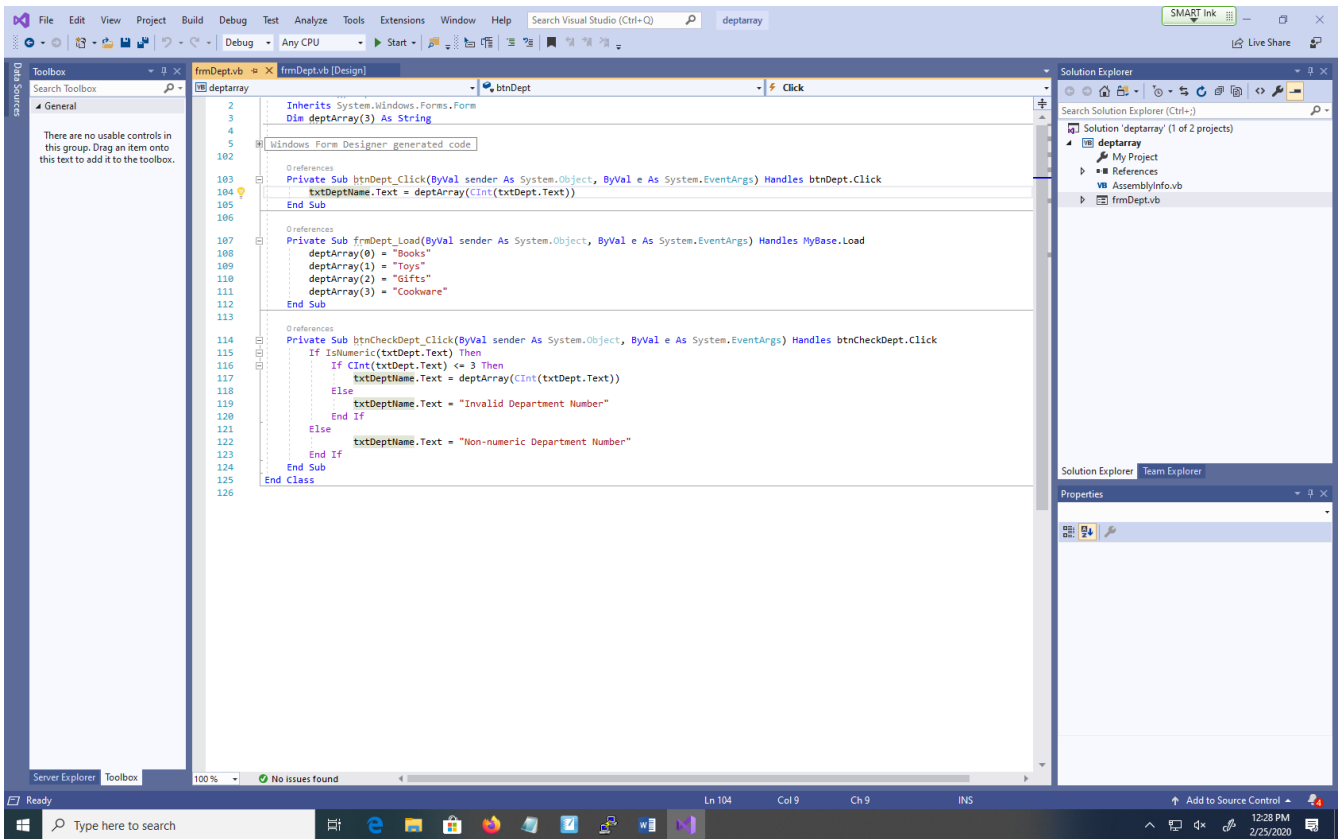


	Beginning examples of code continued (html version)
Conditions	If programs (zipped) If examples (word version) If examples (html version) If examples with files (word version) If examples with files (html version) More if examples (word version) More if examples (html version)
Loops, combo boxes etc.	Introduction to loops examples (zipped) Introduction to loops examples (word version) Introduction to loops examples (html version) Loops continued examples and number guess (zipped) Guess number (word version) Guess number (html version) Loops continued (word version) Loops continued (html version) Loops and combo (zipped) Loops and combo (word version) Loops and combo (html version) Breaks and validation code (zipped) Break and validation code (word version) Break and validation code (html version)
Procedures and Functions	Procedure and function examples (zipped) Procedure and function examples (word version) Procedure and function examples (html version)
Forms and Menus	Form and menu examples (zipped) Form and menu examples (word version) Form and menu examples (html version)
Arrays	Arrays (zipped) Introduction to arrays (word version) Introduction to arrays (html version) Introduction to arrays continued (word version) Introduction to arrays continued (html version) Two dimensional arrays (word version) Two dimensional arrays (html version) Sort using array (word version) Sort using array (html version) Redim with an array (word version) Redim with an array (html version)
