LOGIC CODE GENERATOR

Technical Resource

Logic Coder

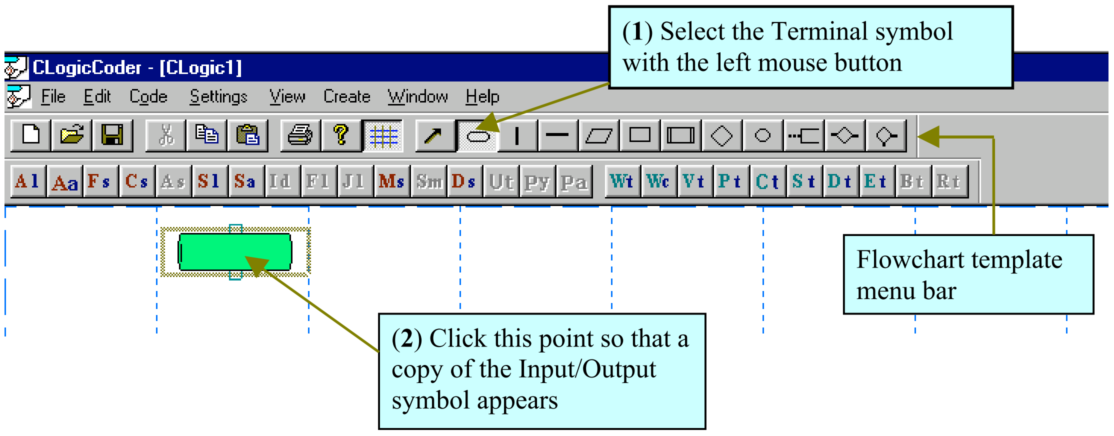
LD SLG +SCS SP

TUTORIAL FOR PROGRAM FLOWCHART

Logic Code Generator Ltd
Suite 501, International House
223 Regent Street
London W1B 2QD
England

USING LOGICCODER TO DRAW THE PROGRAM FLOWCHART

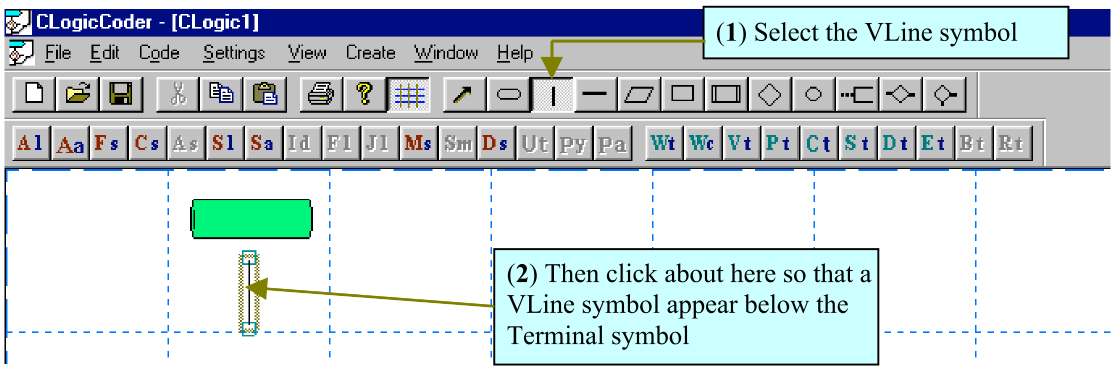
STEP 1:
Select the Input/Output symbol on the symbol template bar and then click the point in the user document interface where you want a copy of it to appear. The figure below gives further illustrative explanation.



CLogicCoder - [CLogic1]
File Edit Code Settings View Create Window Help
A1 Aa Fs Cs As S1 Sa Id F1 J1 Ms Sm Ds Ut Py Pa Wt Wc Vt Pt Ct St Dt Et Bt Rt
(1) Select the Terminal symbol with the left mouse button
Flowchart template menu bar
(2) Click this point so that a copy of the Input/Output symbol appears

Figure 2
Click at the point in the user document window where you want the selected symbol to appear.

STEP 2:
Select the VLine symbol and then click below the middle of the Terminal symbol so that a VLine appear as illustrated in figure 3.



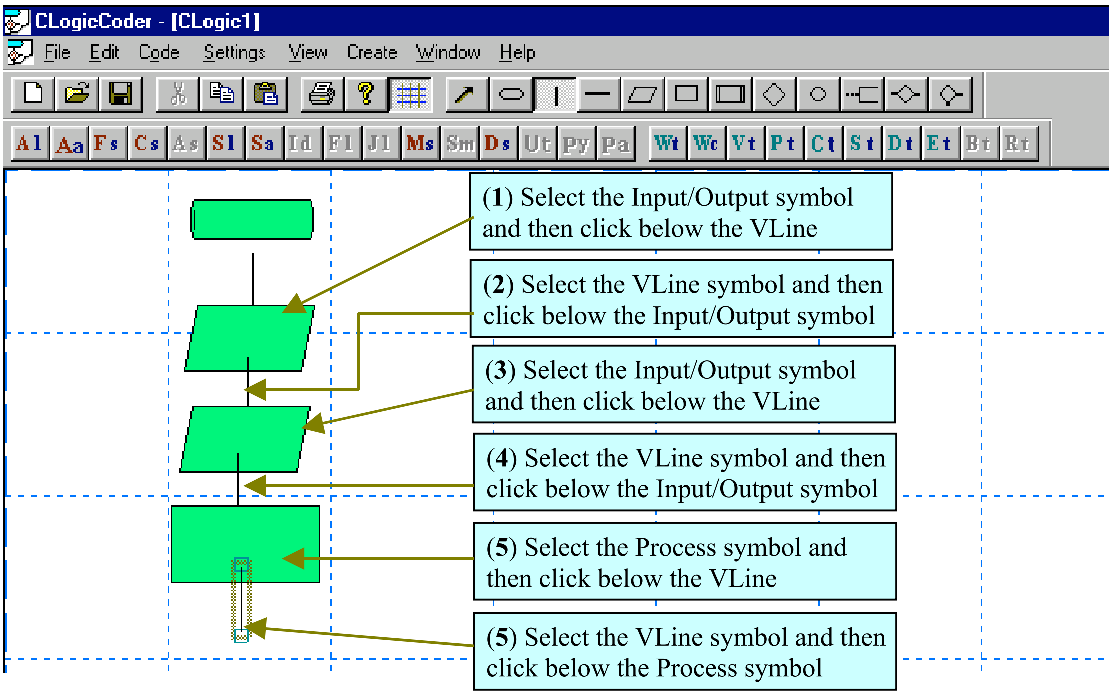
CLogicCoder - [CLogic1]
File Edit Code Settings View Create Window Help
A1 Aa Fs Cs As S1 Sa Id F1 J1 Ms Sm Ds Ut Py Pa Wt Wc Vt Pt Ct St Dt Et Bt Rt
(1) Select the VLine symbol
(2) Then click about here so that a VLine symbol appear below the Terminal symbol

Figure 3
LogicCoder allow you to place interconnecting lines between two symbols without having to connect them exactly. The system provides function that does automatic interconnection of these lines.

Whenever you click in the user document window with a selected flowchart symbol from the flowchart template toolbar, either of the following 2 things happen.
(1) A copy of the selected flowchart symbol appears in the user document window at the point where you click.

(2) The flowchart symbol at the point where you click is selected in the user document window.

STEP 3:
Select and place a copy of each flowchart symbol so that your document window appears similar to that illustrated in figure 4. Do not worry if interconnecting lines are not neatly place. I will show you how to use the Autoadjust functions to fix their interconnection.


CLogicCoder - [CLogic1]
File Edit Code Settings View Create Window Help
Al Aa Fs Cs As Sl Sa Id Fl Jl Ms Sm Ds Ut Py Pa Wt Wc Vt Pt Ct St Dt Et Bt Rt

(1) Select the Input/Output symbol and then click below the VLine

(2) Select the VLine symbol and then click below the Input/Output symbol

(3) Select the Input/Output symbol and then click below the VLine

(4) Select the VLine symbol and then click below the Input/Output symbol

(5) Select the Process symbol and then click below the VLine

(5) Select the VLine symbol and then click below the Process symbol

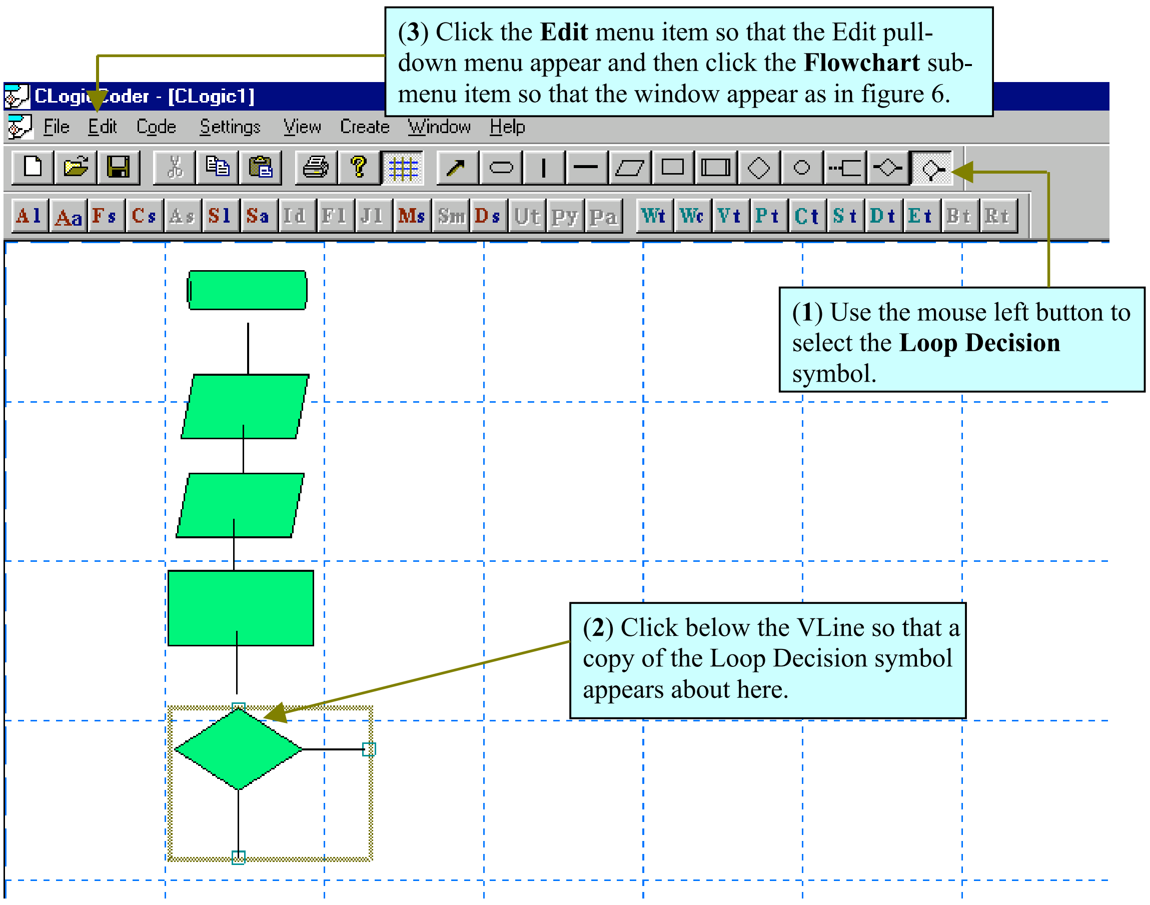
Figure 4
Select each symbol in sequence and then click the location in the user document where you want them to appear.

STEP 4:
I now show you how to insert and then fragment a compound symbol when drawing a flowchart.

The last two symbols on the flowchart template bar are example compound symbols that you can use to quickly draw loops and case structures when implementing decision making control logic. More advance versions of LogicCoder uses other compound symbols to represent other complex substructures such as start and end terminal symbols or a closing connector.

Select the Loop Decision symbol on the template menu bar and then click below the last VLine symbol in the flowchart so that a Loop Decision appear as illustrated in figure 5.

Notice that a selected symbol appears with a yellow rectangle around it. In the next step, you will use the **Fragment Structure** function to break the compound symbol into it‰constituent. You should always break compound symbols into their components before an attempt to use the flowchart to generate a source program. In addition, you should ensure that only the symbol you intend to fragment is selected before you attempt to execute this function.


CLogicCoder - [CLogic1]
File Edit Code Settings View Create Window Help
Al Aa Fs Cs As Sl Sa Id Fl Jl Ms Sm Ds Ut Py Pa Wt Wc Vt Pt Ct St Dt Et Bt Rt

(**3**) Click the **Edit** menu item so that the Edit pull-down menu appear and then click the **Flowchart** sub-menu item so that the window appear as in figure 6.

(**1**) Use the mouse left button to select the **Loop Decision** symbol.

(**2**) Click below the VLine so that a copy of the Loop Decision symbol appears about here.

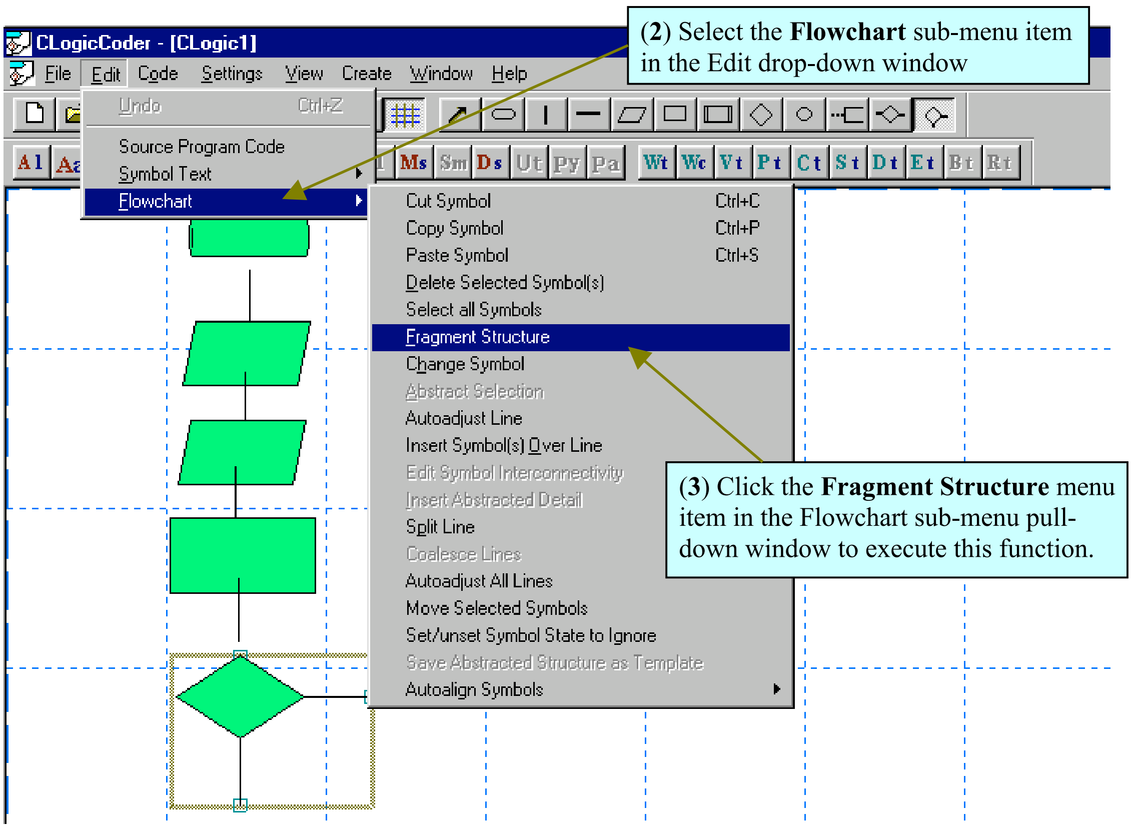
Figure 5
You use compound symbols to quickly draw flowcharts that have branch points or decisions in them.

