

WEEK 4

ADALAB: Johnson-Trotter Algorithm for Permutations

VIBHA HUGAR 1BM21CS255

CODE

```
#include <stdio.h>

#include <stdbool.h>

#define RIGHT_TO_LEFT false
#define LEFT_TO_RIGHT true

int searchArr(int a[], int n, int mobile) {
    for (int i = 0; i < n; i++)
        if (a[i] == mobile)
            return i + 1;
    return -1;
}

int getMobile(int a[], bool dir[], int n) {
    int mobile_prev = 0, mobile = 0;
    for (int i = 0; i < n; i++) {
        if (dir[a[i] - 1] == RIGHT_TO_LEFT && i != 0) {
            if (a[i] > a[i - 1] && a[i] > mobile_prev) {
                mobile = a[i];
                mobile_prev = mobile;
            }
        }
    }

    if (dir[a[i] - 1] == LEFT_TO_RIGHT && i != n - 1) {
```

```

        if (a[i] > a[i + 1] && a[i] > mobile_prev) {
            mobile = a[i];
            mobile_prev = mobile;
        }
    }
}

```

```

if (mobile == 0 && mobile_prev == 0)
    return 0;
else
    return mobile;
}

```

```

void swap(int *x, int *y) {
    int temp = *x;
    *x = *y;
    *y = temp;
}

```

```

void printOnePerm(int a[], bool dir[], int n) {
    int mobile = getMobile(a, dir, n);
    int pos = searchArr(a, n, mobile);

    if (dir[a[pos - 1] - 1] == RIGHT_TO_LEFT)
        swap(&a[pos - 1], &a[pos - 2]);
    else if (dir[a[pos - 1] - 1] == LEFT_TO_RIGHT)
        swap(&a[pos], &a[pos - 1]);

    for (int i = 0; i < n; i++) {

```

```
    if (a[i] > mobile) {  
        if (dir[a[i] - 1] == LEFT_TO_RIGHT)  
            dir[a[i] - 1] = RIGHT_TO_LEFT;  
        else if (dir[a[i] - 1] == RIGHT_TO_LEFT)  
            dir[a[i] - 1] = LEFT_TO_RIGHT;  
    }  
}
```

```
for (int i = 0; i < n; i++)  
    printf("%d", a[i]);  
printf(" ");  
}
```

```
int fact(int n) {  
    int res = 1;  
    for (int i = 1; i <= n; i++)  
        res = res * i;  
    return res;  
}
```

```
void printPermutation(int n) {  
    int a[n];  
    bool dir[n];  
  
    for (int i = 0; i < n; i++) {  
        a[i] = i + 1;  
        printf("%d", a[i]);  
    }  
    printf("\n");  
}
```

```

for (int i = 0; i < n; i++)

    dir[i] = RIGHT_TO_LEFT;

for (int i = 1; i < fact(n); i++)

    printOnePerm(a, dir, n);
}

int main() {

    int n;

    printf("Enter the value of n: ");

    scanf("%d", &n);

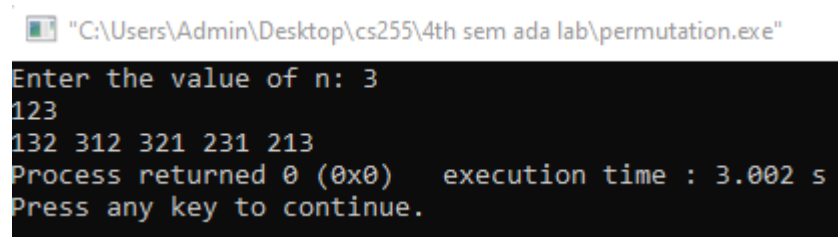
    printPermutation(n);

    return 0;

}

```

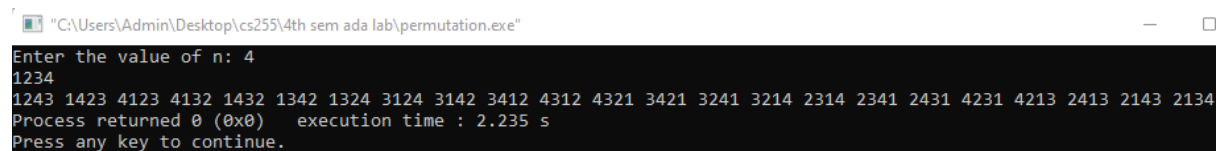
OUTPUT



```

"C:\Users\Admin\Desktop\cs255\4th sem ada lab\permutation.exe"
Enter the value of n: 3
123
132 312 321 231 213
Process returned 0 (0x0)   execution time : 3.002 s
Press any key to continue.

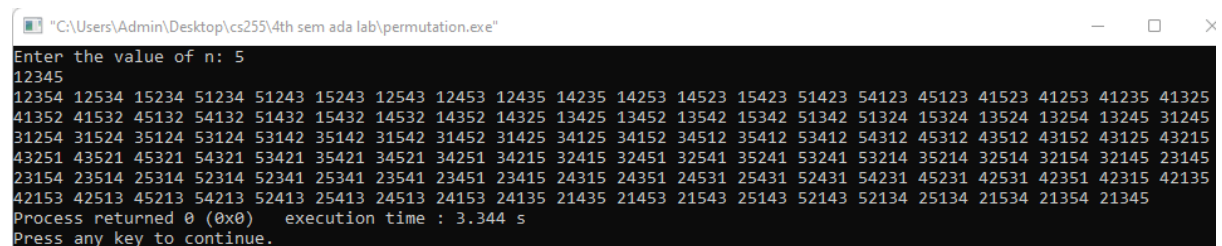
```



```

"C:\Users\Admin\Desktop\cs255\4th sem ada lab\permutation.exe"
Enter the value of n: 4
1234
1243 1423 4123 4132 1432 1342 1324 3124 3142 3412 4312 4321 3421 3241 3214 2314 2341 2431 4231 4213 2413 2143 2134
Process returned 0 (0x0)   execution time : 2.235 s
Press any key to continue.

```



```

"C:\Users\Admin\Desktop\cs255\4th sem ada lab\permutation.exe"
Enter the value of n: 5
12345
12354 12534 15234 51234 51243 15243 12543 12453 12435 14235 14253 14523 15423 51423 54123 45123 41523 41253 41235 41325
41352 41532 45132 54132 51432 15432 14532 14352 14325 13425 13452 13542 15342 51342 51324 15324 13524 13254 13245 31245
31254 31524 35124 53124 53142 35142 31542 31452 31425 34125 34152 34512 35412 53412 54312 45312 43512 43152 43125 43215
43251 43521 45321 54321 53421 35421 34521 34251 34215 32415 32451 32541 35241 53241 53214 35214 32514 32154 32145 23145
23154 23514 25314 52314 52341 25341 23541 23451 23415 24315 24351 24531 25431 52431 54231 45231 42531 42351 42315 42135
42153 42513 45213 54213 52413 25413 24513 24153 24135 21435 21453 21543 25143 52143 52134 25134 21534 21354 21345
Process returned 0 (0x0)   execution time : 3.344 s
Press any key to continue.

```

[illegible]