Printing	Print Projects (zipped) Printing (word version) Printing (html version)
	Introduction to ADO (zipped) Introduction to ADO (word version) Introduction to ADO (html version) More introduction including grids (zipped) More introduction including grids (word version) More introduction including grids (html version)



The screenshot shows the Visual Studio IDE with the following code in the main editor:

```
1 Public Class frmDept
2     Inherits System.Windows.Forms.Form
3     Dim deptArray(3) As String
4
5     Windows Form Designer generated code
6
7     Private Sub btnDept_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDept.Click
8         txtDeptName.Text = deptArray(CInt(txtDept.Text))
9     End Sub
10
11     Private Sub frmDept_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
12         deptArray(0) = "Books"
13         deptArray(1) = "Toys"
14         deptArray(2) = "Gifts"
15         deptArray(3) = "Cookware"
16     End Sub
17
18     Private Sub btnCheckDept_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCheckDept.Click
19         If IsNumeric(txtDept.Text) Then
20             If CInt(txtDept.Text) <= 3 Then
21                 txtDeptName.Text = deptArray(CInt(txtDept.Text))
22             Else
23                 txtDeptName.Text = "Invalid Department Number"
24             End If
25         Else
26             txtDeptName.Text = "Non-numeric Department Number"
27         End If
28     End Sub
29 End Class
```

Handwritten blue notes on the code:

- A list of department names: `Books`, `Toys`, `Gifts`, `Cookware`.
- A note `Dept Array (0)` with an arrow pointing to the `deptArray(0)` assignment in the `frmDept_Load` event handler.

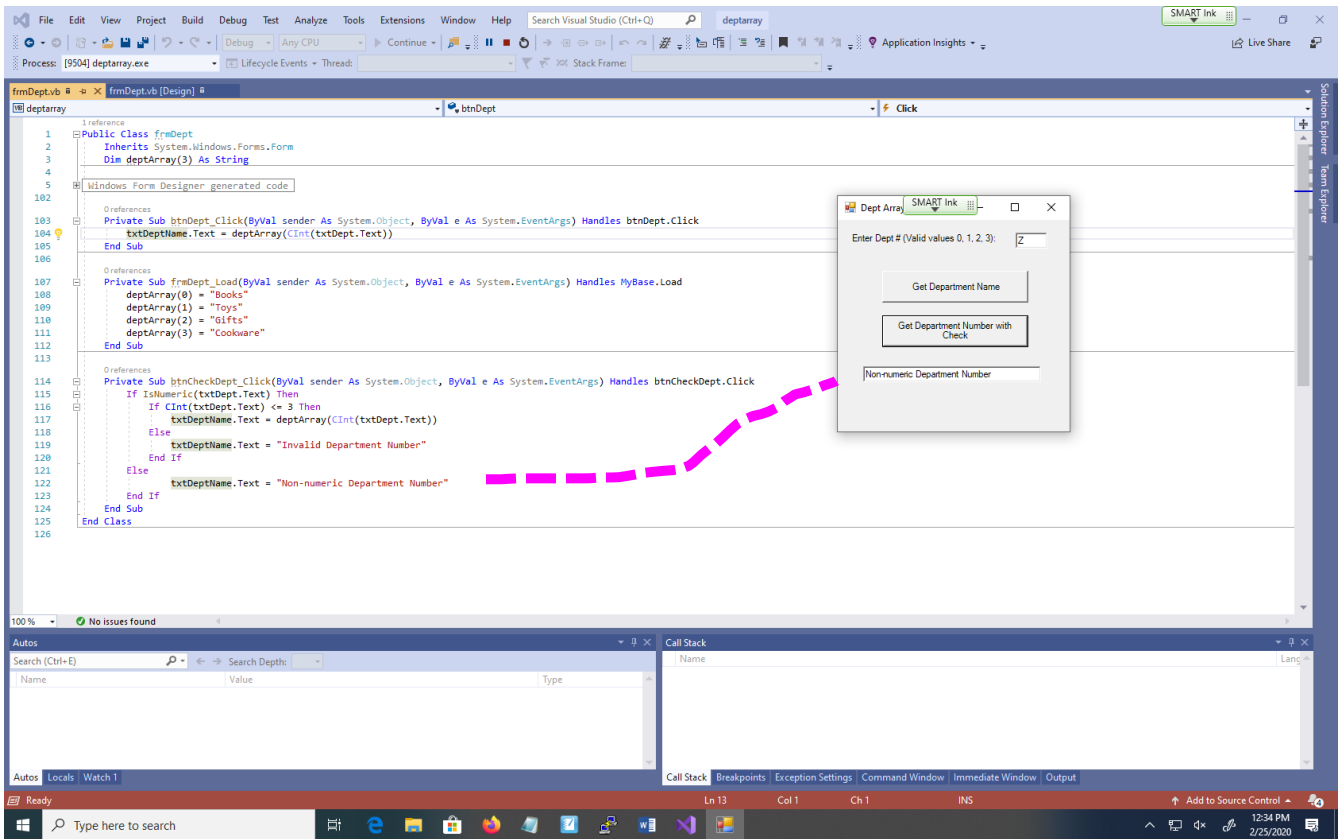
The screenshot displays the Visual Studio IDE with a VB.NET project named 'deplarray'. The code in 'frmDept.vb' defines a class 'frmDept' with two methods: 'btnDept_Click' and 'btnCheckDept_Click'. The 'btnDept_Click' method calls 'deptArray(CInt(txtDept.Text))' to update 'txtDeptName.Text'. The 'btnCheckDept_Click' method checks if the input is numeric and within the range 0-3, then updates 'txtDeptName.Text' with the corresponding department name from the 'deptArray' array. The 'deptArray' is initialized with 'Books', 'Toys', 'Gifts', and 'Cookware'.

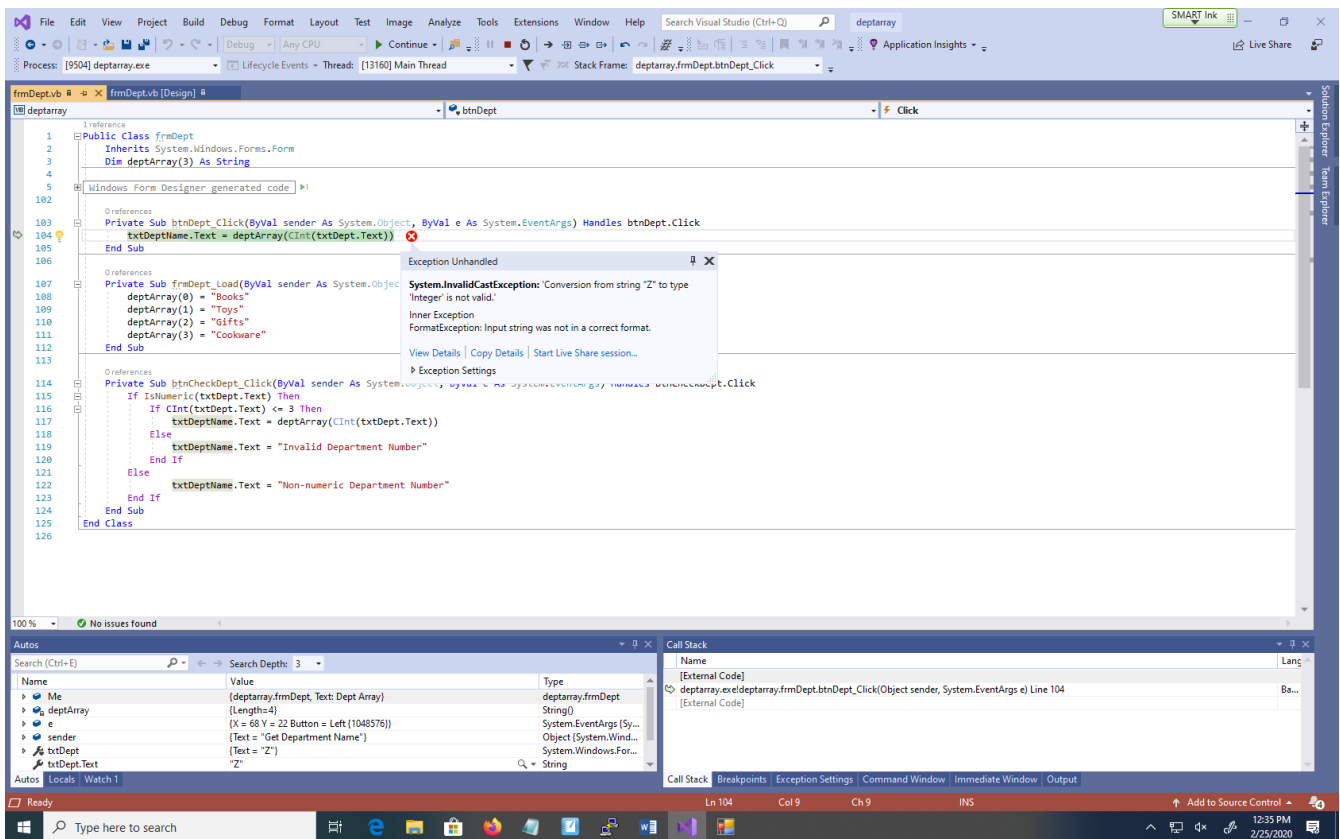
A dialog box titled 'Dept Array' is shown in the foreground, titled 'Enter Dept # (Valid values 0, 1, 2, 3):'. It contains a text box with the value '2', two buttons: 'Get Department Name' and 'Get Department Number with Check', and a text box displaying 'Gifts'.

```
1 Public Class frmDept
2     Inherits System.Windows.Forms.Form
3
4     Dim deptArray(3) As String
5
6     'Windows Form Designer generated code
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103 Private Sub btnDept_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDept.Click
104     txtDeptName.Text = deptArray(CInt(txtDept.Text))
105 End Sub
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187 Private Sub frmDept_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
188     deptArray(0) = "Books"
189     deptArray(1) = "Toys"
190     deptArray(2) = "Gifts"
191     deptArray(3) = "Cookware"
192 End Sub
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

The screenshot displays the Visual Studio IDE with a VB.NET code file named 'frmDept.vb' open. The code defines a class 'frmDept' with several methods. A dialog box titled 'Dept Array' is overlaid on the code, showing a text input field with the value '5', two buttons labeled 'Get Department Name' and 'Get Department Number with Check', and a label 'Invalid Department Number'. A pink dashed line highlights the 'Invalid Department Number' label in the code, which is connected to the dialog box. The code includes a 'btnDept_Click' method that calls 'deptArray(CInt(txtDept.Text))' and a 'btnCheckDept_Click' method that checks if the input is numeric and within the range 0-3. The IDE interface shows the 'deplarray' project, the 'frmDept' class, and the 'Click' event handler. The bottom status bar indicates 'Ready' and 'Ln 13 Col 1 Ch 1 INS'.

