/\* Dynamic Programming C/C++ implementation of LIS problem \*/

#include<stdio.h>

#include<stdlib.h>

/\* lis() returns the length of the longest increasing

  subsequence in arr[] of size n \*/

int lis( int arr[], int n )

{

    int \*lis, i, j, max = 0;

    lis = (int\*) malloc ( sizeof( int ) \* n );

    /\* Initialize LIS values for all indexes \*/

    for (i = 0; i < n; i++ )

        lis[i] = 1;

    /\* Compute optimized LIS values in bottom up manner \*/

    for (i = 1; i < n; i++ )

        for (j = 0; j < i; j++ )

            if ( arr[i] > arr[j] && lis[i] < lis[j] + 1)

                lis[i] = lis[j] + 1;

    /\* Pick maximum of all LIS values \*/

    for (i = 0; i < n; i++ )

        if (max < lis[i])

            max = lis[i];

    /\* Free memory to avoid memory leak \*/

    free(lis);

    return max;

}

/\* Driver program to test above function \*/

int main()

{

    int arr[] = { 10, 22, 9, 33, 21, 50, 41, 60 };

    int n = sizeof(arr)/sizeof(arr[0]);

    printf("Length of lis is %dn", lis( arr, n ) );

    return 0;

}