STEP 5:
In this step you will use the **Fragment Structure** function to break the Loop Decision into its component symbols.

(1) Select the **Edit** menu item on the main menu bar and then select the **Flowchart** sub-menu item so that the pull-down window illustrated in figure 6 appear.

(2) Select the **Fragment Structure** menu item by clicking with the mouse left button or use the arrow keys to highlight it and then press the Enter key once.

Your document window should now appear similar to that illustrated in figure 7. You can also execute the **Fragment Structure** function by clicking on the shortcut toolbar button that is labelled **Fs,** see figure 7.



(**2**) Select the **Flowchart** sub-menu item in the Edit drop-down window

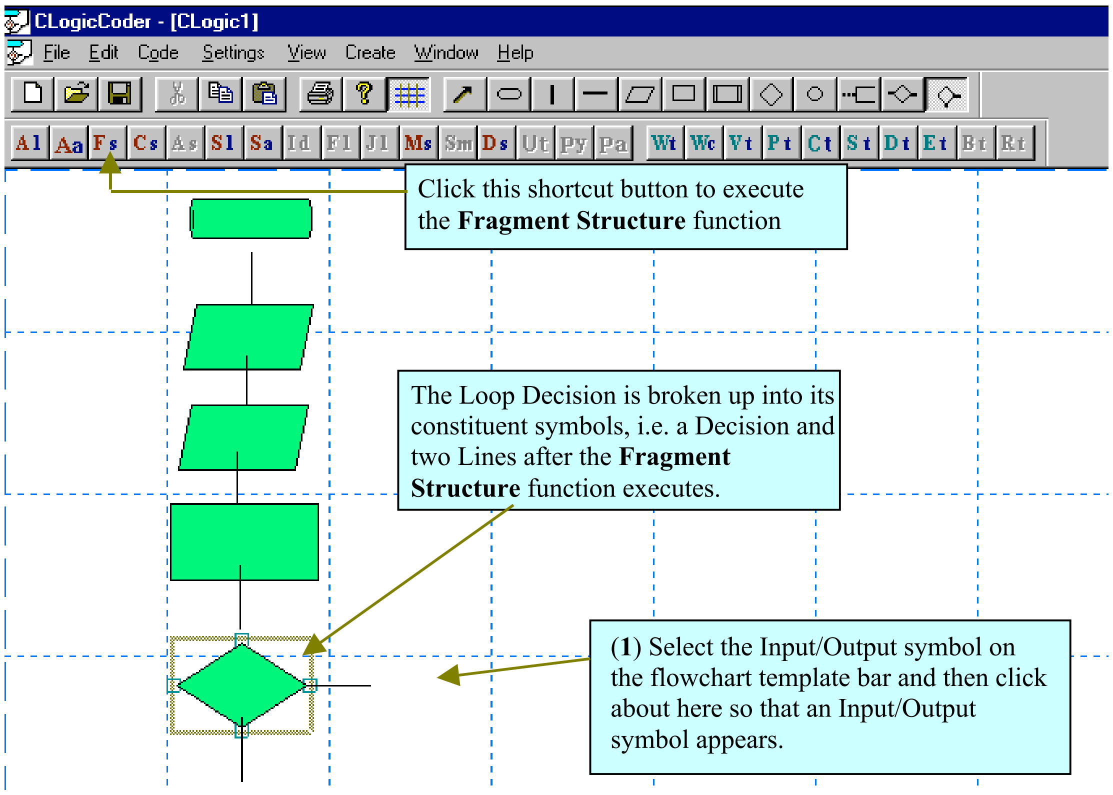
(**3**) Click the **Fragment Structure** menu item in the Flowchart sub-menu pull-down window to execute this function.

Figure 6
You execute the Fragment Structure function to break a compound symbol into its constituent symbols.

In the next step you will complete the flowchart so that it implement the control logic as illustrated in figure 1. You will then add descriptive and source code text to each symbol in the flowchart. The description is that illustrated in figure 1. The descriptive text is the same as the Algorithmic text.

When you start or open a flowchart document, LogicCoder by default sets the text view to Algorithmic. The Algorithmic text view of a flowchart allows you to see description of each instruction in the algorithm illustrated by the flowchart. In the section of this tutorial headed: **SAVING THE SOURCE CODE TEXT FILE,** I will show you how to switch the text view of a flowchart. The ı Source Codeˆ view of a flowchart presents actual program command statement(s) in each symbol in the flowchart that is to be executed in the required program. Each program command statement matches a description in a symbol.

You can also save the whole source code text view or load a whole source code text view for a given flowchart from backing storage. The source code text view you load can be for a single language or it can be for different programming languages. Therefore, you can use a single flowchart to write programs in different languages or you can use the same flowchart to write various forms of a program in the same source language.


CLogicCoder - [CLogic1]
File Edit Code Settings View Create Window Help
Al Aa Fs Cs As Sl Sa Id Fl Jl Ms Sm Ds Ut Py Pa Wt Wc Vt Pt Ct St Dt Et Bt Rt

Click this shortcut button to execute the **Fragment Structure** function

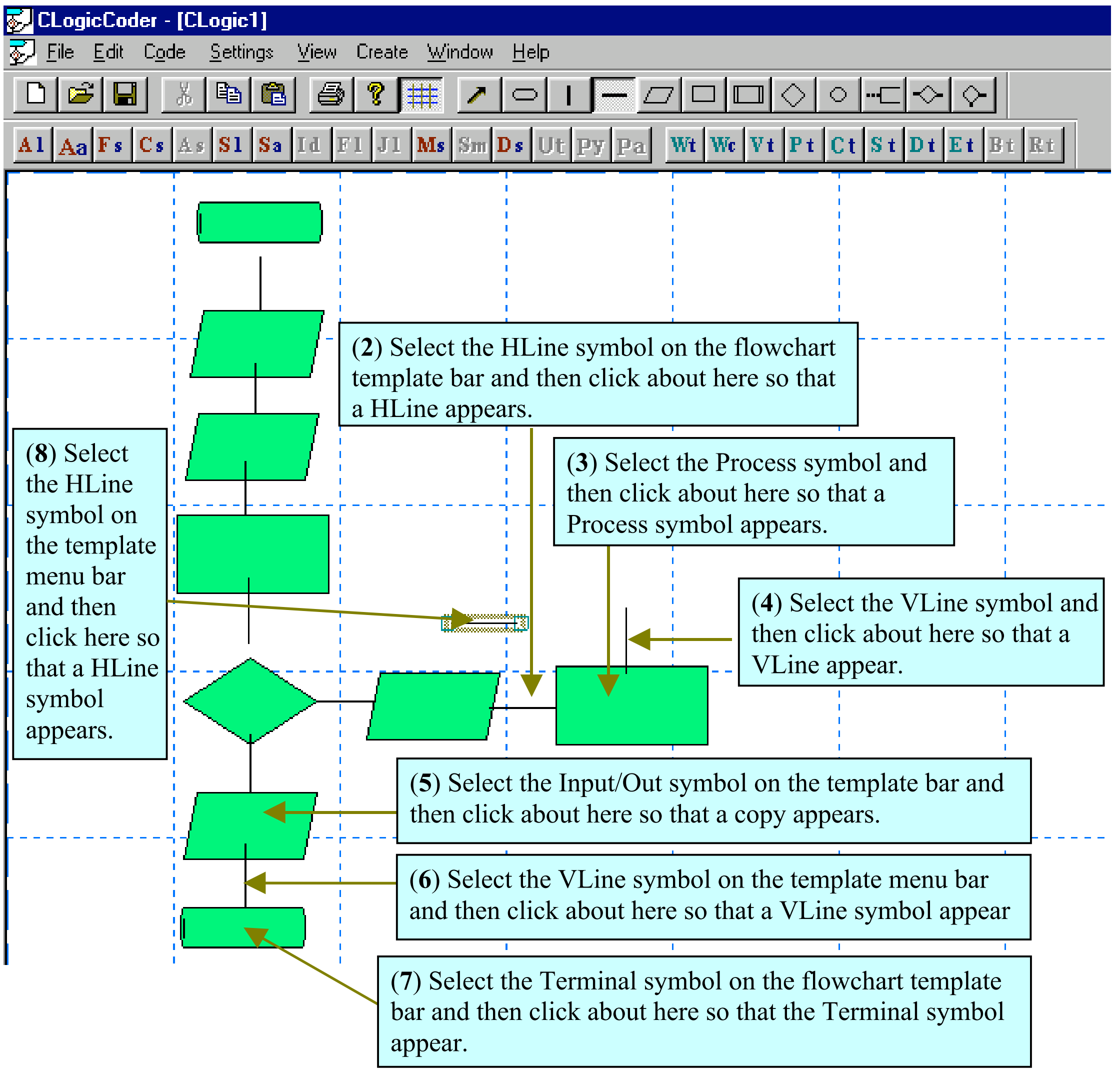
The Loop Decision is broken up into its constituent symbols, i.e. a Decision and two Lines after the **Fragment Structure** function executes.

(1) Select the Input/Output symbol on the flowchart template bar and then click about here so that an Input/Output symbol appears.

Figure 7
A Loop Decision symbol consists of a Decision symbol, a VLine and an HLine symbol as illustrated here.

In the next step, you will complete the flowchart so that it looks similar to that in figure 1

STEP 6
Select each required symbol on the flowchart template bar and then click them in position as illustrated in figures 7 and figure 8.


CLogicCoder - [CLogic1]
File Edit Code Settings View Create Window Help
Al Aa Fs Cs As Sl Sa Id Fl Jl Ms Sm Ds Ut Py Pa Wt Wc Vt Pt Ct St Dt Et Bt Rt

(**2**) Select the HLine symbol on the flowchart template bar and then click about here so that a HLine appears.

(**3**) Select the Process symbol and then click about here so that a Process symbol appears.

(**4**) Select the VLine symbol and then click about here so that a VLine appear.

(**5**) Select the Input/Out symbol on the template bar and then click about here so that a copy appears.

(**6**) Select the VLine symbol on the template menu bar and then click about here so that a VLine symbol appear

(**7**) Select the Terminal symbol on the flowchart template bar and then click about here so that the Terminal symbol appear.

(**8**) Select the HLine symbol on the template menu bar and then click here so that a HLine symbol appears.

Figure 8
With LogicCoder you simply click each flowchart symbol into place and then use the Autoadjust function to adjust their interconnecting lines.

In the next step you will use the Autoadjust function to cause the interconnecting lines to adjust themselves so that they connect their enclosing symbols in pairs.

STEP 7:
(1) Select the **Edit** menu item on the main menu bar and then select the **Flowchart** item in the pull-down sub-menu that appears.

At this point, your document window should appear similar to that in figure 9.

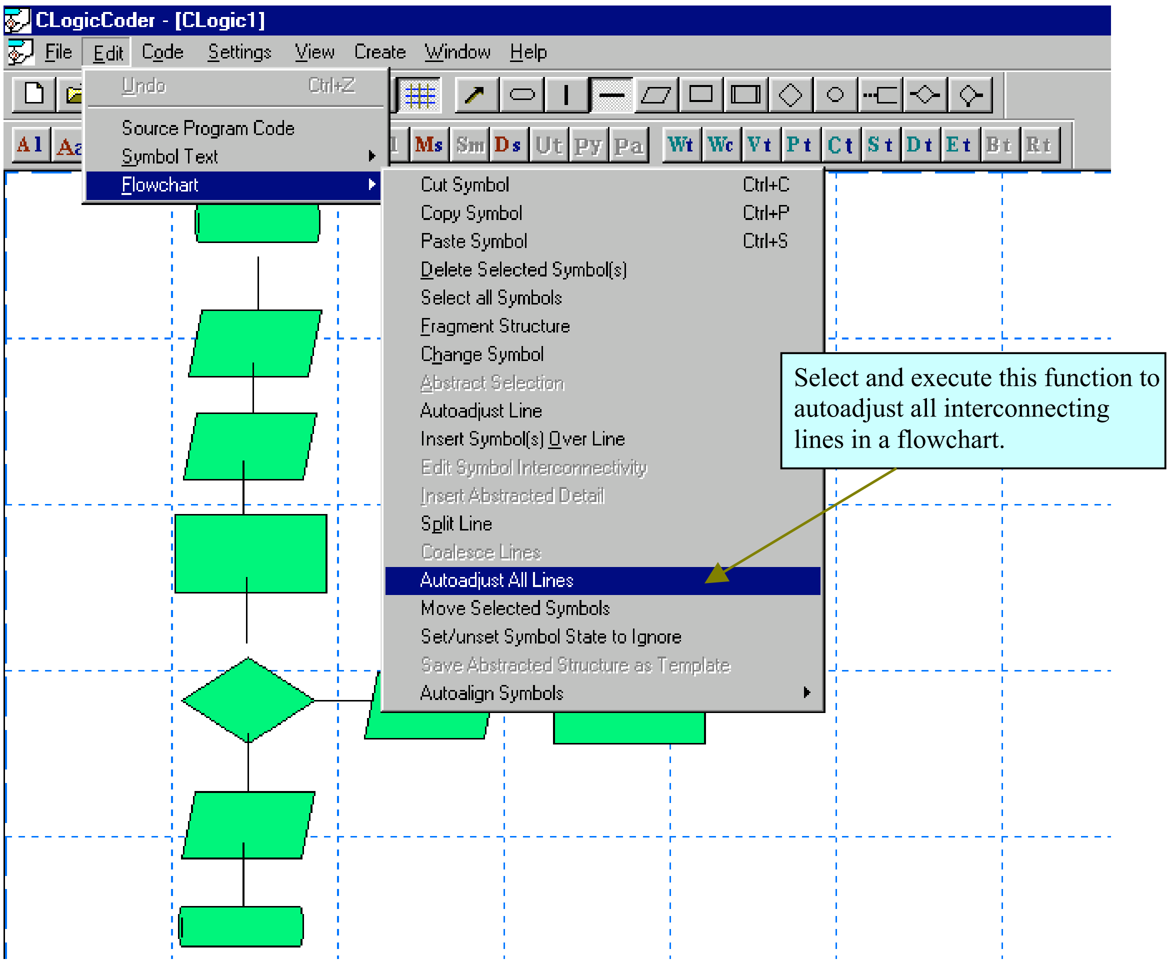


Figure 9
You can adjust individually selected lines or you can adjust all lines in a flowchart by execution of a single function.

(1) Use the mouse left button to click the **Autoadjust All Line** menu item so that the function execute.

At the end of executing this function, your flowchart should look similar to that illustrated in figure 10.

You can also execute the **Autoadjust All Lines** functions in a flowchart by clicking on the shortcut button on the Flowchart Edit shortcut toolbar. To do this, you click the button labelled **Aa**.

