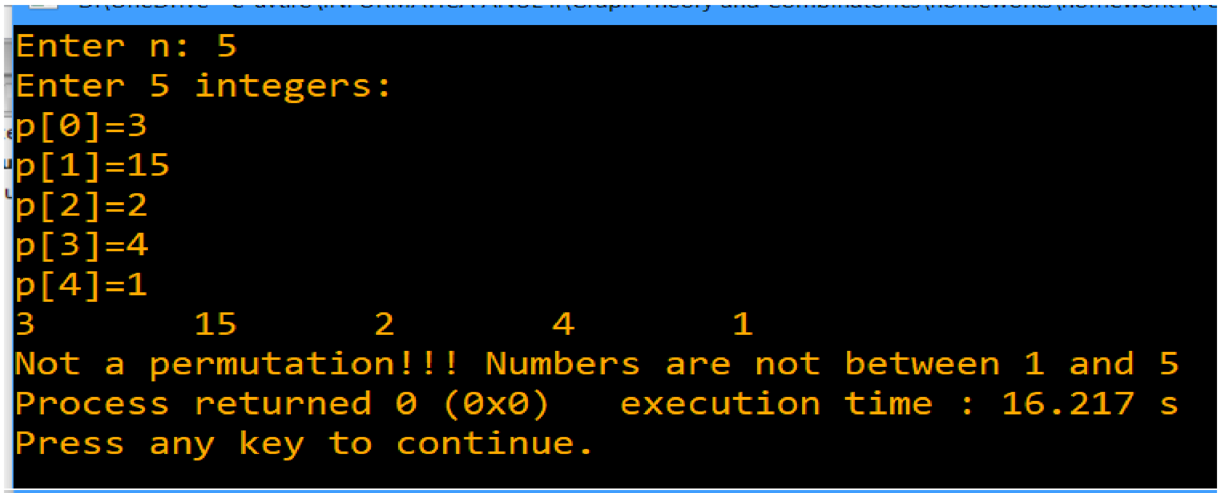
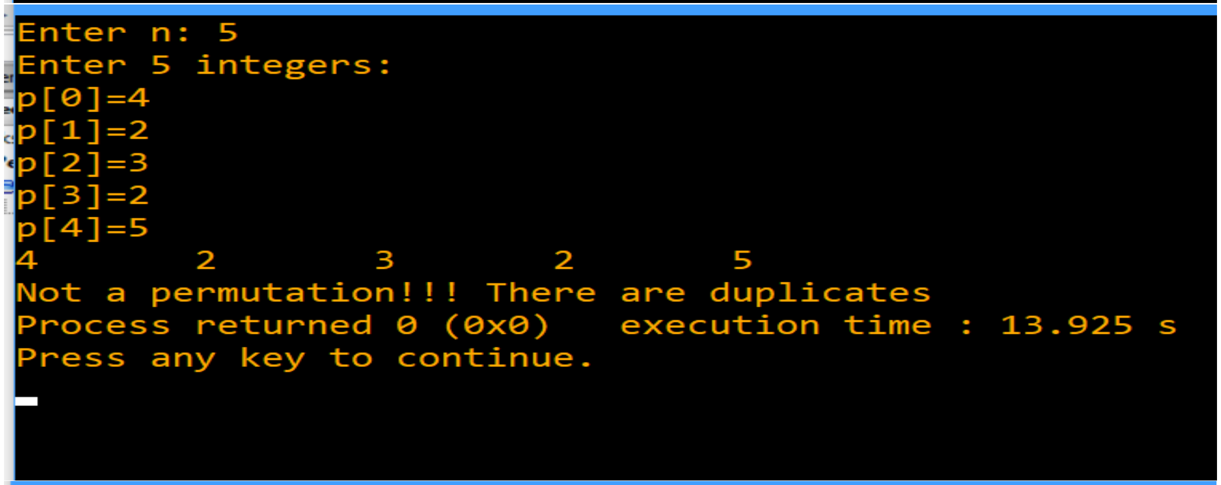
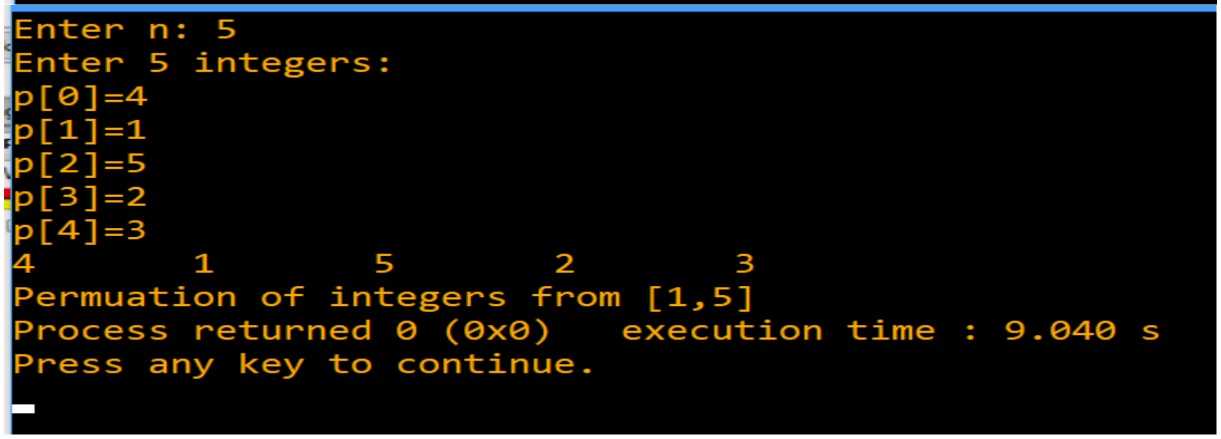
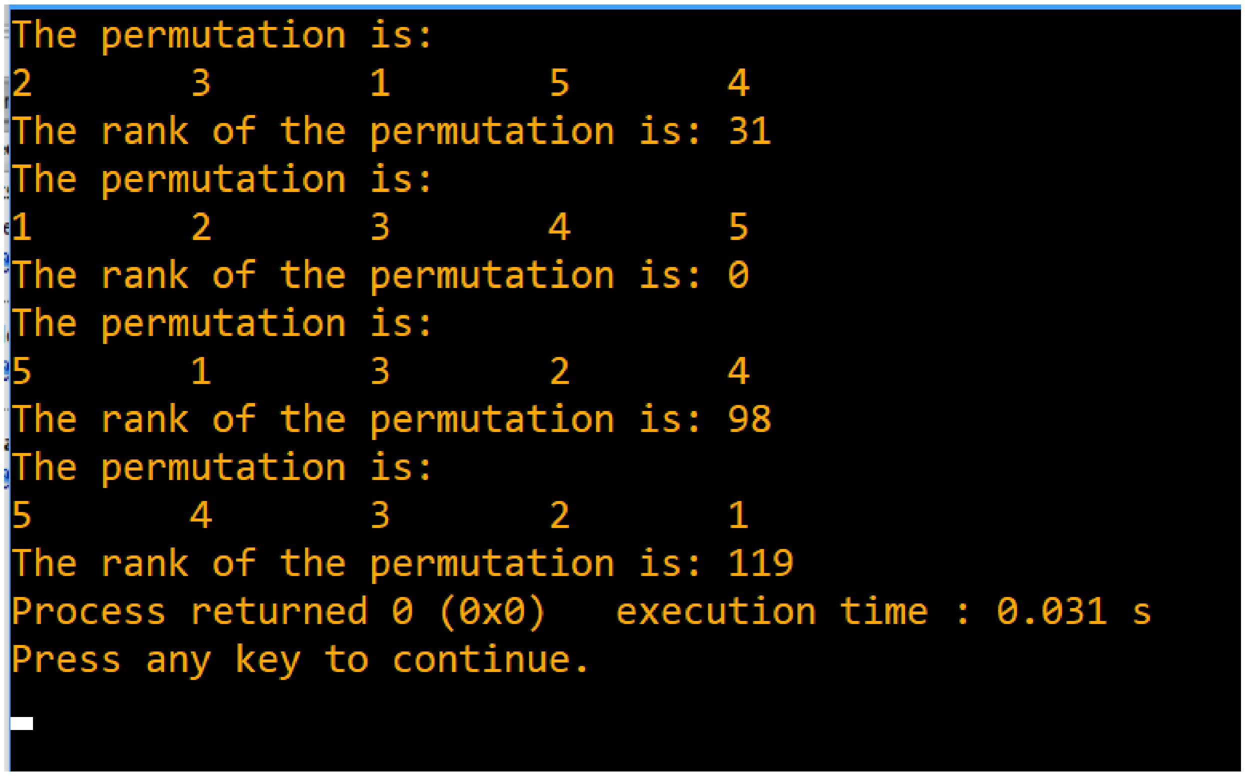
```
/*
PROBLEM 1 / LECTURE 2
Write a program which reads a sequence of n numbers, and
then it displays:
"a permutation" if the sequence is a permutation of {1..n}
"not a permutation" otherwise.
