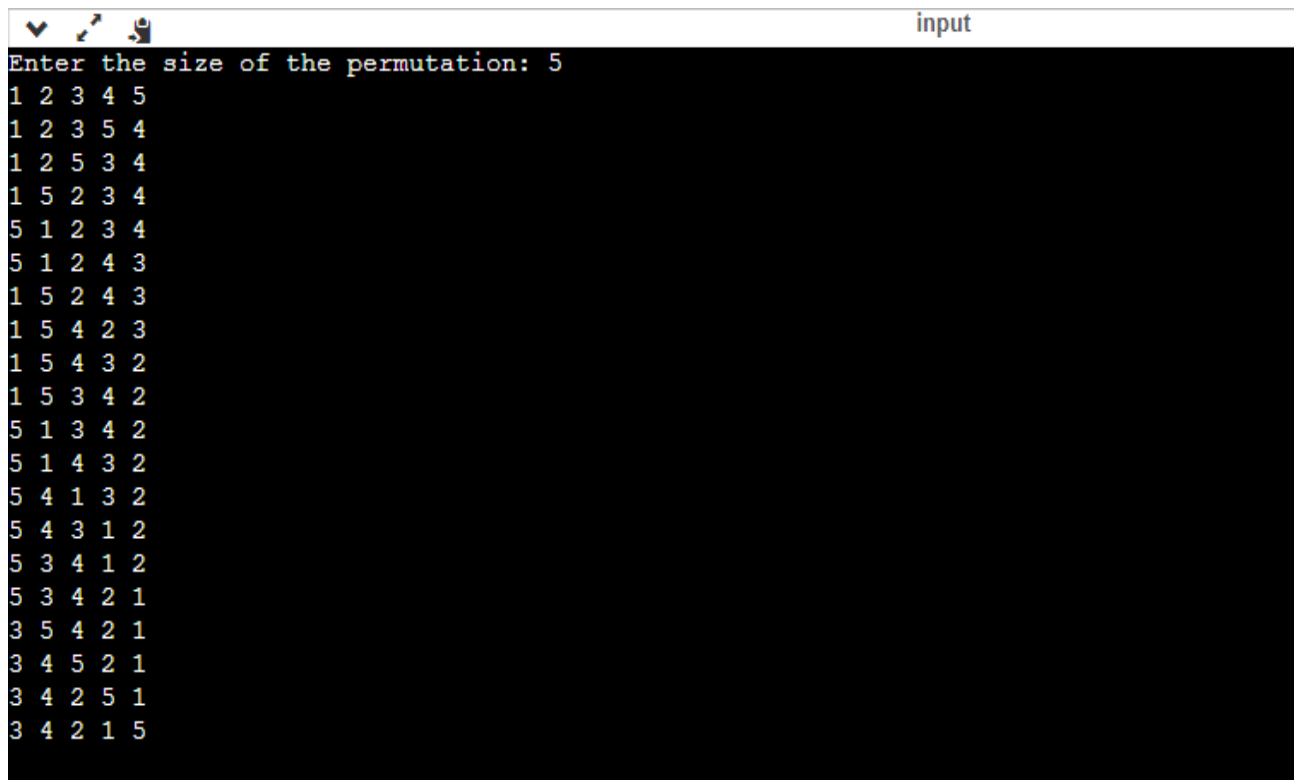
        return i;    } }
    return -1; }

bool performStep() {
    int mobile = getLargestMobile();
    if (mobile == 0) {
        return false;    }
    int pos = getLargestMobilePos(mobile);
    int dir = pi[mobile - 1];
    if (dir == -1) {
        int temp = p[pos - 1];
        p[pos - 1] = p[pos];
        p[pos] = temp;
        int temp2 = pi[mobile - 1];
        pi[mobile - 1] = pi[mobile - 2];
        pi[mobile - 2] = temp2;
    } else if (dir == 1) {
        int temp = p[pos + 1];
        p[pos + 1] = p[pos];
        p[pos] = temp;
        int temp2 = pi[mobile - 1];
        pi[mobile - 1] = pi[mobile];
        pi[mobile] = temp2;    }
    for (int i = 0; i < n; i++) {
        if (p[i] > mobile) {
            pi[p[i] - 1] *= -1;    } }
    return true; }

void generatePermutations() {
    initPermutation();
    printPermutation();
}

```

```
while (performStep()) {  
    printPermutation(); } }  
  
int main() {  
    printf("Enter the size of the permutation: ");  
    scanf("%d", &n);  
    if (n > MAX_N || n < 1) {  
        printf("Invalid input!\n");  
        return 0;    }  
    generatePermutations();  
    return 0;  
}
```



The screenshot shows a terminal window with a title bar that includes a window icon, a maximize icon, and a close icon, followed by the text "input". The terminal content displays the prompt "Enter the size of the permutation: 5" followed by a list of 20 permutations of the numbers 1 through 5, each on a new line. The permutations are: 1 2 3 4 5, 1 2 3 5 4, 1 2 5 3 4, 1 5 2 3 4, 5 1 2 3 4, 5 1 2 4 3, 1 5 2 4 3, 1 5 4 2 3, 1 5 4 3 2, 1 5 3 4 2, 5 1 3 4 2, 5 1 4 3 2, 5 4 1 3 2, 5 4 3 1 2, 5 3 4 1 2, 5 3 4 2 1, 3 5 4 2 1, 3 4 5 2 1, 3 4 2 5 1, and 3 4 2 1 5.

```
input  
Enter the size of the permutation: 5  
1 2 3 4 5  
1 2 3 5 4  
1 2 5 3 4  
1 5 2 3 4  
5 1 2 3 4  
5 1 2 4 3  
1 5 2 4 3  
1 5 4 2 3  
1 5 4 3 2  
1 5 3 4 2  
5 1 3 4 2  
5 1 4 3 2  
5 4 1 3 2  
5 4 3 1 2  
5 3 4 1 2  
5 3 4 2 1  
3 5 4 2 1  
3 4 5 2 1  
3 4 2 5 1  
3 4 2 1 5
```