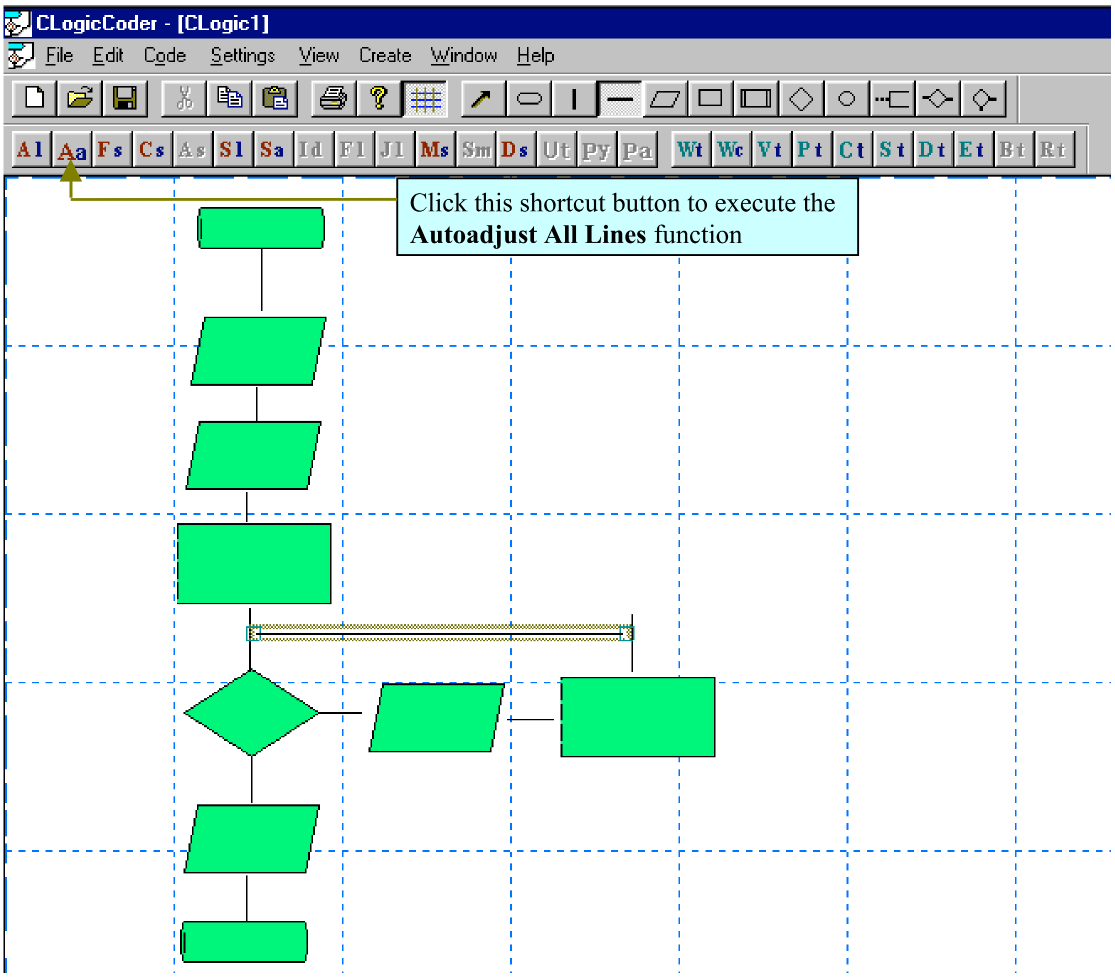

CLogicCoder - [CLogic1]
File Edit Code Settings View Create Window Help
Al Aa Fs Cs As Sl Sa Id Fl Jl Ms Sm Ds Ut Py Pa Wt Wc Vt Pt Ct St Dt Et Bt Rt
Click this shortcut button to execute the **Autoadjust All Lines** function

Figure 10
When the autoadjust function is executed on each line symbol, it is adjusted so that it connects exactly to the interconnecting symbol(s).

More advance versions of LogicCoder also provides other autoadjust functions that allow you to align groups of symbols vertical or horizontal along common axes. There are also functions that allow you to align connectors with case decisions in a complex flowchart.

ADDING TEXT TO EACH SYMBOL

STEP 1:
Use the mouse left button to click on the **Select** button on the flowchart template toolbar so that it appears sunken as illustrated in figure 11.

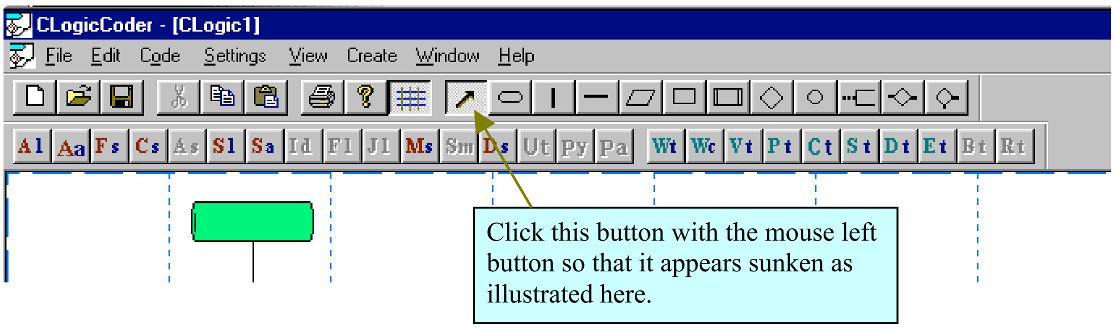

CLogicCoder - [CLogic1]
File Edit Code Settings View Create Window Help
Al Aa Fs Cs As Sl Sa Id Fl Jl Ms Sm Ds Ut Py Pa Wt Wc Vt Pt Ct St Dt Et Bt Rt
Click this button with the mouse left button so that it appears sunken as illustrated here.

Figure 11
Selecting the Select button on the Template toolbar ensures that no flowchart symbol is selected.

If you click in the user document window whenever the Select button is selected, LogicCoder will not draw a flowchart symbol. In this state, LogicCoder will highlight the symbol you click on. In this case, the symbol is selected. Any previously selected symbol is unselected.

You can also set the Select button to select a group of symbols by liasing with the mouse left button. In this case, you press and drag the mouse left button so that the set of symbols you want to select appears within the outlined rectangle that appear as you drag with the mouse left button. Figure 24 illustrates how this is done.

STEP 2
Use the mouse left button to click the first terminal symbol in the flowchart so that it is selected.

A yellow rectangular box appears around a symbol whenever it is selected in a flowchart document.

In the next step, you will execute the **Write Symbol Text** function to place ı Algorithmicˆ and ı Source Codeˆ text in a flowchart symbol.

STEP 3:
(1) Use the mouse left button to click on the **Edit** item on the main menu bar so that Edit pull-down sub-menu appear.

(2) Click on the **Symbol Text** item in the Edit sub-menu so that a subsidiary pull-down menu appears as illustrated in figure 12.

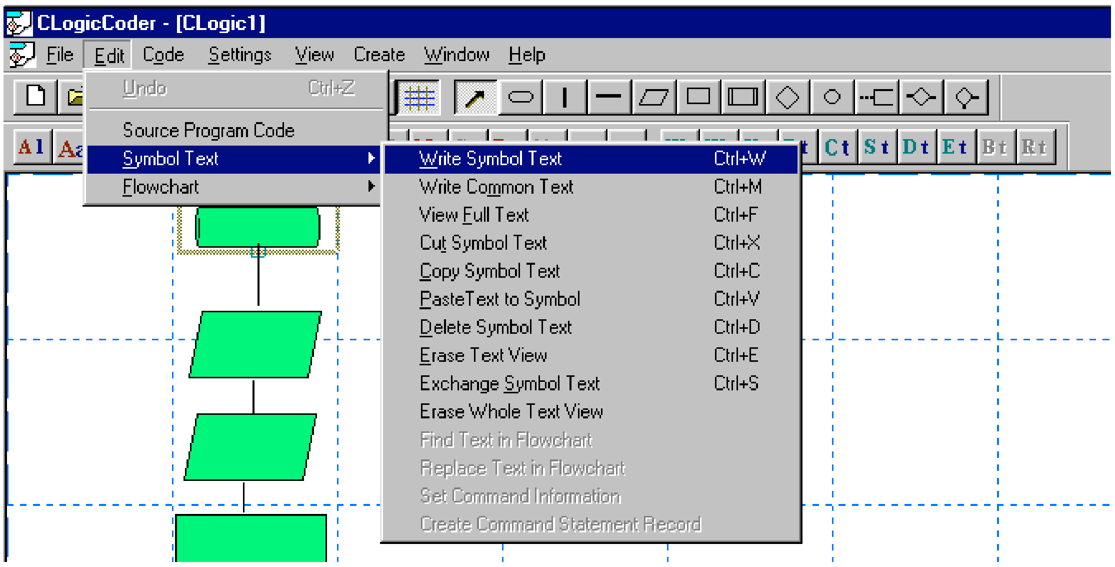


Figure 12
You use the Write Symbol Text function to place text values in selected flowchart symbols.

STEP 4:
Select the **Write Symbol Text** menu item so that the function execute.

At this point, you should have a document window similar to that illustrated in figure 13.

The Write Symbol Text function allow you to place text in symbols by either writing the text value through the two text editing boxes or you can select a text value from a group of values that is appropriate for the selected symbol. Notice that the dialog window presented by this function also display the kind of selected symbol whenever it executes. This function allows you to place either a ı Yesˆ or a ı Noˆ value in certain line symbols. The selected line symbol must be connected to a non Switch-Case decision symbol. In this case, if you select either of the two connected lines and then place a text value, the function will search for the other connected line symbol and then place the complementary text value in it.

LogicCoder also allow you to place text in other kinds of line symbols that form part of the entry in a select case structure. However, this version of LogicCoder does not allow you to implement flowcharts with the select case structure.

STEP 5:
Click the button labelled **START** in the **Terminal Text** group of buttons so that it appear sunken as illustrated in figure 13 and then click the OK button.

At this point, LogicCoder places the text value ı STARTˆ in the ı Algorithmicˆ and the ı Source Codeˆ text views of the selected symbol.

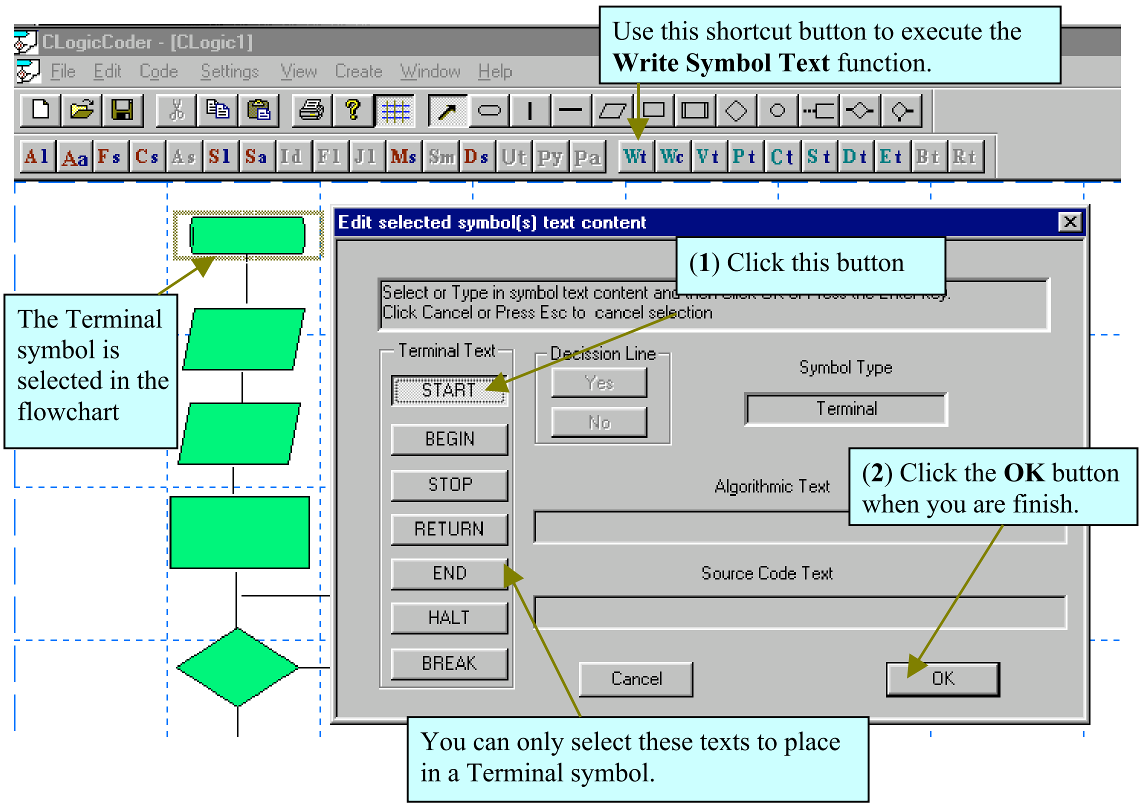

Use this shortcut button to execute the Write Symbol Text function.
CLogicCoder - [CLogic1]
File Edit Code Settings View Create Window Help
Al Aa Fs Cs As Sl Sa Id Fl Jl Ms Sm Ds Ut Py Pa Wt Wc Vt Pt Ct St Dt Et Bt Rt
Edit selected symbol(s) text content
(1) Click this button
Select or Type in symbol text content and then click OK or Press the Enter key.
Click Cancel or Press Esc to cancel selection
Terminal Text
START
BEGIN
STOP
RETURN
END
HALT
BREAK
Decission Line
Yes
No
Symbol Type
Terminal
Algorithmic Text
Source Code Text
Cancel
OK
(2) Click the OK button when you are finish.
The Terminal symbol is selected in the flowchart
You can only select these texts to place in a Terminal symbol.

Figure 13
You can execute the Write Symbol Text function by use of the shortcut button labelled Wt as illustrated here.

Refer to figure 1 for a quick summary review of the description of the program algorithm to be implemented by our flowchart.

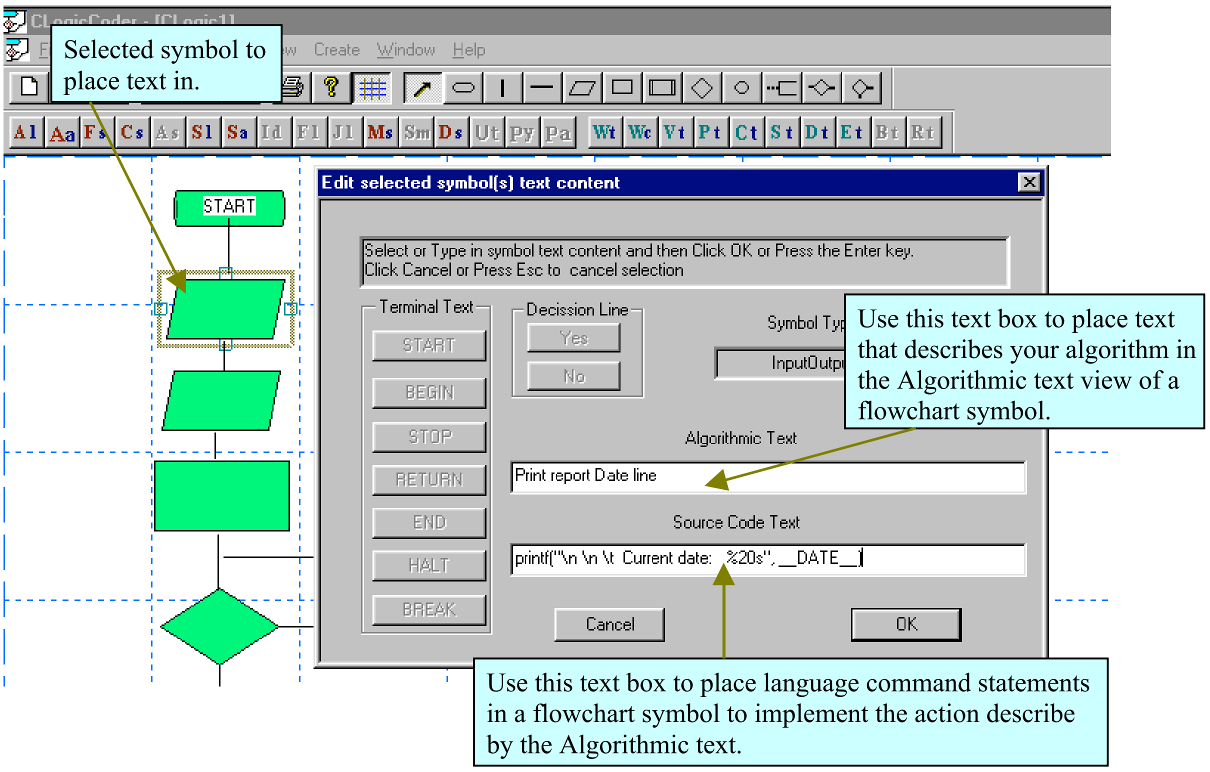
In the next few steps, you will place source code command statements and descriptive text values in the sequence of symbols that follows.

