```
#include<stdlib.h>
#include<stdio.h>
#include<conio.h>
int Previous=145;
int Current=143;
int Check(int Variable)
{
        if(Variable>0)
        {
                return Variable;
        }
        else
        {
                return (Variable*-1);
        }
}
int main()
{
        int Queue[] = {86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130};
        int i,Start,Length=sizeof(Queue)/sizeof(int),Distance=0,Last,First,a,j;
         for (i = 0; i < Length; ++i)
        {
            for (j = i + 1; j < Length; ++j)
            {
                if (Queue[i] > Queue[j])
                {
                    a =  Queue[i];
                    Queue[i] = Queue[j];
                    Queue[j] = a;
                }
            }
        }
        Last = Current;
    if(Current>=Previous){
        for(i=0; i<Length; i++){
            if(Queue[i]>Current){
                Start=i;
                break;
            }
        }
        printf("Order    : %4d",Last);
        for(i=Start; i<Length; i++){
            printf(", %4d",Queue[i]);
             for(i=Start; i<Length; i++)
        for(i=Start; i>0; i--){
            printf(", %4d",Queue[i-1]);
            First = Queue[i];

            Distance+= Check(Last-First);
            printf("[%d]",Check(Last-First) );
            Last = First;
        }
}
}
      else if(Current<Previous){
        for(i=0; i<Length; i++){
            if(Queue[i]>Current){
                Start=i;
                break;
            }
        }
        printf("Sequence : %4d",Last);
        for(i=Start-1; i>=0; i--){
            printf(", %4d",Queue[i]);
```

```
            First = Queue[i];

            Distance+= Check(Last-First);
            printf("[%d]",Check(Last-First) );
            Last = First;
        }
        for(i=Start; i<Length; i++){
            printf(", %4d",Queue[i]);
            First = Queue[i];

            Distance+= Check(Last-First);
            printf("[%d]",Check(Last-First) );
            Last = First;
        }
    }
    printf("\nDistance : %d\n",Distance);
    return 0;
    getch();
}
```