NAME: VALLISHA.M

USN : 1BM19CS177

Course : Analysis and Design of Algorithms (Lab - 1)

Date : 14-06-2021

Course Code : 19CS4PCADA

Johnson Trotter Algorithm

```
#include <stdio.h>

int right = 1;
int left = 0;

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

int   getMobileElement(int a[], int dir[], int n){
      int mobile-prev=0, mobile=0;
      for (int i=0; i<n;i++)
      {        if (dir[a[i]-1] == left &&i!=0)
               {   if ( a[i] > a[i-1] && a[i] > mobile_prev   )
                   {     mobile= a[i];
                         mobile-prev = mobile;

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

Vallisha.M

```
    if (a[i] > a[i+1] && a[i] > mobile - perere)
    {
          mobile = a[i];
          Mobile - prere = mobile;
        }
      }
  }

  if (mobile == 0 && mobile - perere == 0)
          return 0;
  else
        return mobile;
}
void swap (int* a, int* b){
      int t = *a;
       *a = *b;
        *b = t;
}
int printOnePermutation (int a[], int dir[], int n)
{   int mobile = getMobileElement (a, dir, n);
    int pos = search (a, n, mobile);
                                            left
    if (dir [a[pos-1] -1] == ~~right~~)
          swap ( &a [pos-1], & a [pos -2]);
    else if (dir [a[pos-1] -1] == right)
          swap (&a [pos], &a [pos -1]);
```

Vallisha.M
1BM19CS177

```
for (int i=0; i<n; i++)
  {
        if (a[i] >mobile)
        {       if (dir[a[i]-1] ==right)
                        dir[a[i]-1] = left;

                else if (dir[a[i]-1] == left)
        }                       dir [a[i]-1] = right;
  }

  for (int i=0; i<n; i++)
              printf("%d", a[i]); //modification
    printf("\n");
}
 int fact (int n) {
    int res =1;
   for (int i=1; i<=n; i=i+1)
          res =res*i;

   return res;

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

```
for (int i=0; i<n; i++)
        dir [i] = left;
  for (int i=1; i<fact(n) ;i++)
            printOnePermutation (a, dir, n);
}
int  main ()
{
       int n;
       printf ("Enter n : ");
         scanf("%d", &n);
          printf (" The permutations are ; \n");
          onebyonePermutation (n);
          return 0;
}
```

For modification to find permutation of ABCD, in the print statements marked as modification, change the statement to;

```
printf ("%c", (a[i]+64));
```

(There are two such statements)

Then for "ABCD", just input 4 ~~in the main program~~ when asked to enter n.