STEP 6:
Use the mouse left button to select the first Input/Output symbol in the flowchart. That is, the Input/Output symbol that immediately follows the first Terminal symbol.

Recall that whenever we read the control logic given in a flowchart, we read from top to bottom and from left to right. Therefore, we do not use arrowhead flow-lines and we do not have left handed loops. This convention helps to simplify the analysis and evaluation of the control logic in a flowchart.

STEP 7:
Use the shortcut button on the **Edit** function shortcut toolbar to execute the **Write Symbol Text** function.

At this point, your document window should look similar to that illustrated in figure 14



Selected symbol to place text in.

Select or Type in symbol text content and then Click OK or Press the Enter key.
Click Cancel or Press Esc to cancel selection

Use this text box to place text that describes your algorithm in the Algorithmic text view of a flowchart symbol.

Algorithmic Text: Print report Date line

Source Code Text: printf("\n \n \t Current date: %20s", __DATE__)

Use this text box to place language command statements in a flowchart symbol to implement the action describe by the Algorithmic text.

Figure 14
The Write Symbol Text function allows you to simultaneously place descriptive and language command statement text in a flowchart symbol. LogicCoder switches off other text entry input window whenever you are entering text into an Input/Output, Process, Decision or Predefine symbol.

STEP 8:
Click on the Algorithmic Text edit box so that it receive the focus and then enter the following descriptive text. **Print report Date line**

The Algorithmic text editor allow you to enter text that describe the event or process that is to take place at that point in the program sequence represented by the flowchart symbol.

STEP 9:
(1) Click on the Source Code text edit box so that it receives the input focus and then type in the following C/C++ source code.

```
printf("\n \n \t  Current date:   %20s", __DATE__)
```

(2) Click the OK button in the Write Symbol text dialog window when you are finish.

LogicCoder place the text you entered into the flowchart symbol. The descriptive text is display in the flowchart symbol as illustrated in figure 15.

The command statement is the actual instruction in a source language that tells the computer to do the thing that you describe with the Algorithmic text.

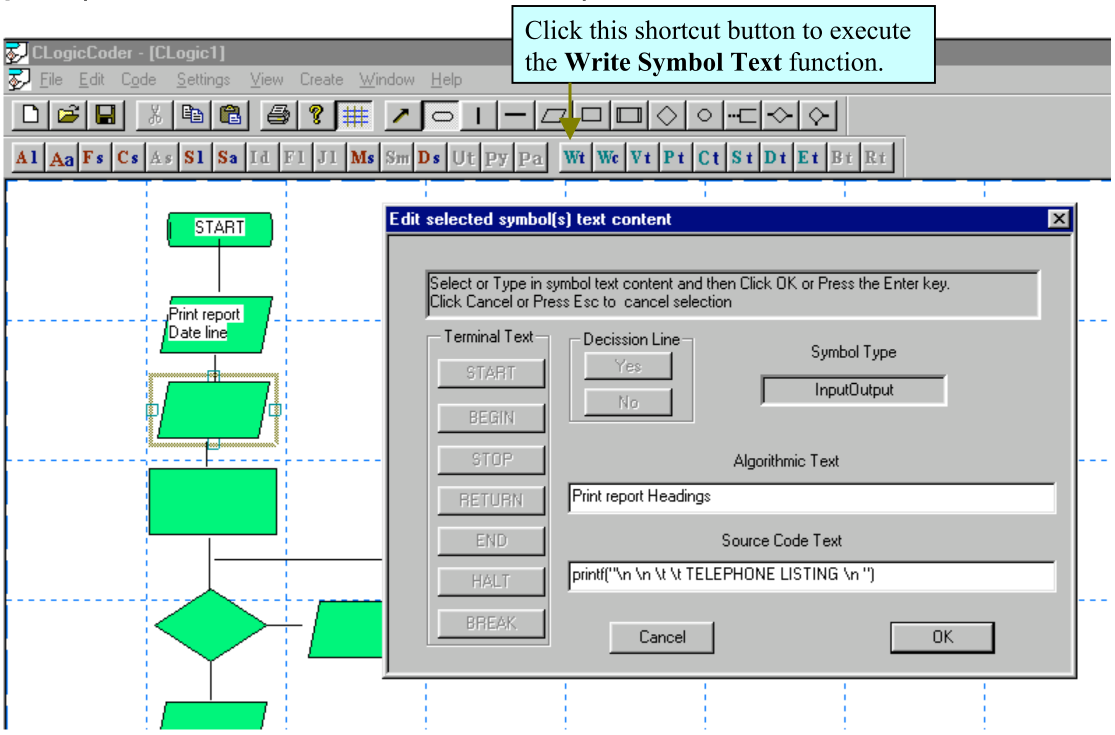
You do not need to terminate a command statement with a statement terminator such as ı ;ˆ in the case of C/C++ or similar languages.

STEP 10:
Click on the second Input/Output symbol and then execute the **Write Symbol Text** function once more.

STEP 11:
Place the following text in each text view of the flowchart symbol.

Algorithmic text
Print report Headings

Source Code text
printf("\n \n \t \t TELEPHONE LISTING \n ")



Click this shortcut button to execute the **Write Symbol Text** function.
CLogicCoder - [CLogic1]
File Edit Code Settings View Create Window Help
Al Aa Fs Cs As Sl Sa Id Fl Jl Ms Sm Ds Ut Py Pa Wt Wc Vt Pt Ct St Dt Et Bt Rt
START
Print report
Date line
Edit selected symbol(s) text content
Terminal Text
START
BEGIN
STOP
RETURN
END
HALT
BREAK
Decission Line
Yes
No
Cancel
OK

Select or Type in symbol text content and then Click OK or Press the Enter key.
Click Cancel or Press Esc to cancel selection

Symbol Type: InputOutput

Algorithmic Text: Print report Headings

Source Code Text: printf("\n \n \t \t TELEPHONE LISTING \n ")

Figure 15
You can use the Wt shortcut button on the document Edit shortcut toolbar to execute the Write Symbol Text function.

Click the OK button when you are finish entering the text values listed above.

At this point, your flowchart should look similar to that illustrated in figure 16.

In the next step you will select each symbol in the sequence I have labelled them and then place the listed text below in them.

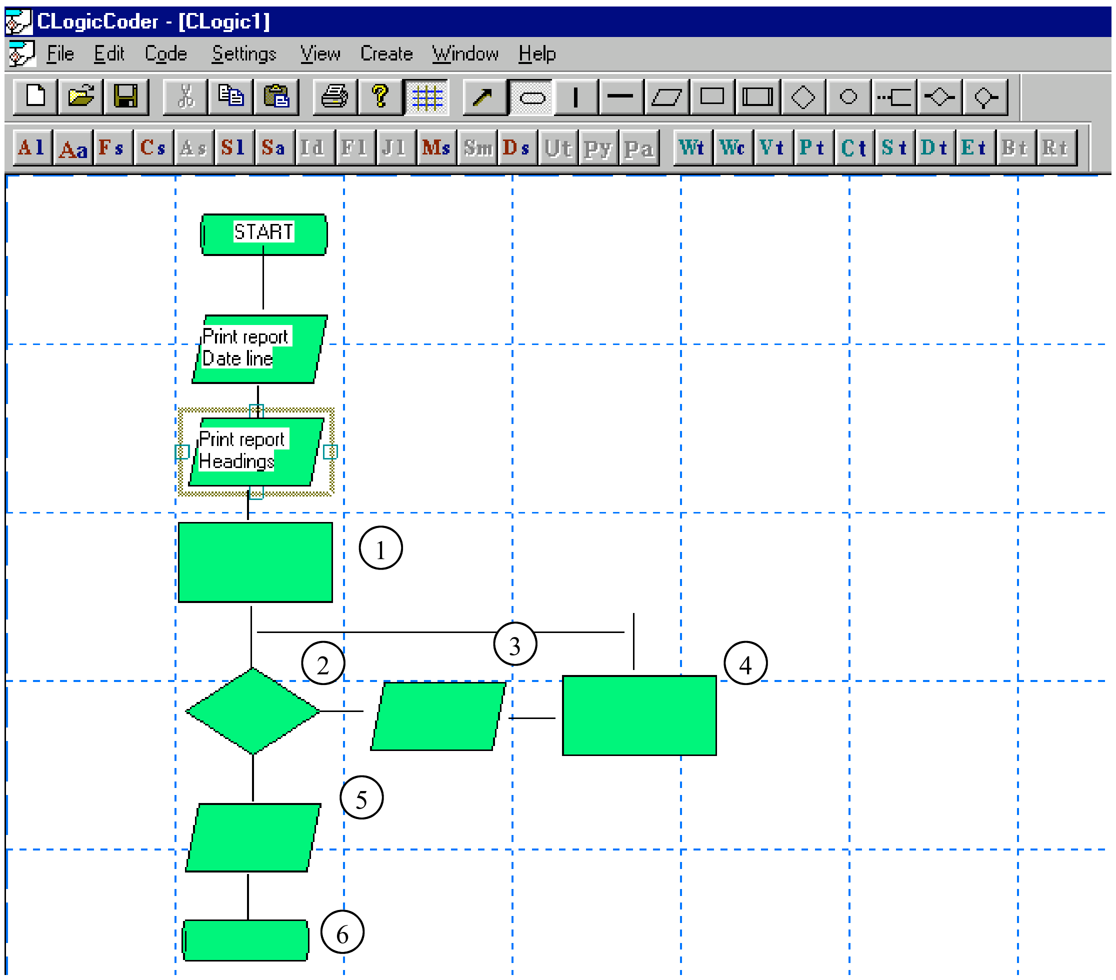

CLogicCoder - [CLogic1]
File Edit Code Settings View Create Window Help
Al Aa Fs Cs As Sl Sa Id Fl Jl Ms Sm Ds Ut Py Pa Wt Wc Vt Pt Ct St Dt Et Bt Rt
START
Print report
Date line
Print report
Headings
1
2
3
4
5
6

Figure 16
There are other functions available in LogicCoder that makes it easy to place, move, search and manipulate the text content of multiple symbols in a flowchart.

STEP 12:
Select each symbol in the order listed by the labels as illustrated in figure 16 and then place the text content listed below in the same order in each of them.

①
Algorithmic text
Initialise loop control variables

Source Code text
for r = 0

Notice that in this case, the source code text is prefixed with the ı forˆ statement. This prefix is a hint to LogicCoder telling it that you want the loop control in the flowchart to be implemented as a ı forˆ statement. The alternative implementation is a while statement. LogicCoder will then use the expression that follows the ı forˆ statement as the initialising statement in the complete expression that constitute the head of the ı forˆ statement. The sample generated source code in figure 22 illustrates this. There are several ways in which you can hint LogicCoder to implement a loop with use of the ı forˆ statement as opposed to the while statement. LogicCoder use the while statement as the default implementation for the standard loop control logic.

In this case, LogicCoder also checks for a Process symbol at the end of the loop and uses its source code content as the 3rd statement in the head of the ı forˆ statement as required by C/C++ syntax. If there are no Process symbol at the end of the body of the loop, LogicCoder ignores the situation and only uses the content of the loop decision to complete the control statement in the for expression.

2
Algorithmic text
End of file?

Source Code text
strcmp(Telefile[r].Name, END) != 0

At this point the program check to determine if the last record in the input data file has been read. If this is indeed the case, then the strcmp( ) function returns a non-zero value to the ı forˆ statement. The ı forˆ statement then pass control to the first instruction outside the body of the loop. Hence, the loop is terminated at this point.

3
Algorithmic text
Print report line

Source Code text
printf("\n\t%i \t%-s \t%s", Telefile[r].Code, Telefile[r].Num, Telefile[r].Name)

At this point in the program, a line consisting of data values taken from the current record is printed on the output report. The C printf( ) function is used to do this. LogicCoder provides you with the **Write Common Text function** that allows you to enter the same text value into multiple selected symbols all at one go. You can then select individual symbol and edit their content.

4
Algorithmic text
Increment record counter

Source Code text
r++

At this point, the file record counter is to be incremented by 1 so that the next record in the input data file is referenced.

Recall that at step 1 we prefixed the initialising control variable with a ı forˆ statement!. In this case, you need not always put a control variable expression of this form at this point. You could place the complete ı forˆ expression in the Decision symbol and switch off the 2 Process symbols. You switch off a symbol in a flowchart by placing it in the ignore state.

LogicCoder ignores a symbol in the ı ignoreˆ state when it generates a source program from the flowchart. In this case, it passes over the symbol or it generates a null statement in the source code.

5

Algorithmic text
Print end of listing message

Source Code text
printf("\n \n \t END OF TELEPHONE LISTING:")

The last printf( ) statement is used to print the end of listing message on the output report. You could use the **Write Common Text** function to enter a common printf( ) statement in all symbols where it is used and then edit them to implement the specific function required.

6

Algorithmic text
END

Source Code text
END

STEP 13:
Select the HLine that follows the Decision symbol into the body of the loop and then execute the **Write Symbol Text** function.

Click the ı Noˆ button in the Write Symbol Text dialog window that appear and then press the Enter key or click the OK button.

Your document window should look similar to that in figure 17 before you click the OK button or press then Enter key.

Notice that LogicCoder places the ı Yesˆ text on the alternative line that follows the Decision symbol. LogicCoder search for a corresponding line that emanates from a Decision symbol and then place the complementary text in it whenever you choose to enter a ı Yesˆ or a ı Noˆ text in a line symbol that connects from the Decision symbol. Both line symbols must flow from the Decision symbol. LogicCoder will not allow you to place text in a line symbol that flows into a Decision symbol.

LogicCoder does not allow you to place such text in a line symbol that follows a Switch° case Decision symbol. This is because the control logic implemented by a Switch-case Decision is different from that implemented by a loop or a case decision.

