```
Name: Ravi. Sajjanar Date 10/6/2021
  Course: ADA Lab CIB-1 Sign (RAYS).
Code: 19CSYPCADA
  USN: 1BM19 C8127
 sem/sec: 4-c
   Batch : 2nd home (1-17)
                  illim dielara
Q7) Implement Johnson Trotter algorithm to generate
       permutation:
# include < math. b>
    int left-to-right=1; (11/10 ((1/10))),
    void swap ( ab int * a , int * 4) {
         int temp = * x;
() *x = *y;
          * y = temp; 110 = 11dom) /
        } i didore nosta? ? sels
    int search char (int al], int mobile, int n) s
for (i20; ic n; i+) (2) arrighting to
      if (aci) = = mobile) lidson In
(11 Judam, 1) return 177; 2 209 to
+ jet - st- tolpgo = = [1-[1 209] n] silo) fi
      : ( } = 201/D/, [1-01/D9) 1018
```

```
int get mobile (int al], int n, int direly) sint
      int mobile = 0;
         for (int i=0; icn; i++) {
        if (dir[a[i]-1] == right_to-left $4 1/20) $
           if (aci)> ari-7] 44 ari] > mobile-prev) s
mobile = ali];
mobile prev = mobile jumpique (1)
        if (dir [a[i]-1]== left-to_right & & i!=n-1) }
            if (a(i)> a(i+1) 44 a(i)> mobile-prev) q
               mobile=a[i];
mobile=a[i];
mobile=prev=mobile;

(bay all we have been
        if (motorib===0 64 motorib=prev==0) { setum 0; }
        else { return mobile; }
flatai phome hai, l'In hai y and donne mi
      int printon prem (int al], int dis [], int n) &
          int mobile = getmobile (a,n, dir);
            int pos = search char (a, mobile, n);
             if (dix [a[po8-1]-1] = = right-to-left)
           swap (pa[pos-1], & a[pos-2]);
```

```
else & swap (pa [pos-1], pa [pos]);
 For (int 120; icn; i++) { [ 11]
    if (atij > motoile) ? [i] i)
       if cdir [a[i]-1]= orght-to-left] ?
          dir [a[i]-1]= left-to-right;
       else & dir [ali]-1] = right to-left;
            }(++1; (0) i; (0) 1 (ai) 0)
 for (int i = 0; kn; i++) & hing
      & printf ("1.d\t", a(i)); &
                         ("a/") / lang
    pmintf (" /n");
  Int fact (int n) &
      for (intial; ik=n; i++) {
             P=p*ij
         return p;
             11155 11115
(Clithe & bit) Hard
```

```
for (int i=0; i < n; i++) & my / my ) grant 10)
for
                if (a[i]>mobile) &
              if [dir[a[i]-1]== right -to-left) ?
                  dix [a [i]-1] = left to -right; 11
            flips of helper = [1-[1] orbor 11
           else 3 / 1/1 / 1 - 61.70/ 1/1
               dir [a [i]-i]= right-to-left j
             illal - A - liter : El Elloy sib
           for (int izo; icn; i+r) }
                    print f("j-d';a[i]); 1 kai) oil
                  3 (1170 ("1/b 1") / tried
           printf (" \n");
           2
              fact (int n) 9.
                   int p=1; } (a lai) no) in1
                    for (int i=1; i <=n; i++) {
                     octum pi
         void per (int n) {
                     int a[n];
                     int dir [n];
                     for Cint izo; icn; i++) }
                           ali] = i + 1;
                             printf ("1-d" a , a [i]);
                       3
```

mint & (""); for (int i=0; icn; i++) & dir [i] = right to left; for (int i=0; ic(fact(m)-1); i++) { printon prem (a, dir, n); int main() & int nj Printf ("Enter the no of terms \n"); scanf (".f.d", 4n); per(n);

```
Pelsophian
 Modification:
 Generate permutation for ABCD
 # include < Stdio.h>
# include < math.h>
  int left to right = 1;
   int right - to- left = 0;
   void swap (char * x, char * y)
   & char temp = * x;
  * 4 = +emp;
   int searchcharr (char al), int mobile, int n)?
    The continue of the grant
   if (a [i] == mobile)

return it!
   int get mobile (chas all, int n, int dir []) q
  3 (Mal 3)
     char mobile = 11,
    char mobile prev= ";
    for (int i=0; i<n; i++) \q
         if (dis [a[i]-1]= right_to_left $$ i!=0)
            ? if (aci)>aci-1] & & aci)>mobile-prev)
         ( M & mobile = a [i];
                  mobile-prev = mobile;
```

```
int fact (int m) {
  int b=t.
  for (int i=1; i = n; i++) {

P=p x i;
 3 return b?
 void per (int n) ?
        chas a [n];
    int dis [n];
    int K= 'A';
   for (int i=0; ixn; i++, K++) {
  ali]= K;
  printf("1.clt", a[i]);
     Printf (" /0");
       for (int i=0; Kn; i++) {
          dir(i) = right-to-left;
      for (int i=0; ic (fact (n)-1); i+1) }
             printonprem (a, dir, n);
    int main () §
          pa (4);
```