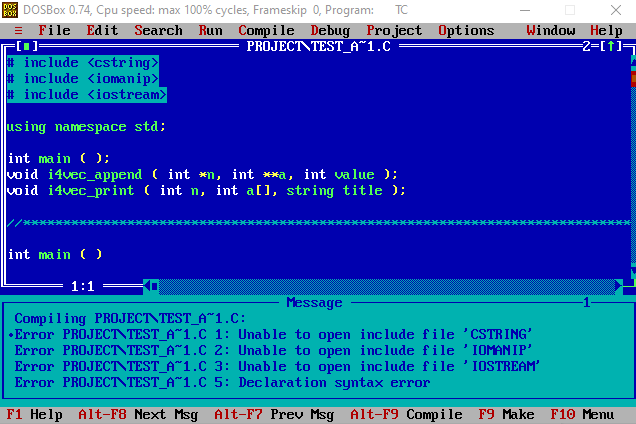
**Actividad 2.- Usar Turbo C++.**

Cargar, compilar y ejecutar un programa en Turbo C++.



Programa:

# include <cstring>

# include <iomanip>

# include <iostream>

using namespace std;

int main ( );

void i4vec\_append ( int \*n, int \*\*a, int value );

void i4vec\_print ( int n, int a[], string title );

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*80

int main ( )

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*80

//

// Purpose:

//

// MAIN is the main program for ARRAY\_APPEND.

//

// Discussion:

//

// Given an existing array of length N, we wish to append one entry

// to the end. This requires creating a new array one entry longer,

// copying the data, and switching names.

//

// This time, I didn't need Miro's help.

//

// Licensing:

//

// This code is distributed under the GNU LGPL license.

//

// Modified:

//

// 14 May 2018

//

// Author:

//

// John Burkardt

//

{

int \*a;

int i;

int n;

int value;

cout << "\n";

cout << "ARRAY\_APPEND:\n";

cout << " C++ version\n" ;

cout << " Append one entry to an array.\n";

//

// Initialize the array.

//

n = 5;

a = new int[n];

for ( i = 0; i < n; i++ )

{

a[i] = i;

}

//

// Display the current array.

//

i4vec\_print ( n, a, " A, as initialized:" );

//

// Append the value 17 to the array.

//

value = 17;

i4vec\_append ( &n, &a, value );

//

// Display the modified array.

//

i4vec\_print ( n, a, " A, after appending" );

//

// Free memory.

//

delete [] a;

//

// Terminate.

//

cout << "\n";

cout << "ARRAY\_APPEND:\n";

cout << " Normal end of execution.\n";

return 0;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*80

void i4vec\_append ( int \*n, int \*\*a, int value )

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*80

//

// Purpose:

//

// I4VEC\_APPEND appends an entry to an I4VEC.

//

// Licensing:

//

// This code is distributed under the GNU LGPL license.

//

// Modified:

//

// 14 May 2018

//

// Author:

//

// John Burkardt

//

// Parameters:

//

// Input/output, int \*N, the current size of the array. On output,

// the array is one entry longer.

//

// Input/output, int \*\*A, the array. On output, the array has had

// VALUE appended.

//

// Input, int VALUE, a value to be appended to A.

//

{

int \*a\_old;

int i;

//

// Save a pointer to the old array.

//

a\_old = \*a;

//

// Create a new array.

//

\*a = new int[\*n+1];

//

// Copy the old data and append the new item.

//

for ( i = 0; i < \*n; i++ )

{

(\*a)[i] = a\_old[i];

}

(\*a)[\*n] = value;

//

// Increase N.

//

\*n = \*n + 1;

//

// Free memory.

//

delete [] a\_old;

return;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*80

void i4vec\_print ( int n, int a[], string title )

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*80

//

// Purpose:

//

// I4VEC\_PRINT prints an I4VEC.

//

// Discussion:

//

// An I4VEC is a vector of I4's.

//

// Licensing:

//

// This code is distributed under the GNU LGPL license.

//

// Modified:

//

// 14 November 2003

//

// Author:

//

// John Burkardt

//

// Parameters:

//

// Input, int N, the number of components of the vector.

//

// Input, int A[N], the vector to be printed.

//

// Input, string TITLE, a title.

//

{

int i;

cout << "\n";

cout << title << "\n";

cout << "\n";

for ( i = 0; i < n; i++ )

{

cout << " " << setw(8) << i

<< ": " << setw(8) << a[i] << "\n";

}

return;

}