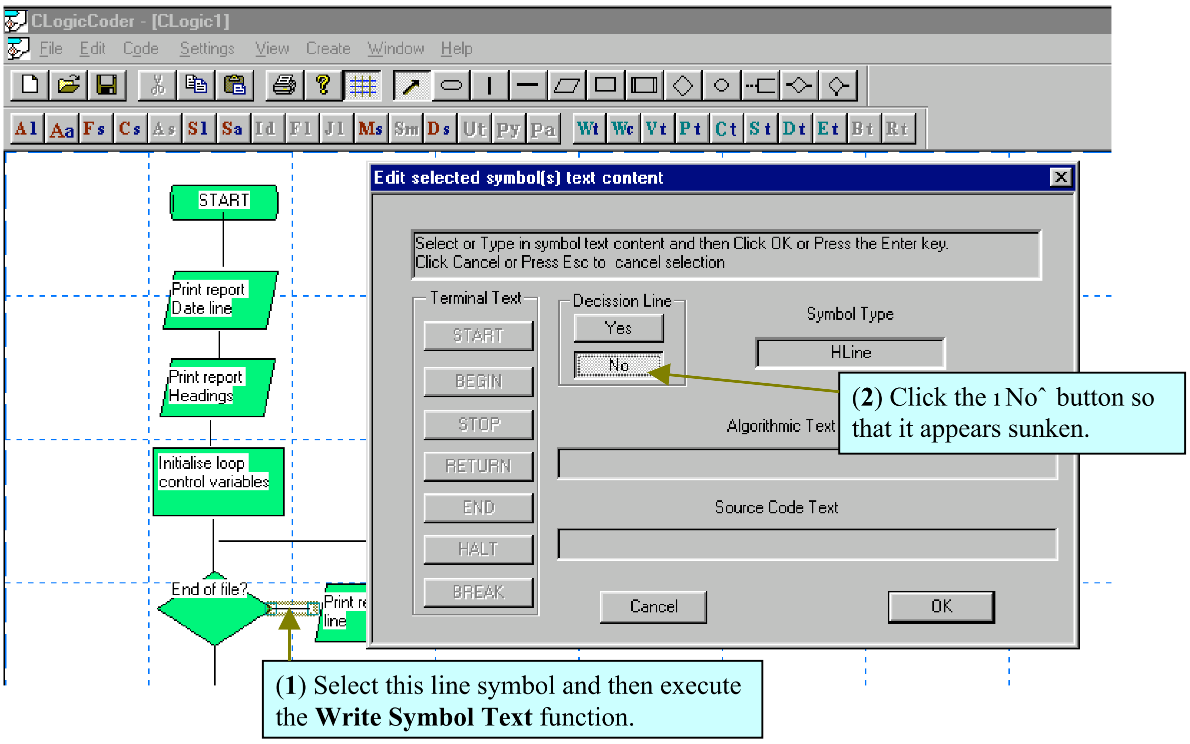

CLogicCoder - [CLogic1]
File Edit Code Settings View Create Window Help
START
Print report Date line
Print report Headings
Initialise loop control variables
End of file?
Print re line
Edit selected symbol(s) text content
Select or Type in symbol text content and then Click OK or Press the Enter key.
Click Cancel or Press Esc to cancel selection
Terminal Text
START
BEGIN
STOP
RETURN
END
HALT
BREAK
Decission Line
Yes
No
Symbol Type
HLine
Algorithmic Text
Source Code Text
Cancel
OK
(2) Click the ‘No’ button so that it appears sunken.
(1) Select this line symbol and then execute the Write Symbol Text function.

Figure 17
You can only enter ı Yesˆ or ı Noˆ text in a line symbol that emanates from a Loop or Case Decision symbol. LogicCoder ensures that text in line symbols that emanates from a Decision symbol is always complement of each other.

SAVING THE FLOWCHART

STEP 1:
Use the mouse left button to click the **File** menu item on the main menu bar so that the **File** pull-down sub-menu appears.

STEP 2:
(1) Use the mouse left button to click the **Save** menu item in the **File** pull-down sub-menu so that the **Save** function executes.

(2) Save the file with the following name: Tutor00.flw.
Make note of the location where this file is saved. You will look for the generated source code in this same location.

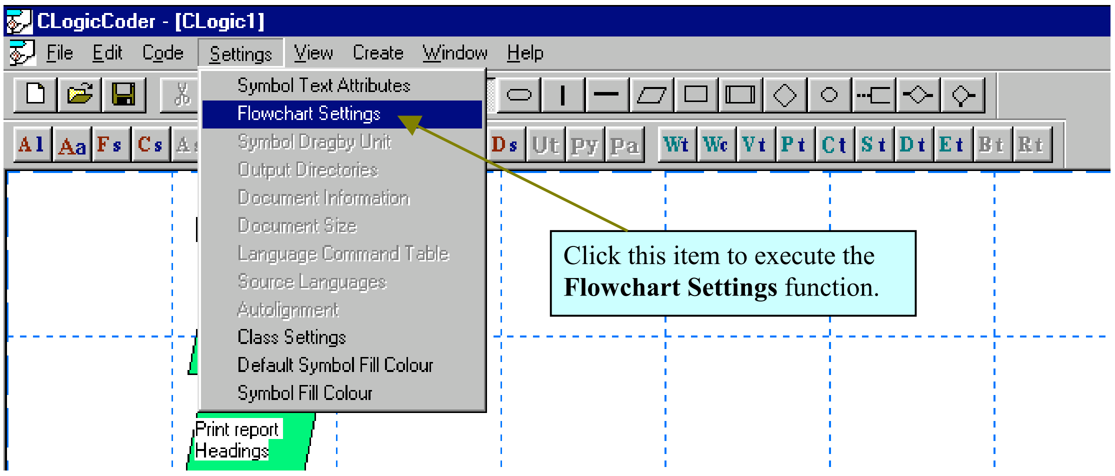
When you execute the **Save** function, LogicCoder saves both the flowchart and its text content together. However, you can also save the source code text separate from the flowchart and you can also load specific sequence of source code text into a flowchart.

If you save the source code text of a flowchart with the intention to reuse it at a future date, then you should pay careful attention to any changes that you do to the outlay of the control logic in the flowchart. Edit operations on a flowchart such as a symbol deletion affects the sequence of registered symbols in a flowchart. If the registered sequence does not match that in the saved source code, then the source once text will not load properly into the flowchart. More advance version of LogicCoder provides functions that help you to manage these kinds of situations.

CREATING THE ANSI C SOURCE PROGRAM

In this section I show you how to create and insert data structures into a C++ program with use of the struct statement. I also show you how to create the required C/C++ source program from the flowchart.

STEP 1:
Use the mouse left button to click the **Settings** item on the main menu bar so that the Settings pull-down sub-menu appear as illustrated in figure 18.



Click this item to execute the **Flowchart Settings** function.

Figure 18
Use the Settings menu item to select and set attributes when drawing a flowchart or when writing a source program from a flowchart.

STEP 2:
Click the **Flowchart Settings** menu item in the pull-down window so that the dialog window as that in figure 19 appears.

STEP 3:
Enter the following text in the text editor window labelled: **Preprocess Directives**

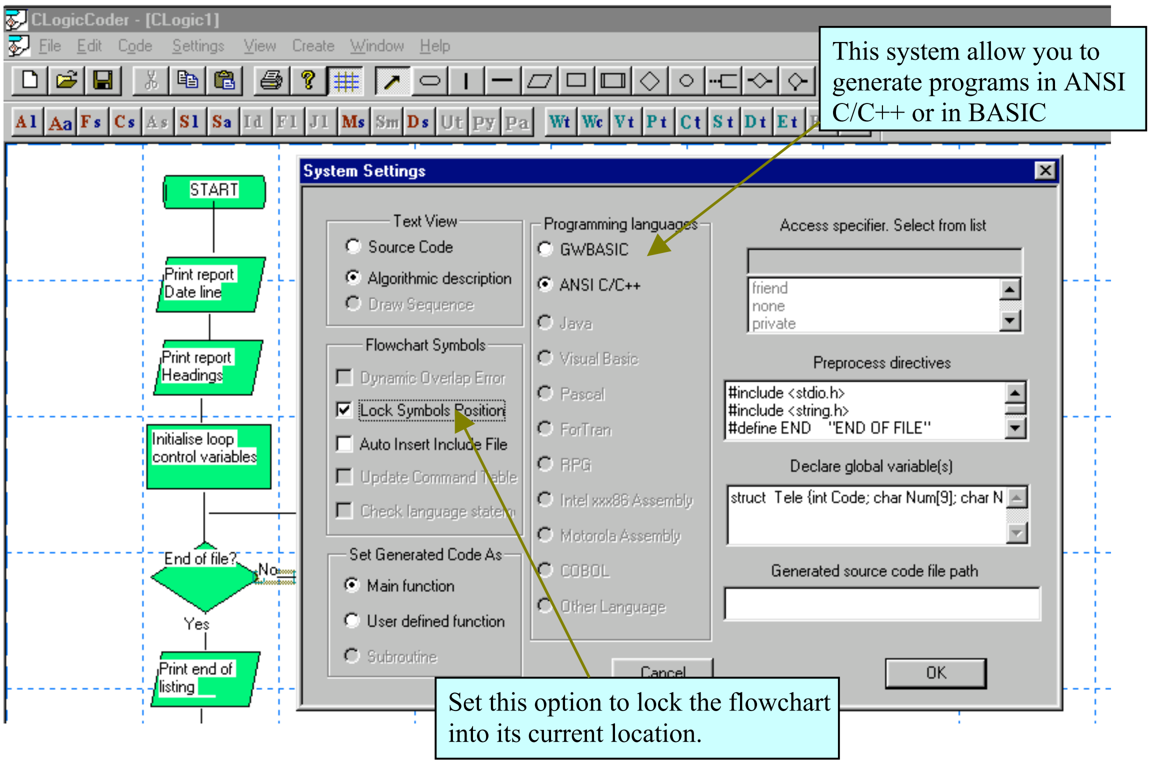
```
#include <stdio.h>
#include <string.h>
#define END    "END OF FILE"
```

More advance versions of LogicCoder has functions that tells it to insert known include file headers in a generated C/C++ source program base on the use of command statements in the source code.

STEP 4:
Enter the following text in the text editor window labelled: **Declare Global Variable**(s)

```
struct Tele {int Code; char Num[9]; char Name[12];};
struct Tele Telefile[ ] =
```

STEP 5:
Click the **Lock Symbols Positions** check box so that it appears as illustrated in figure 19.


CLogicCoder - [CLogic1]
File Edit Code Settings View Create Window Help
Al Aa Fs Cs As Sl Sa Id Fl Jl Ms Sm Ds Ut Py Pa Wt Wc Vt Pt Ct St Dt Et
START
Print report Date line
Print report Headings
Initialise loop control variables
End of file?
No
Yes
Print end of listing
System Settings
Cancel
OK

This system allow you to generate programs in ANSI C/C++ or in BASIC

Text View
- ( ) Source Code
- (●) Algorithmic description
- ( ) Draw Sequence

Flowchart Symbols
- ☐ Dynamic Overlap Error
- ☑ Lock Symbols Position
- ☐ Auto Insert Include File
- ☐ Update Command Table
- ☐ Check language statem

Set Generated Code As
- (●) Main function
- ( ) User defined function
- ( ) Subroutine

Programming languages
- ( ) GWBASIC
- (●) ANSI C/C++
- ( ) Java
- ( ) Visual Basic
- ( ) Pascal
- ( ) ForTran
- ( ) RPG
- ( ) Intel xxx86 Assembly
- ( ) Motorola Assembly
- ( ) COBOL
- ( ) Other Language

Access specifier. Select from list
- friend
- none
- private

Preprocess directives
```
#include <stdio.h>
#include <string.h>
#define END    "END OF FILE"
```

Declare global variable(s)
```
struct  Tele {int Code; char Num[9]; char N
```

Generated source code file path

Set this option to lock the flowchart into its current location.

Figure 19
Notice that the default setting of the system is to generate programs in ANSI C/C++. You can switch this setting by clicking any of the available source language in the system. There are other functions and settings options available in more advance versions of LogicCoder.

STEP 6:
Click the OK button when you are finish step 6

Try dragging any symbol in the flowchart once the dialog window has disappeared from the document window. The flowchart symbols should not move whenever you attempt to drag them with the mouse.

In the next step you will execute the function that writes the required C/C++ source program.

STEP 7:
Click the **Code** item on the main menu bar so that the **Code** pull-down sub-menu appears as illustrated in figure 20.

STEP 8:
Click the **Write Program** menu item so that LogicCoder generates the required C/C++ source code.

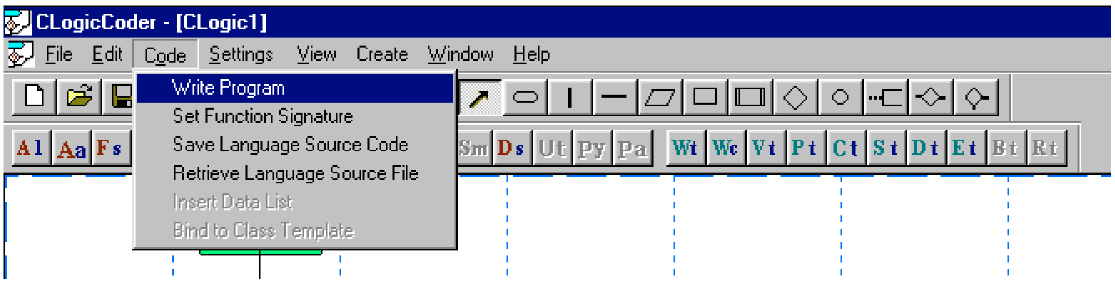

CLogicCoder - [CLogic1]
File Edit Code Settings View Create Window Help
Write Program
Set Function Signature
Save Language Source Code
Retrieve Language Source File
Insert Data List
Bind to Class Template

Figure 20
Use the Write Program function to tell LogicCoder to generate the required source program in the set source language.

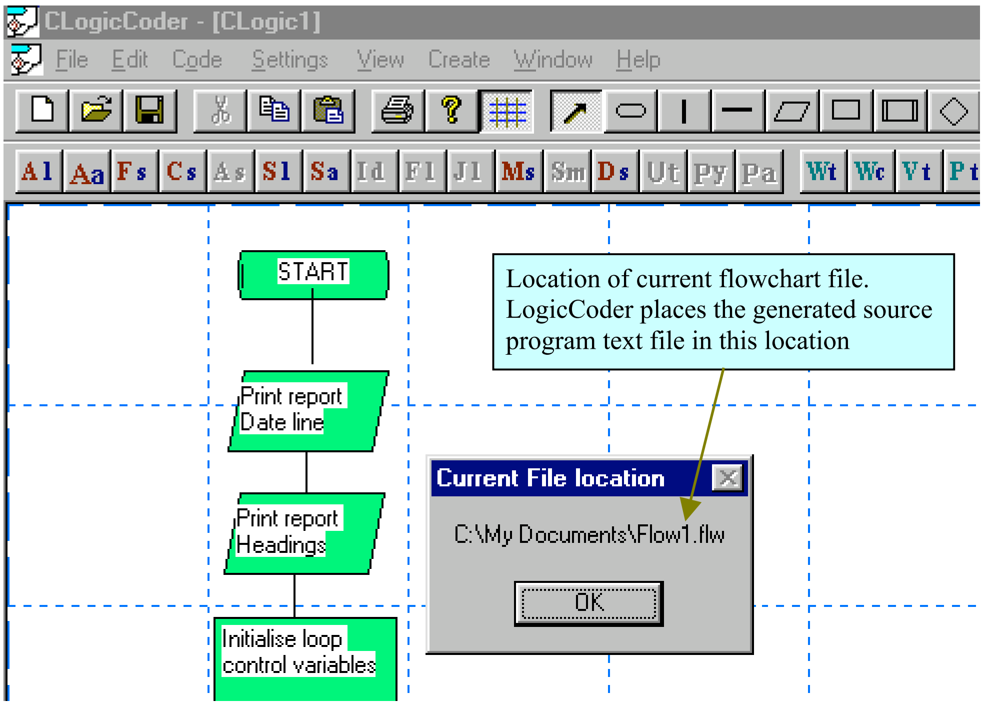

CLogicCoder - [CLogic1]
File Edit Code Settings View Create Window Help
START
Print report Date line
Print report Headings
Initialise loop control variables
Location of current flowchart file. LogicCoder places the generated source program text file in this location
Current File location
C:\My Documents\Flow1.flw
OK

Figure 21
LogicCoder display message boxes to show the file location and that the flowchart is connected OK before it proceeds to generate the required source program. LogicCoder places the generated source program in the same location as the flowchart file.