*/
#include <stdio.h>
#include <stdlib.h>
#define N 100

int main()
{
   int n, p[N], i, j;

   printf("Enter n: ");scanf("%d", &n);
   printf("Enter %d integers:\n", n);
   for(i=0; i<n; i++){
      printf("p[%d]=",i);
      scanf("%d", &p[i]);
   }

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

   for(i=0; i<n ;i++){
      if(p[i]<1 || p[i]> n){
         printf("\nNot a permutation!!! Numbers are not between 1 and %d", n);
         exit(0);
      }
   }

   for(i=0; i<n-1; i++){
      for(j=i+1; j<n; j++){
         if(p[i]==p[j]){
            printf("\nNot a permutation!!! There are duplicates");
            exit(0);
         }
      }
   }
   printf("\nPermuation of integers from [1,%d]", n);
   return 0;
}
```


```
Enter n: 5
Enter 5 integers:
p[0]=3
p[1]=15
p[2]=2
p[3]=4
p[4]=1
3       15      2       4       1
Not a permutation!!! Numbers are not between 1 and 5
Process returned 0 (0x0)   execution time : 16.217 s
Press any key to continue.
```

```
Enter n: 5
Enter 5 integers:
p[0]=4
p[1]=2
p[2]=3
p[3]=2
p[4]=5
4       2       3       2       5
Not a permutation!!! There are duplicates
Process returned 0 (0x0)   execution time : 13.925 s
Press any key to continue.
```

```
Enter n: 5
Enter 5 integers:
p[0]=4
p[1]=1
p[2]=5
p[3]=2
p[4]=3
4       1       5       2       3
Permuation of integers from [1,5]
Process returned 0 (0x0)   execution time : 9.040 s
Press any key to continue.
