LAB 7- Johnson Trotter algorithm to generate permutations.

Anitej Prasad

1BM19CS194

4-D

```
#include<stdio.h>
int left_to_right=1;
int right_to_left=0;

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

int getmobile(int a[], int dir[],int n)
{
	int mob_prev=0,mob=0,i;
```

```
        for(i=0;i<n;i++)
        {
                if(dir[a[i]-1]==right_to_left && i!=0)
                {
                        if(a[i]>a[i-1]&&a[i]>mob_prev)
                        {
                                mob=a[i];
                                mob_prev=mob;
                        }
                }
                if(dir[a[i]-1]==left_to_right && i!=n-1)
                {
                        if(a[i]>a[i+1]&&a[i]>mob_prev)
                        {
                                mob=a[i];
                                mob_prev=mob;
                        }
                }
        }
        if(mob==0&&mob_prev==0)
        return 0;
        else
        return mob;
}

int printoneperm(int a[], int dir[],int n)
```

```
{
        int mob,pos,temp,i;
        mob=getmobile(a,dir,n);
        pos=searcharr(a,n,mob);
        if(dir[a[pos-1]-1]==right_to_left)
        {
                temp=a[pos-1];
                a[pos-1]=a[pos-2];
                a[pos-2]=temp;
        }
        else if(dir[a[pos-1]-1]==left_to_right)
        {
                temp=a[pos];
                a[pos]=a[pos-1];
                a[pos-1]=temp;
        }
        for(i=0;i<n;i++)
        {
                if(a[i]>mob)
                {
                        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(i=0;i<n;i++)
        {
                printf("%d",a[i]);
        }
        printf("   ");
        return 0;
}



void printpermutaions(int n)
{
        int a[n],i;
        int dir[n];
        for(i=0;i<n;i++)
        {
                a[i]=i+1;
                printf("%d",a[i]);
        }
        printf("  ");
        for(i=0;i<n;i++)
        dir[i]=right_to_left;
        while(getmobile(a,dir,n)!=0)
        printoneperm(a,dir,n);
}
```

```
void main()
{
        int n;
        printf("Enter the number\n");
        scanf("%d",&n);
   printf("The Permutations are : \n");
        printpermutaions(n);
        printf("\n");
}
```