LogicCoder presents you with a series of message boxes that indicates where the program flowchart is stored on your system, diagnostic message that indicates connectivity of the flowchart and a message to indicate that the system is generating the required program. Make note of the location where the generated program is stored. When a source program is generated, it is stored in a text file with the name default to that of the flowchart except that the extension is that of the standard extension for the source language your system is set to.

STEP 9:
Use NotePad.exe to open the file that has the generated source program.

Your source code should look similar to that illustrated in figure 22.

```
#include <stdio.h>
#include <string.h>
#define END    "END OF FILE"

struct  Tele {int Code; char Num[9]; char Name[12];};
struct Tele Telefile[ ] =

void main(void)
       {
        int r;

        printf("\n \n \t  Current date:   %20s", __DATE__);
        printf("\n \n \t \t TELEPHONE LISTING \n ");
       for(r = 0; strcmp(Telefile[r].Name, END) != 0;  r++)
          printf("\n\t%i \t%-s \t%s", Telefile[r].Code, Telefile[r].Num, Telefile[r].Name);
        printf("\n \n \t END  OF TELEPHONE LISTING:");
        }
```

Figure 22
The generated source program should look similar to this. Notice that LogicCoder does source code indentation layout to highlight the control logic in the generated source program.

SYMBOL HIGHLIGHTING AND SUB-STRUCTURE ABSTRACTION

LogicCoder allow you to set the symbol display colour in either of two ways ways.
(1) You can select one or more symbols at a time and use the **Symbol Fill Colour** function to select the colour you want or
(2) You can set the default fill colour as you draw flowchart symbols in a document. In our tutorial program, the default symbol fill colour is the value selected by the system at start-up. I.e. Green. You can change this value by executing the **Default Symbol Fill Colour** function in the **Settings** pull-down menu.

It is some times useful to highlight a group or sequence of symbols in a flowchart by setting them to a common colour. This may be for pedagogic or documentation reason for example. In this section of the tutorial we want to highlight sections of the flowchart we intend to abstract in order to create a simply abstract view of our system.

An abstract view of a system simplifies the overall understanding of the control logic of the system without loss of essential information. More advance version of LogicCoder allows you to select sections within a flowchart and then abstract them to the Predefine Abstract symbol. The **Predefine Abstract** symbol is represented as explained in figure 23. LogicCoder also allow you to expand an abstracted view so as to replace its original detail.

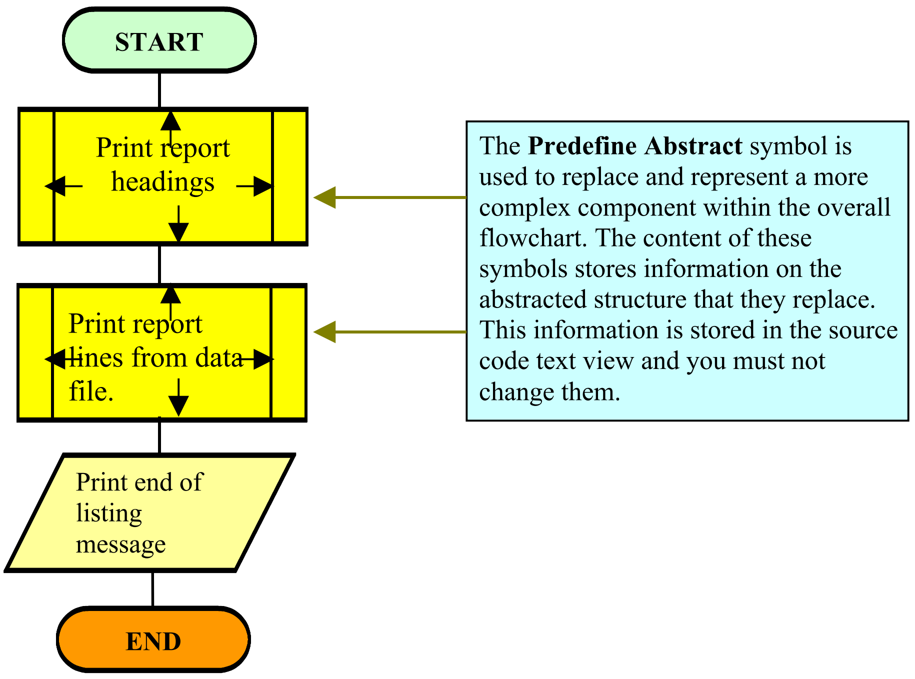
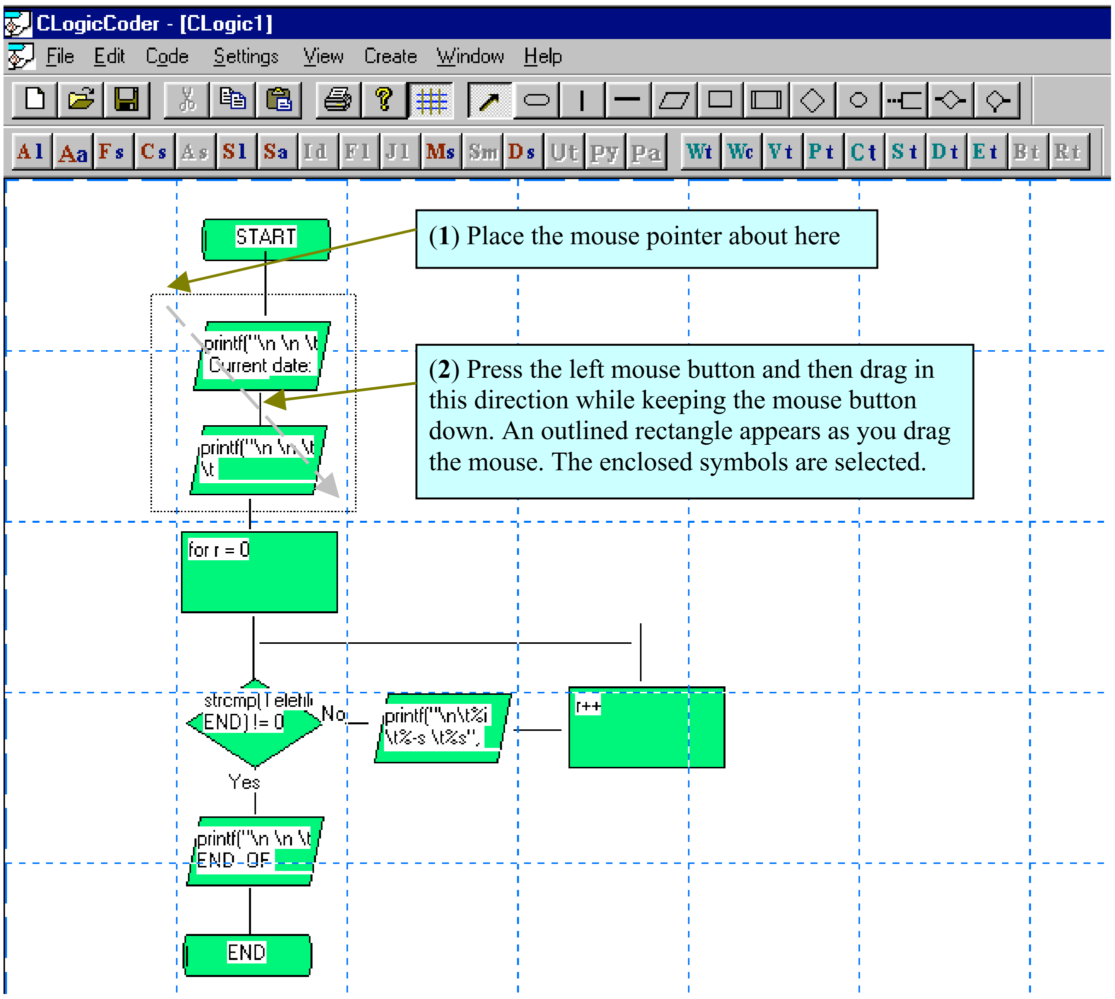

START
Print report headings
Print report lines from data file.
Print end of listing message
END
The **Predefine Abstract** symbol is used to replace and represent a more complex component within the overall flowchart. The content of these symbols stores information on the abstracted structure that they replace. This information is stored in the source code text view and you must not change them.

Figure 23
An abstracted view of the flowchart illustrated in figure 1, to be created with LogicCoder. Once you have created this view of your program, you can then store (save) your flowchart document as a different file of this abstracted view.

In the next few exercises, you will learn how to select groups of symbols to perform common operations on them.

STEP 1:
Place the mouse pointer to the left and just below the first Terminal symbol in the user document window and then press the left mouse button. Drag the mouse while keeping the mouse button down as indicated in figure 24.



(**1**) Place the mouse pointer about here

(**2**) Press the left mouse button and then drag in this direction while keeping the mouse button down. An outlined rectangle appears as you drag the mouse. The enclosed symbols are selected.

Figure 24
You can select groups of symbols in a flowchart by liaison with the mouse or you can select them individually by pressing the Shift key and then click each symbol you want.

STEP 2:
Use the mouse left button to click on the **Settings** menu item on the main menu bar so that the Settings pull-down sub-menu appear and then click on the **Symbol Fill Colour** sub-menu item.

LogicCoder present you with the standard colour selection dialog window as illustrated in figure 25.

STEP 3:
Select the Blue colour item in the colour dialog window and then click the OK button or press the Enter key.

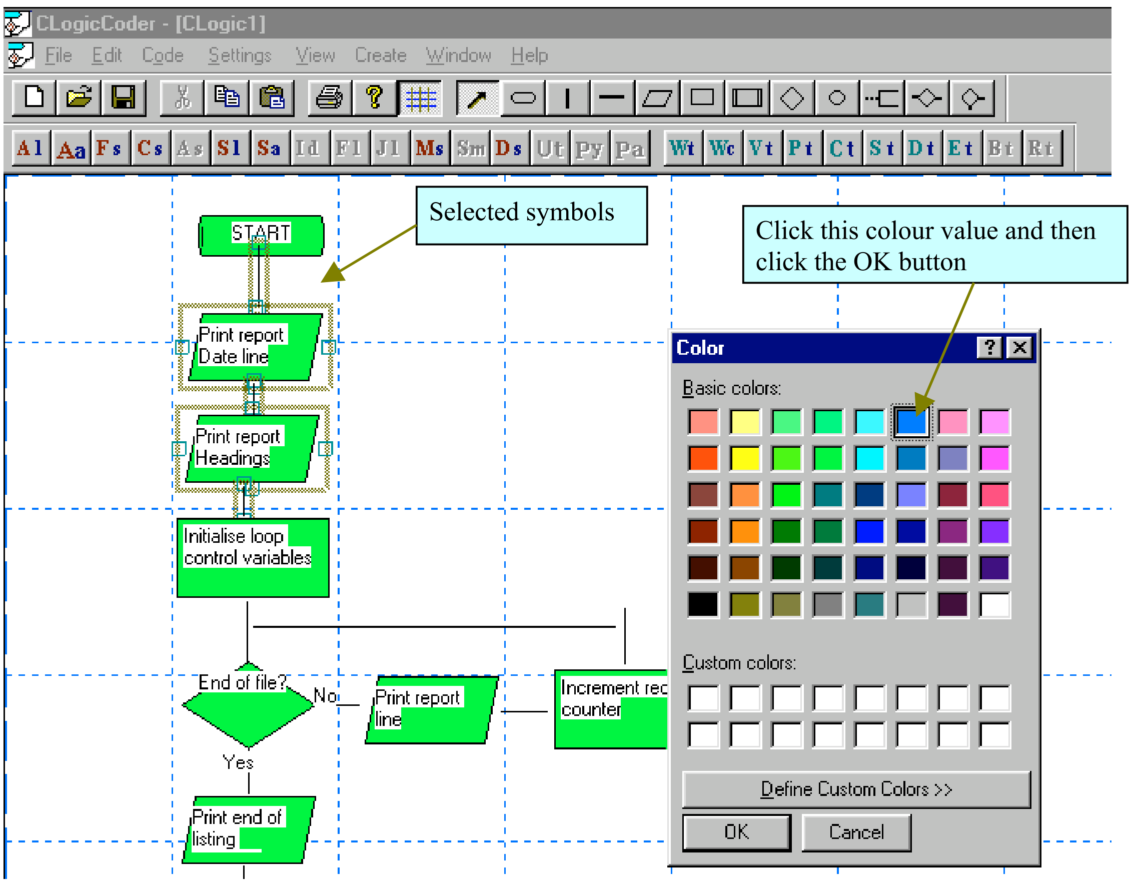


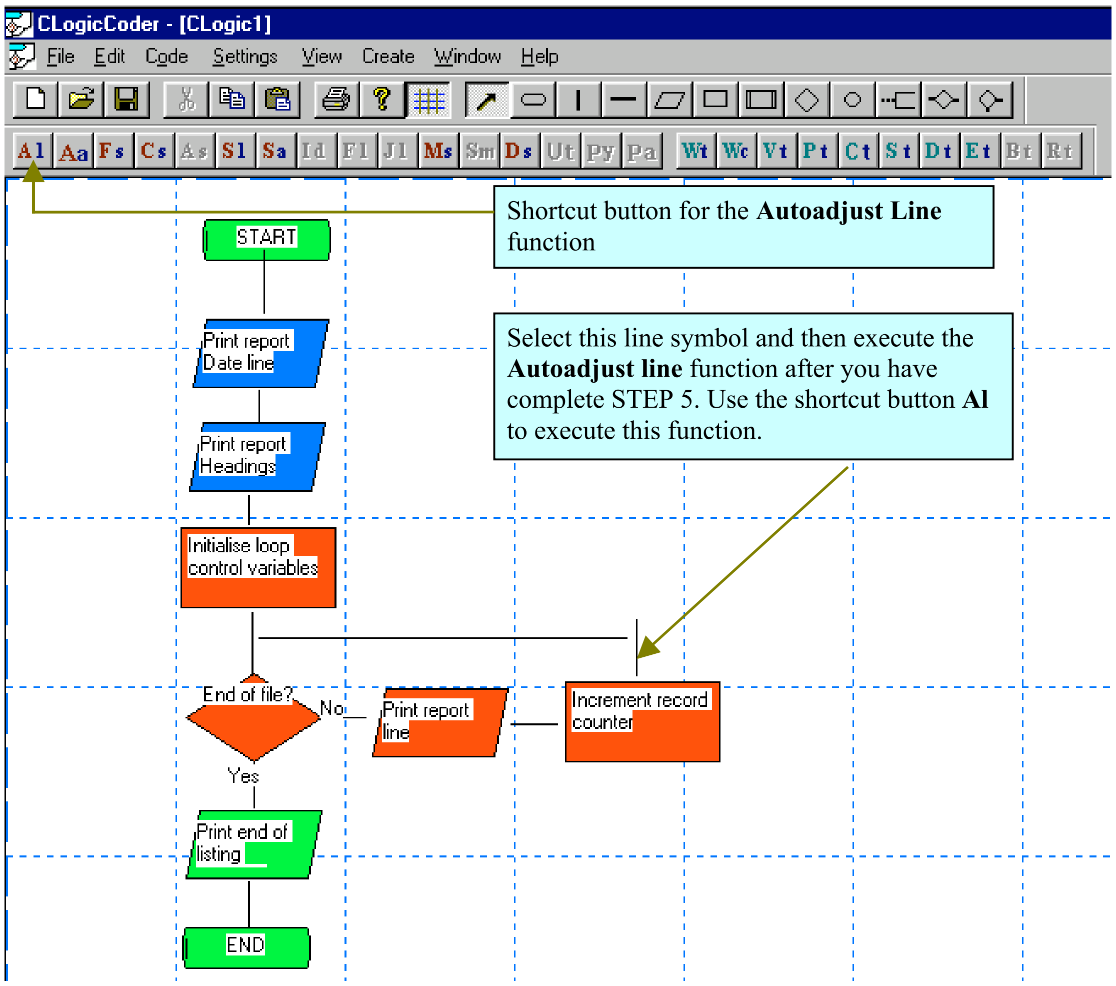
Figure 25
You select the set of symbols and then select the colour you want them to have. You also use similar operation to set the default fill colour when drawing a flowchart document.