```

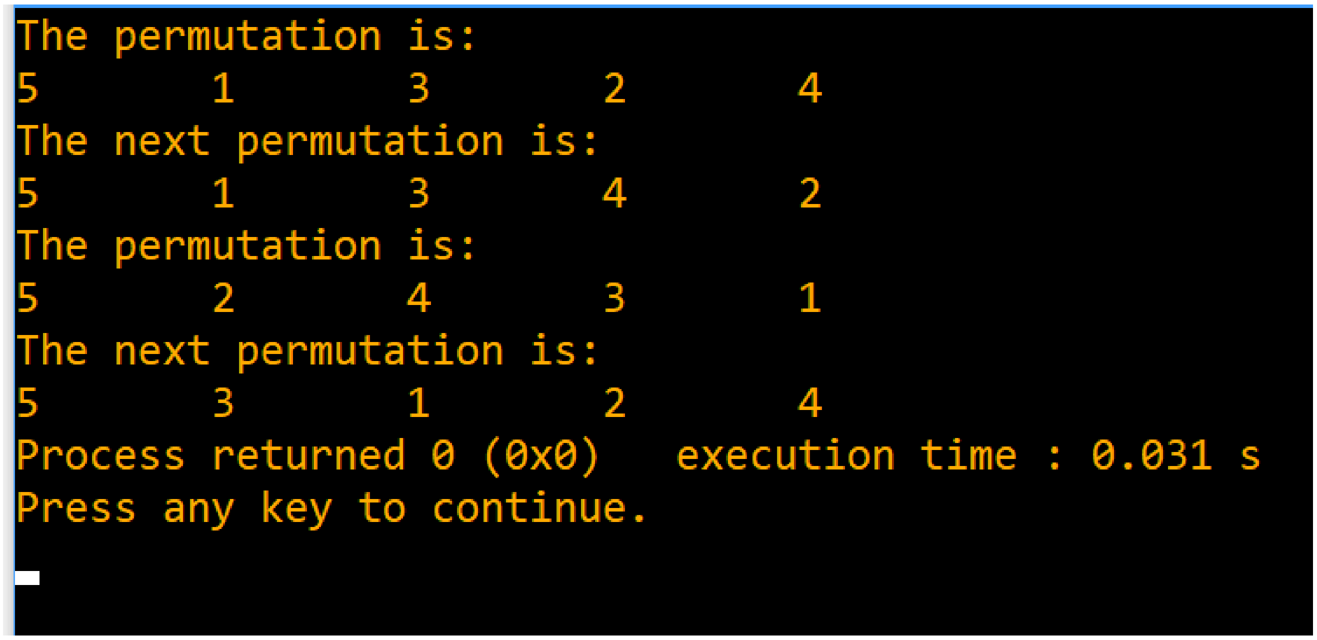
```
/*
Determining the RANK of a given permutation.
Implementation of pseudocode from SLIDE 13 / LECTURE 2
*/

#include <stdio.h>
#include <stdlib.h>

int factor(int n){
   if(n==1 || n==0)
      return 1;
   else
      return factor(n-1)*n;
}
int rankOfPermutation(int p[], int n){
   int q[n-1];
   int i;
   if(n==1)
      return 0;
   else{
      // adjust p[1..n-1] to become a permutation of {1,...,n-1}
      // memorized in the array q[0 .. n-2]
      for(i=1; i< n-1; i++)
         if(p[i]<p[0])
            q[i-1] = p[i];
         else
            q[i-1] = p[i]-1;
   }
   return rankOfPermutation(q, n-1) + (p[0]-1)*factor(n-1);
}
int main()
{
   int k;
   int p1[] = {2, 3, 1, 5, 4};
   int p2[] = {1, 2, 3, 4, 5};
   int p3[] = {5, 1, 3, 2, 4};
   int p4[] = {5, 4, 3, 2, 1};
   printf("The permutation is:\n");
   for(k=0; k<sizeof(p1)/sizeof(p1[0]); k++)
      printf("%d\t", p1[k]);
   printf("\nThe rank of the permutation is: %d\t", rankOfPermutation(p1,sizeof(p1)/sizeof(p1[0])));
   printf("\nThe permutation is:\n");
   for(k=0; k<sizeof(p2)/sizeof(p2[0]); k++)
      printf("%d\t", p2[k]);
   printf("\nThe rank of the permutation is: %d\t", rankOfPermutation(p2,sizeof(p2)/sizeof(p2[0])));
   printf("\nThe permutation is:\n");
   for(k=0; k<sizeof(p3)/sizeof(p3[0]); k++)
      printf("%d\t", p3[k]);
```

```
    printf("\nThe rank of the permutation is: %d\t", rankOfPermutation(p3,sizeof(p3)/sizeof(p3[0])));
    printf("\nThe permutation is:\n");
    for(k=0; k<sizeof(p4)/sizeof(p4[0]); k++)
        printf("%d\t", p4[k]);
    printf("\nThe rank of the permutation is: %d\t", rankOfPermutation(p4,sizeof(p4)/sizeof(p4[0])));
    return 0;
}
```



```
The permutation is:
2       3       1       5       4
The rank of the permutation is: 31
The permutation is:
1       2       3       4       5
The rank of the permutation is: 0
The permutation is:
5       1       3       2       4
The rank of the permutation is: 98
The permutation is:
5       4       3       2       1
The rank of the permutation is: 119
Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.