```
1 Public Class frmDept
2 Inherits System.Windows.Forms.Form
3 Dim deptArray(3) As String
4
5 Windows Form Designer generated code
6
7
8
9
102
103 Private Sub btnDept_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDept.Click
104 txtDeptName.Text = deptArray(CInt(txtDept.Text))
105 End Sub
106
107
108 Private Sub frmDept_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
109 deptArray(0) = "Books"
110 deptArray(1) = "Toys"
111 deptArray(2) = "Gifts"
112 deptArray(3) = "Cookware"
113 End Sub
114
115 Private Sub btnCheckDept_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCheckDept.Click
116 If IsNumeric(txtDept.Text) Then
117 If CInt(txtDept.Text) <= 3 Then
118 txtDeptName.Text = deptArray(CInt(txtDept.Text))
119 Else
120 txtDeptName.Text = "Invalid Department Number"
121 End If
122 Else
123 txtDeptName.Text = "Non-numeric Department Number"
124 End If
125 End Sub
126 End Class
```





The screenshot shows the Visual Studio IDE with a C# code file named `frmDept.vb` open. The code defines a class `frmDept` that inherits from `System.Windows.Forms.Form` and has a `DeptArray` property. It contains three event handlers: `btnDept_Click`, `frmDept_Load`, and `btnCheckDept_Click`. The `btnDept_Click` handler is currently selected, and an exception is shown: `System.IndexOutOfRangeException: 'Index was outside the bounds of the array.'`

```
1 public class frmDept
2 inherits System.Windows.Forms.Form
3 Dim deptArray(3) As String
4
5 Windows Form Designer generated code
6
7
8
9
102
103 Private Sub btnDept_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDept.Click
104     txtDeptName.Text = deptArray(CInt(txtDept.Text))
105 End Sub
106
107 Private Sub frmDept_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
108     deptArray(0) = "Books"
109     deptArray(1) = "Toys"
110     deptArray(2) = "Gifts"
111     deptArray(3) = "Cookware"
112 End Sub
113
114 Private Sub btnCheckDept_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCheckDept.Click
115     If IsNumeric(txtDept.Text) Then
116         If CInt(txtDept.Text) <= 3 Then
117             txtDeptName.Text = deptArray(CInt(txtDept.Text))
118         Else
119             txtDeptName.Text = "Invalid Department Number"
120         End If
121     Else
122         txtDeptName.Text = "Non-numeric Department Number"
123     End If
124 End Sub
125 End Class
126
```

The Autos window shows the following variables:

Name	Value	Type
Me	(deptarray.frmDept, Text: Dept Array)	deptarray.frmDept
deptArray	(Length=4)	String()
e	(X = 93 Y = 20 Button = Left [1040576])	System.EventArgs [System.Windows.Forms]
sender	(Text = "Get Department Name")	Object [System.Windows.Forms]
txtDept	(Text = "5")	System.Windows.Forms.TextBox
txtDept.Text	"5"	String

The Call Stack window shows the following call:

Name	Language
deptarray.exe!deptarray.frmDept.btnDept_Click(Object sender, System.EventArgs e) Line 104	Basic
[External Code]	

The screenshot displays the Visual Studio IDE with a VB.NET project named 'deptrarray'. The code in 'frmDept.vb' defines a class 'frmDept' with three methods: 'btnDept_Click', 'frmDept_Load', and 'btnCheckDept_Click'. The 'btnDept_Click' method calls 'deptArray(CInt(txtDept.Text))'. The 'frmDept_Load' method initializes an array 'deptArray' with values 'Books', 'Toys', 'Gifts', and 'Cookware'. The 'btnCheckDept_Click' method checks if the input is numeric and within the range 0-3, then updates 'txtDeptName.Text' with the corresponding department name or an error message.