Once you press the Enter key or click the OK button, the fill colour of the selected symbols should change to blue ˚ the selected colour.

STEP 4:
Use the same process as in step 1 to select the first process symbol, the decision symbol and the symbols that constitute the body of the loop in the flowchart and then execute the **Symbol Fill Colour** function once more as in step 3. This time you will select a different fill colour.

STEP 5:
Select the colour red in the colour dialog window and then click the OK button or press the Enter key once.

Your document should appear similar to that in figure 25.


CLogicCoder - [CLogic1]
File Edit Code Settings View Create Window Help
Al Aa Fs Cs As Sl Sa Id Fl Jl Ms Sm Ds Ut Py Pa Wt Wc Vt Pt Ct St Dt Et Bt Rt
START
Print report Date line
Print report Headings
Initialise loop control variables
End of file?
No
Yes
Print report line
Increment record counter
Print end of listing
END

Shortcut button for the **Autoadjust Line** function

Select this line symbol and then execute the **Autoadjust line** function after you have complete STEP 5. Use the shortcut button **Al** to execute this function.

Figure 25
LogicCoder allow you to use colour scheme to highlight selected groups of symbols in a flowchart document.

STEP 6:
Select the last Input/Output symbol and then change its colour so that it appears as illustrated in figure 26.

STEP 7:
Select the last Terminal symbol and then execute the Write Symbol Text function so that its text appears as illustrated in figure 26.

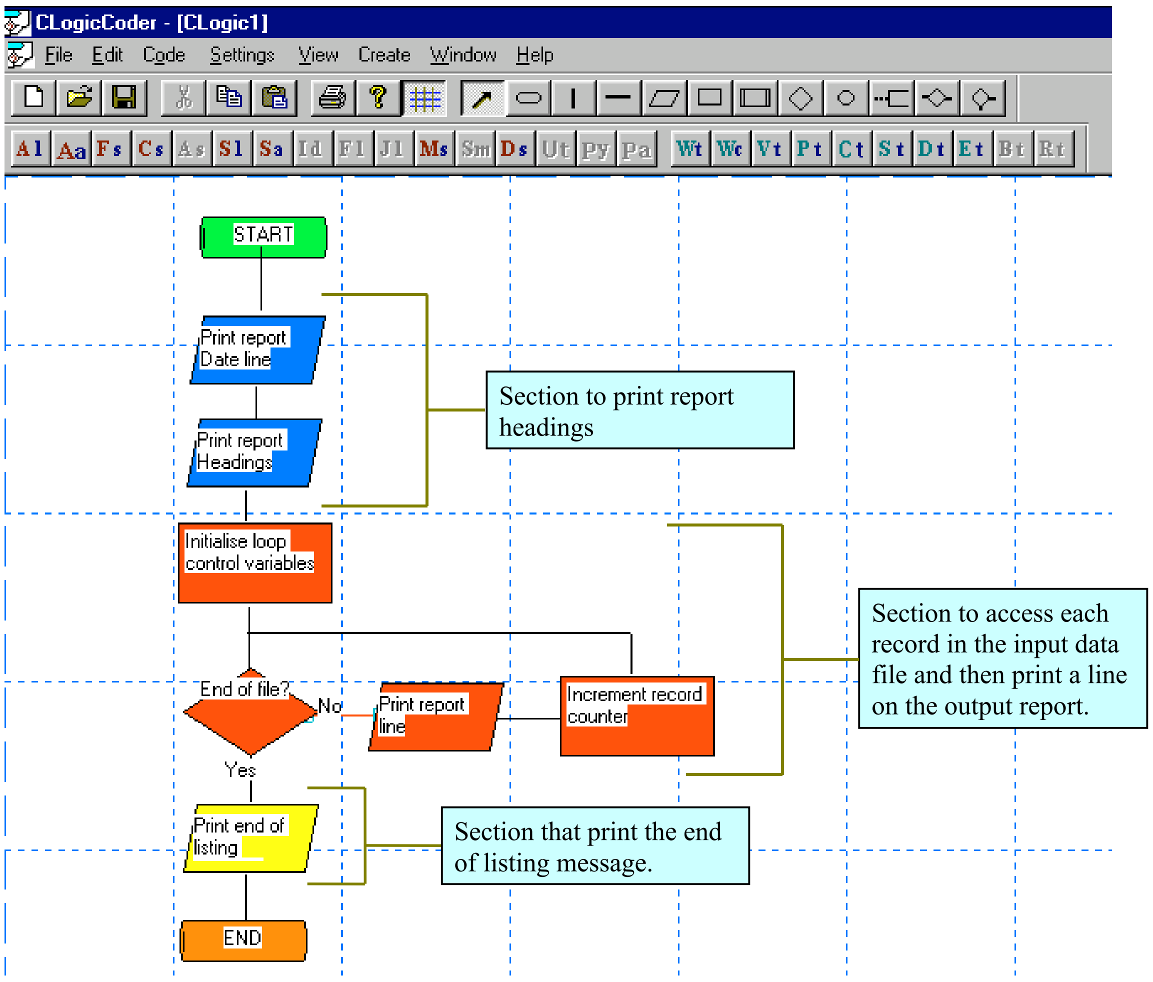

CLogicCoder - [CLogic1]
File Edit Code Settings View Create Window Help
Al Aa Fs Cs As Sl Sa Id Fl Jl Ms Sm Ds Ut Py Pa Wt Wc Vt Pt Ct St Dt Et Bt Rt
START
Print report Date line
Print report Headings
Initialise loop control variables
End of file?
No
Yes
Print report line
Increment record counter
Print end of listing
END
Section to print report headings
Section to access each record in the input data file and then print a line on the output report.
Section that print the end of listing message.

Figure 26
You can use a colour scheme to highlight section of a flowchart that does specific task in a program. You can then create a separate abstract view as illustrated in figure 23.

LogicCoder supports the incremental approach to the development of complex software systems both by the (1) bottom-up and the (2) top-down design approach. (1) The bottom-up approach allow you to design a complex system by adding more functional components to an already existing system until a refined result that meets a specific requirement is reach. You use LogicCoder **Insert Structure** or **Insert Symbols** to insert other functions or flowchart sections into an existing flowchart. You can also change selected symbols in a flowchart.

(2) The top-down approach allow you to start with an abstract definition of a program problem, you identify specific components within the abstraction, repeat the process on these components until their description maps well onto statements within the domain in which the problem solution is to be implemented. The above explanation is an illustration of how you can do this with LogicCoder.

In the next section that follows, you will learn how to

- Save the source code text view of a flowchart
- Clear the source code text of a flowchart
- Switch the source language of the flowchart
- Enter command statement in the selected source language
- Change symbol types in a flowchart
- Set a select symbol to the ignore state when writing a source program
- Generate a source program in the set source language

SAVING THE SOURCE CODE TEXT FILE

STEP 1:
Click the **Code** item on the main menu bar so that the Code pull-down sub-menu appears as illustrated in figure 27.

The Code sub-menu consist of a set of functions that allow you to create source code files from a flowchart and to save and retrieve source code text view files for a given flowchart. You can also do chunk modification to a source code file by insertion of other code sequence such as data list or multiple comment lines.

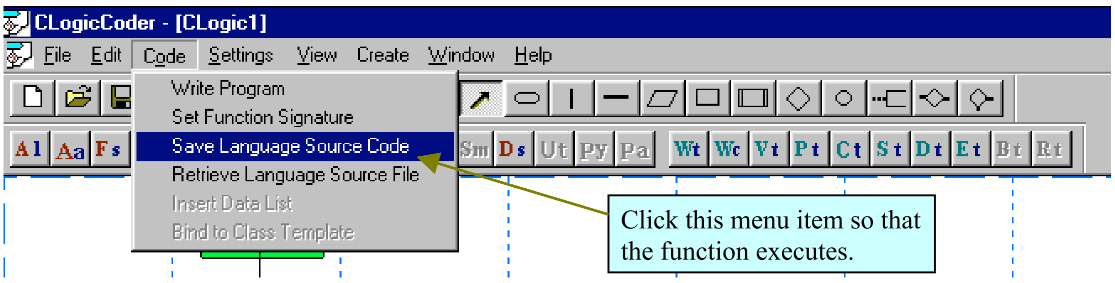

CLogicCoder - [CLogic1]
File Edit Code Settings View Create Window Help
Write Program
Set Function Signature
Save Language Source Code
Retrieve Language Source File
Insert Data List
Bind to Class Template
Click this menu item so that the function executes.

Figure 27
The Save Language Source Code function saves the source code text content of each symbol in the flowchart as a text file. The text that belongs to each symbol is labelled in the sequence of these symbols in the flowchart.

When you save a source code file, LogicCoder keeps track of the individual symbol to which each text string belongs. LogicCoder also store information that determines which source language the source code text file belongs to. LogicCoder do not allow you to load a source code text file into a flowchart that does not belong to the source language for which your system is currently set.

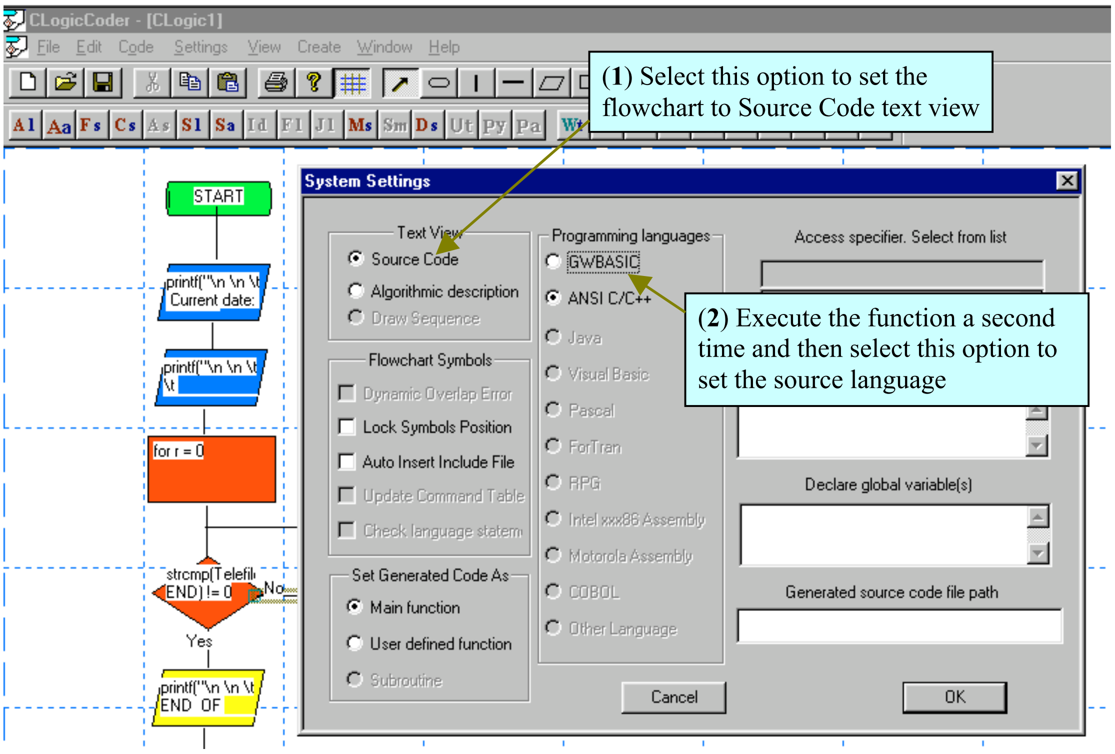
STEP 2:
Use the mouse left button to click the **Language Source Code** item in the **Code** pull-down sub-menu.

At this point, LogicCoder presents you with the standard Windows File Save dialog window. Save the source code text file with file name **Flow1.cxt**. You do not have to place the extension on the file name when saving. LogicCoder uses the standard DOS extension CXT for source code text view file names.

You will now clear the flowchart of the C/C++ source code text view and then reset the system to generate a BASIC program.

STEP 3:
Use the **Settings** function to switch the current text view of the flowchart to the **Source Code text** view and then click the OK button or press the Enter key.

At this point, LogicCoder display the language command statement in each symbol of the flowchart.


CLogicCoder - [CLogic1]
File Edit Code Settings View Create Window Help
START
Yes
No
System Settings
Text View
Programming languages
Access specifier. Select from list
Flowchart Symbols
Declare global variable(s)
Set Generated Code As
Generated source code file path
Cancel
OK

(**1**) Select this option to set the flowchart to Source Code text view

(**2**) Execute the function a second time and then select this option to set the source language

Text View: [✓ Source Code] Algorithmic description, Draw Sequence

Flowchart Symbols: Dynamic Overlap Error, Lock Symbols Position, Auto Insert Include File, Update Command Table, Check language statem

Set Generated Code As: [✓ Main function] User defined function, Subroutine

Programming languages: GWBASIC [✓ ANSI C/C++] Java, Visual Basic, Pascal, ForTran, RPG, Intel xxx86 Assembly, Motorola Assembly, COBOL, Other Language

```
printf("\n \n \t
Current date:
printf("\n \n \t
\t
for r = 0
strcmp(Telefil
END) != 0
printf("\n \n \t
END  OF
```

Figure 28
LogicCoder provides functions that allow you to switch between source code text and Algorithmic text views in a flowchart.

You can use the **Erase Text View** function to remove all the text content for a selected view in a flowchart. However, you must always switch the flowchart text view to that that you intend to erase before you executes the **Erase Text view** function.

STEP 4:
Execute the **Erase Text view** function in the **Code** pull-down sub-menu so that all source code text content disappears form the flowchart.

LogicCoder prompts you to determine that you are certain that you really want to erase the selected text view whenever you execute the **Erase Text View** function. You can then cancel this operation if you change your mind.

It is a good practice to always save the source code text view before you execute the **Erase Text view** function whenever the view is set to Source Code.