```

```
/*
Determining the next permutation of a given permutation.
Implementation of pseudocode from SLIDE 9 / LECTURE 2
*/
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#define N 100

void nextPermutation(int *p, int n){
    int i, j, temp, k, c=0;
    i = n-2;
    while(p[i]>p[i+1]){
        i--;
        c++;
    }
    if(c==n-1){
        printf("\nNO NEXT PERMUTATION!!!");
    }
    else{
    j = n-1;
    while(p[j]<p[i])
        j--;

    // swap p[i] with p[j]
    temp = p[i];
    p[i] = p[j];
    p[j] = temp;

    // revert (p[i+1],...,p[n-1])

    for(k=0; k<(n-i-1)/2; k++){
        // swap p[i+1+k] with p[n-1-k]
        temp = p[i+1+k];
        p[i+1+k] = p[n-1-k];
        p[n-1-k] = temp;
    }
    printf("\nThe next permutation is:\n");
    for(k=0; k<n; k++)
        printf("%d\t", *(p+k));
    }
}
int main()
{
    int k, n;
    char c=' ';
    int p[N];
    //int p[] = {5,4,3,2,1};//{5, 1, 3, 2, 4}; // example from lecture 2
    //int p2[] = {5, 2, 4, 3, 1}; // example from lecture 2
    do{
        printf("Enter n: ");
        scanf("%d", &n);

        printf("Enter the elements of the permutation: ");
        for(k=0; k<n; k++)
            scanf("%d", &p[k]);

        printf("The permutation is:\n");
        for(k=0; k<n; k++)
            printf("%d\t", p[k]);

        nextPermutation(p, n);
        printf("\nIf you want to enter another exemple type y:");
        fflush(stdin);
        scanf("%c", &c);
    }while(c=='y');

    return 0;
}
```

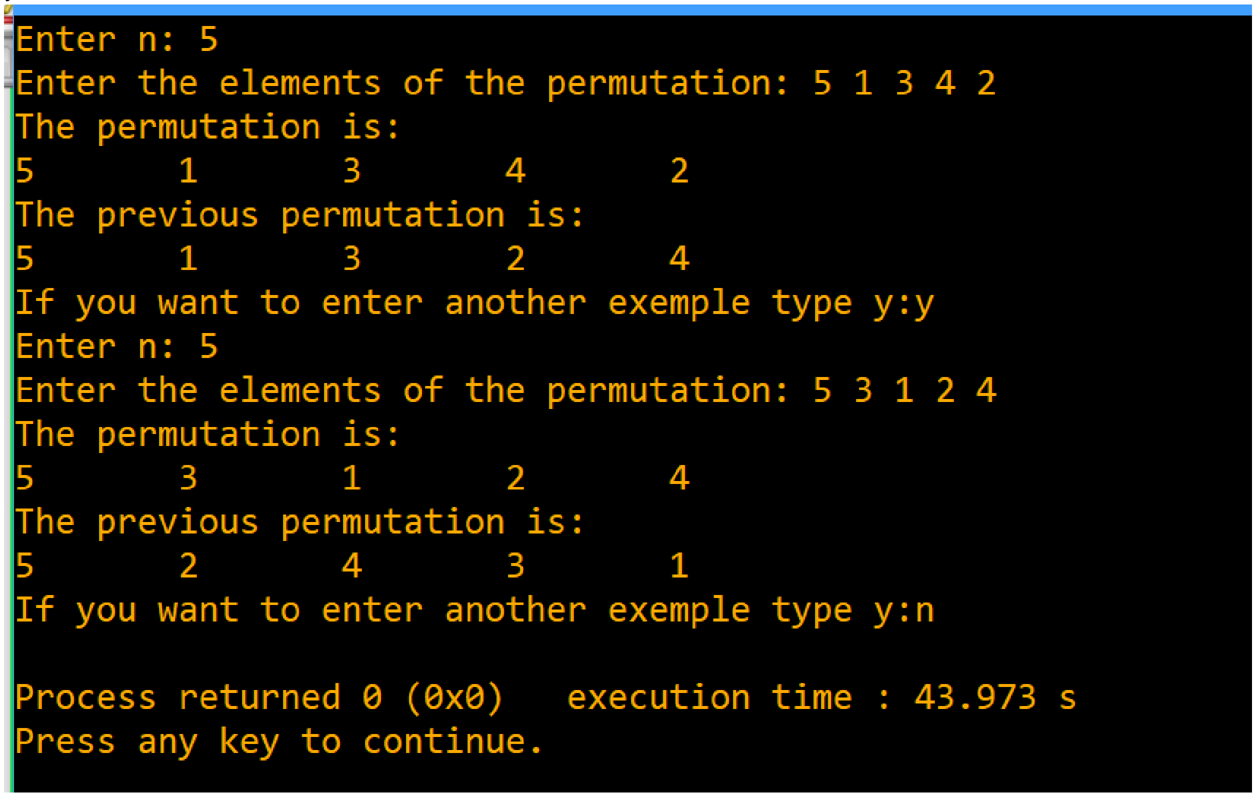


```
The permutation is:
5       1       3       2       4
The next permutation is:
5       1       3       4       2
The permutation is:
5       2       4       3       1
The next permutation is:
5       3       1       2       4
Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.
```

```
/*
Determining the previous permutation of a given permutation.
Implementation of pseudocode from SLIDE 9 / LECTURE 2
*/
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#define N 100

void previousPermutation(int *p, int n){
    int i, j, temp, k, c=0;