This version of LogicCoder does not allow you to save the Algorithmic Text view in a flowchart. Advance versions of LogicCoder allow you to have multiple Algorithmic and source code text views for a single flowchart and to save and retrieve these text views. Therefore, you can have a descriptive flowchart written in more than one natural language but implemented in a single source code language.

SETTING THE SYSTEM TO GENERATE A PROGRAM IN BASIC

In this section, you will do changes to the flowchart layout so that it will generate a BASIC source program for the problem solution. In BASIC, you use the DATA statement to create a list of data values to be used by a program. As with C/C++, you can also create an array of data locations and then initialise the array with values. The pattern of initialising values must match the data structure pattern. However, a C/C++ compiler is more efficient at generating machine code for processing array of data values in terms of memory usage, convenience of implementation and time usage.

We have a different view of the way the system obtains its data values in the BASIC implementation of our program problem solution. This is because the data values used in the program are actually test data for the program. Actually, the final program is suppose to get its data values by reading data from an external device such as a magnetic storage medium.

STEP 1: Change the source language.
Select the **Settings** menu item on the main menu bar by clicking with the mouse left button or by highlighting and then press the Enter key. Then execute the **Flowchart Settings** function.

At this stage, LogicCoder presents you with the **Flowchart Settings** dialog window as illustrated in figure 28.

STEP 2:
Select the GWBASIC option in the **Programming Language** option group as illustrated in figure 28 and then click the OK button.

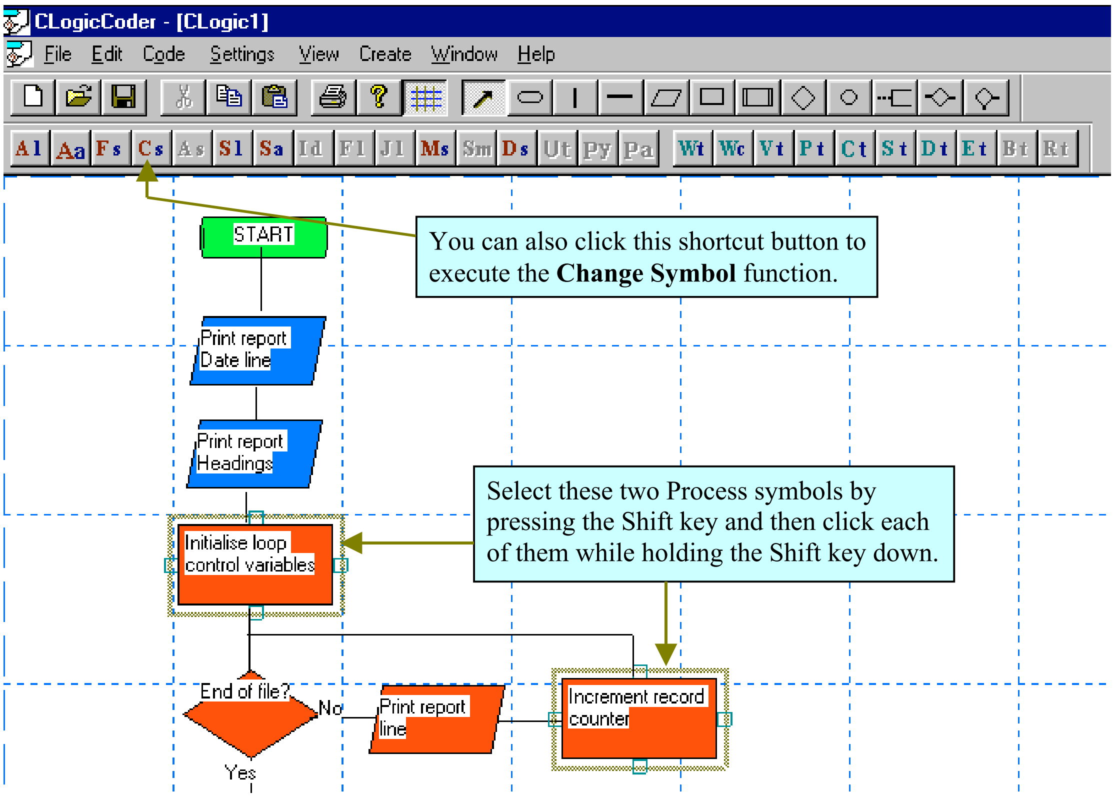
The source language command statement in each flowchart symbol except the Terminal symbols is currently written in ANSI C/C++. You need to remove the source code text of each flowchart symbol. More advance versions of LogicCoder provides you with an option dialog window that allow you to change programming language, save source code text view, retrieve source code or delete source code text view all at one go.

STEP 3: Edit the flowchart
Press and keep down the Shift key and then use the mouse to click on the two Process symbols in the flowchart so that they are selected.

At this point your document window should look similar to that in figure 29.

STEP 4:
Select the **Edit** menu item on the main menu bar so that the Edit pull-down sub-menu appears and the select the **Flowchart** item in the Edit pull-down sub-menu.

Recall that LogicCoder allows you to do either of two kinds of editing on a flowchart document. You can edit the text content of a flowchart or you can edit the control logic illustrated in a flowchart. LogicCoder is the only system of its kind that allows you to separate and edit the implementation control logic of a system from the source code specific text. Therefore, you are presented with a vastly improved opportunity for increase productivity and useful creativity as a programmer.


CLogicCoder - [CLogic1]
File Edit Code Settings View Create Window Help
Al Aa Fs Cs As Sl Sa Id Fl Jl Ms Sm Ds Ut Py Pa Wt Wc Vt Pt Ct St Dt Et Bt Rt
START
Print report Date line
Print report Headings
Initialise loop control variables
End of file?
No
Yes
Print report line
Increment record counter

You can also click this shortcut button to execute the **Change Symbol** function.

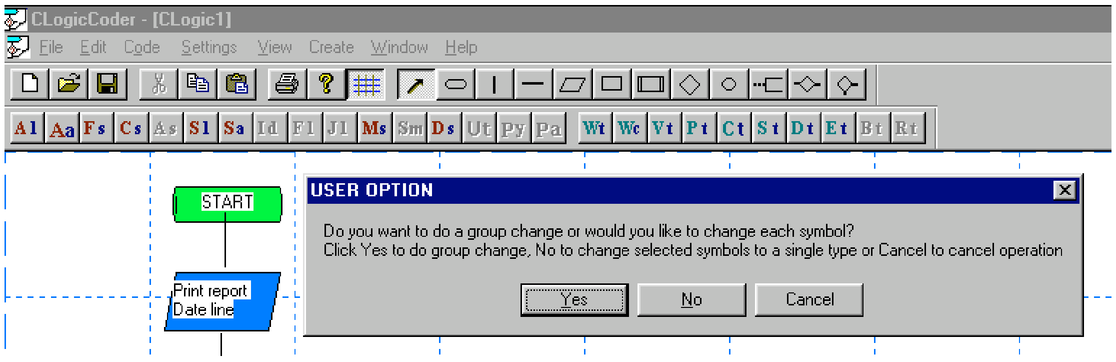
Select these two Process symbols by pressing the Shift key and then click each of them while holding the Shift key down.

Figure 29
You can use the mouse along with the shift key to select individual symbols in a flowchart at random.

STEP 5:
Select the **Change Symbol** item in the **Flowchart** pull-down sub-menu by clicking with the mouse left button or you can highlight it with the arrow key(s) and then press the Enter key once.

At this point, LogicCoder present you with a dialog window as illustrated in figure 30. This dialog window allows you to select the way in which you would like to implement the symbol change. LogicCoder presents this dialog window whenever you select 2 or more symbols when executing this function. You can change the symbols in a group all at once or you can change them individually in step sequence.

Note that whenever you do a symbol type change, the text view contents remain the same. You can change the text content by execution of the **Write Symbol text** function.



Do you want to do a group change or would you like to change each symbol?
Click Yes to do group change, No to change selected symbols to a single type or Cancel to cancel operation

Figure 30
This dialog option window allow you to choose the way in which you would like to implement symbol type changes to a group of selected symbols in a flowchart.

STEP 6:
Click the **No** button in the dialog window.

LogicCoder presents you with a second dialog window that allows you to select the symbol you want to change to.

STEP 7:
Select the Input/Output symbol by clicking the **In/Output** button and then click the **OK** button in the dialog window.

LogicCoder changes the 2 Process symbols to Input/Output symbols.

STEP 8:
Ensure that the two Input/Output symbols remain selected and then execute the **Write Common text** function in the Edit ➔ Symbol Text pull-down sub-menu.

STEP 9:
Place the following text in each of the text views listed below.

Algorithmic text
Read a record

Source Code text
READ N$, T$, A

STEP 10:
Place the following text in each text views of the first and the second Input/Output symbols.

First Input/Output symbol

Algorithmic text
Print report Date line

Source Code text
```
PRINT TAB(20) ıCURRENT DATE:  ˆDATE
```

Second Input/Output symbol
Algorithmic text
Print report Headings

Source Code text
```
PRINT ıˆ: PRINT TAB(20) ıTelephone Listingˆ
```

Note that we have used two PRINT command statements on a single line. When you enter source code command statements in a flowchart, bear in mind that each command statement corresponds to a single line in the generated source code.

STEP 11:
Place the following text in the first Input/Output symbol within the body of the loop control logic.

Algorithmic text
Print report line

Source Code text
```
PRINT TAB(10) N$, A, T$
```

STEP 12:
Place the following text views in the last Input/Output symbol in the flowchart.

Algorithmic text
Print end of listing message

Source Code text
```
PRINT ıˆ: PRINT ıEND OF TELEPHONE LISTINGˆ
```

INSERTING A DATA LIST IN A BASIC PROGRAM

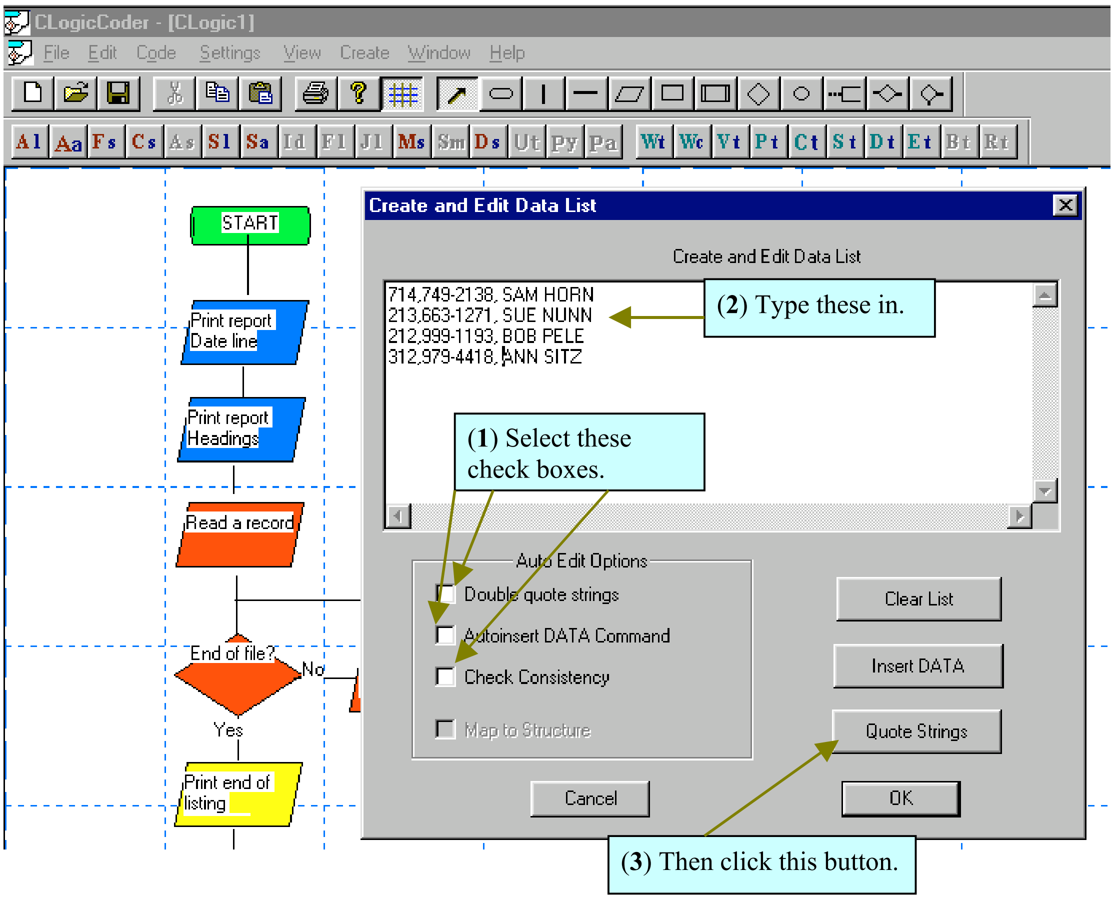
LogicCoder also allow you to insert a data list at the end of a BASIC program. You can select that LogicCoder does consistency checks as it inserts this list and that it delimits string data in the list. LogicCoder will work out the required line number for the list sequence as it does for the generated source code.

STEP 13:
Select and execute the **Insert Data List** function in the **Code** pull-down sub-menu.

At this stage, LogicCoder presents you with the dialog window as illustrated in figure 31.

Use the illustration in figure 31 as a guide for following the next few instructions. Always do the instructions given in the text before any additional instruction given in the illustration.

You must not execute this function before you create a BASIC program from a flowchart. LogicCoder can not insert a data list into an empty program file.



```
714,749-2138, SAM HORN
213,663-1271, SUE NUNN
212,999-1193, BOB PELE
312,979-4418, ANN SITZ
```

Figure 31
You can use the Insert Data List function to insert a list of data values at the end of a BASIC code. You can set options that get LogicCoder to do things like consistency of the data list, quoting of string data values or clear the data you have already entered.

More advance version of LogicCoder allows you to do insertion of other data types such as comment or external code sequence. In this case, the insertion point is always at a selected point that must be a line symbol in the flowchart.

STEP 15:
Enter the following text values in figure 32 into the data list edit box.
Click the **Double Quote** strings check box so that it is selected
Click the **Autoinsert DATA** command check box so that it is selected
Click Quote Strings command button.
Click Insert DATA command button.

Click the OK button

```
714, 749-2138,SAM HORN
213, 663-1271,SUE NUNN
212, 999-1193,BOB PELE
312, 979-4418,ANN SITZ
999,999-0000, END OF FILE
```

Figure 32
The set of test data values to be used in the BASIC program that implements the program problem specification. Each line in the list represents a record. Each record consists of a numeric and then 2 string fields. Most BASIC interpreters require that you delimit string data values within double quotes.

STEP 14:
Save the flowchart by clicking the Save shortcut icon on the edit toolbar.

GENERATING A SOURCE CODE IN BASIC

Use the **Write Program** function in the **Code** sub-menu to tell LogicCoder to generate the BASIC program from the control logic in the flowchart.

Use NotePad to open the file that contains the generated source program and examine it. You can edit and use the source code as a normal BASIC program.

Yours source code should look similar to that illustrated in figure

That is the end of this tutorial for the time being. Remember that you can have a free download of this tutorial in pdf format by clicking on the following link.

RELOADING THE ANSI C/C++ SOURCE CODE

1 Reset the text view to source code and then delete the text view
2 Reset the system to ANSI C/C++
3 Reload the ANSI C/C++ source code text
4 Retype the source code text content of the Decision symbol to the following
5 Select the Input/Output symbols above the Decision and the second Input/Output in the body of the loop and then execute the Set/unset symbol state to Ignore function.