    i = n-2;
    while(p[i]<p[i+1]){
        i--;
        c++;
    }
    if(c==n-1){
        printf("\nNO PREVIOUS PERMUTATION!!!");
    }
    else{
    j = n-1;
    while(p[j]>p[i])
```

```
            j--;

        // swap p[i] with p[j]
        temp = p[i];
        p[i] = p[j];
        p[j] = temp;

        // revert (p[i+1],...,p[n-1])

        for(k=0; k<(n-i-1)/2; k++){
            // swap p[i+1+k] with p[n-1-k]
            temp = p[i+1+k];
            p[i+1+k] = p[n-1-k];
            p[n-1-k] = temp;
        }

        printf("\nThe previous permutation is:\n");
        for(k=0; k<n; k++)
            printf("%d\t", *(p+k));
        }
    }
    int main()
    {
        int k, n;
        char c=' ';
        int p[N];
        //int p[] = {5, 1, 3, 4, 2}; // example from lecture 2
        // previous should be 5, 1, 3, 2, 4
        //int p2[] = {5, 3, 1, 2, 4}; // example from lecture 2
        // previous should be 5, 2, 4, 3, 1
        // {1, 2, 3, 4, 5} -> No previous permutation

        do{
            printf("Enter n: ");
            scanf("%d", &n);

            printf("Enter the elements of the permutation: ");
            for(k=0; k<n; k++)
                scanf("%d", &p[k]);

            printf("The permutation is:\n");
            for(k=0; k<n; k++)
                printf("%d\t", p[k]);

            previousPermutation(p, n);
            printf("\nIf you want to enter another exemple type y:");
            fflush(stdin);
            scanf("%c", &c);
```

```
   }while(c=='y');
   return 0;
}
```



```
Enter n: 5
Enter the elements of the permutation: 5 1 3 4 2
The permutation is:
5       1       3       4       2
The previous permutation is:
5       1       3       2       4
If you want to enter another exemple type y:y
Enter n: 5
Enter the elements of the permutation: 5 3 1 2 4
The permutation is:
5       3       1       2       4
The previous permutation is:
5       2       4       3       1
If you want to enter another exemple type y:n

Process returned 0 (0x0)   execution time : 43.973 s
Press any key to continue.
```