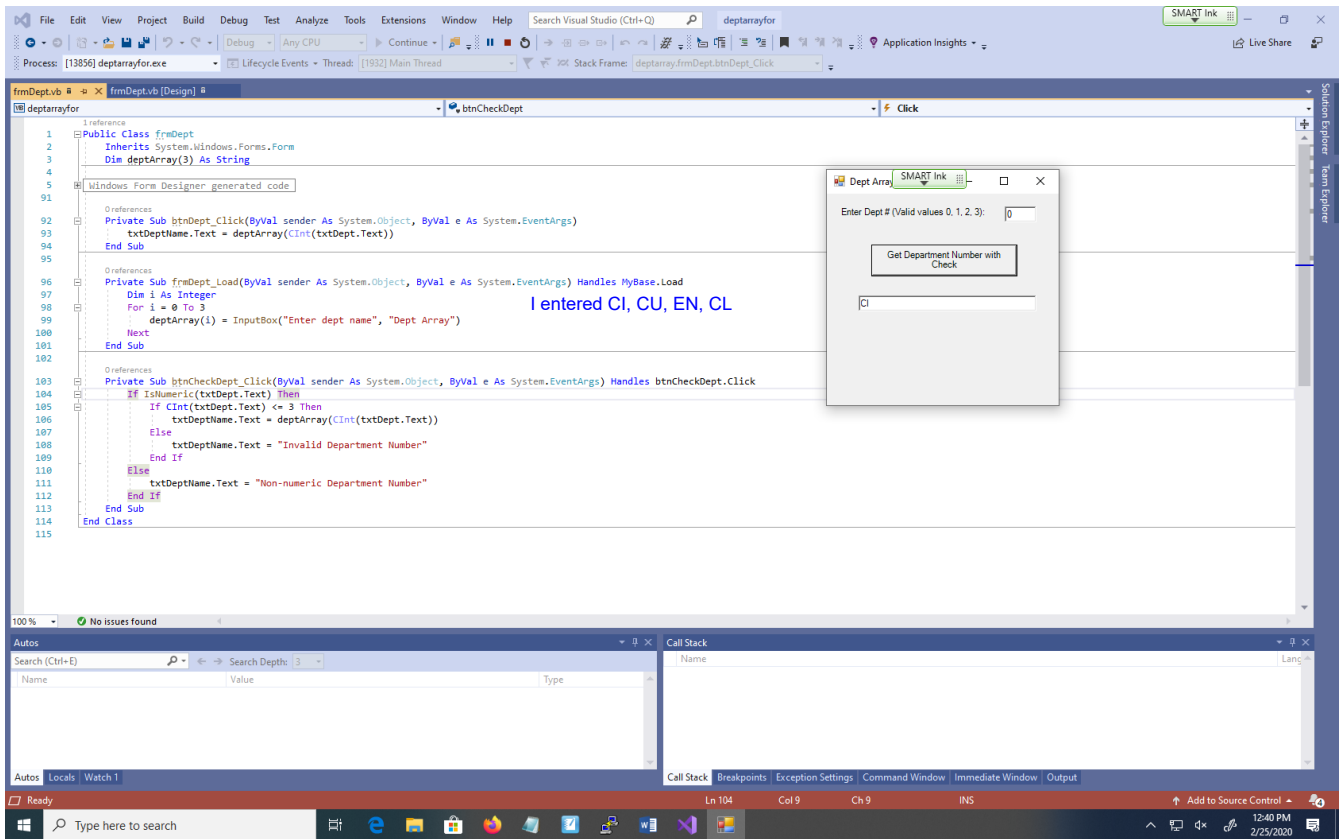
```
1 Public Class frmDept
2 Inherits System.Windows.Forms.Form
3 Dim deptArray(3) As String
4
5 Windows Form Designer generated code
102
103 Private Sub btnDept_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDept.Click
104 txtDeptName.Text = deptArray(CInt(txtDept.Text))
105 End Sub
106
107 Private Sub frmDept_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
108 deptArray(0) = "Books"
109 deptArray(1) = "Toys"
110 deptArray(2) = "Gifts"
111 deptArray(3) = "Cookware"
112 End Sub
113
114 Private Sub btnCheckDept_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCheckDept.Click
115 If IsNumeric(txtDept.Text) Then
116 If CInt(txtDept.Text) <= 3 Then
117 txtDeptName.Text = deptArray(CInt(txtDept.Text))
118 Else
119 txtDeptName.Text = "Invalid Department Number"
120 End If
121 Else
122 txtDeptName.Text = "Non-numeric Department Number"
123 End If
124 End Sub
125 End Class
126
```

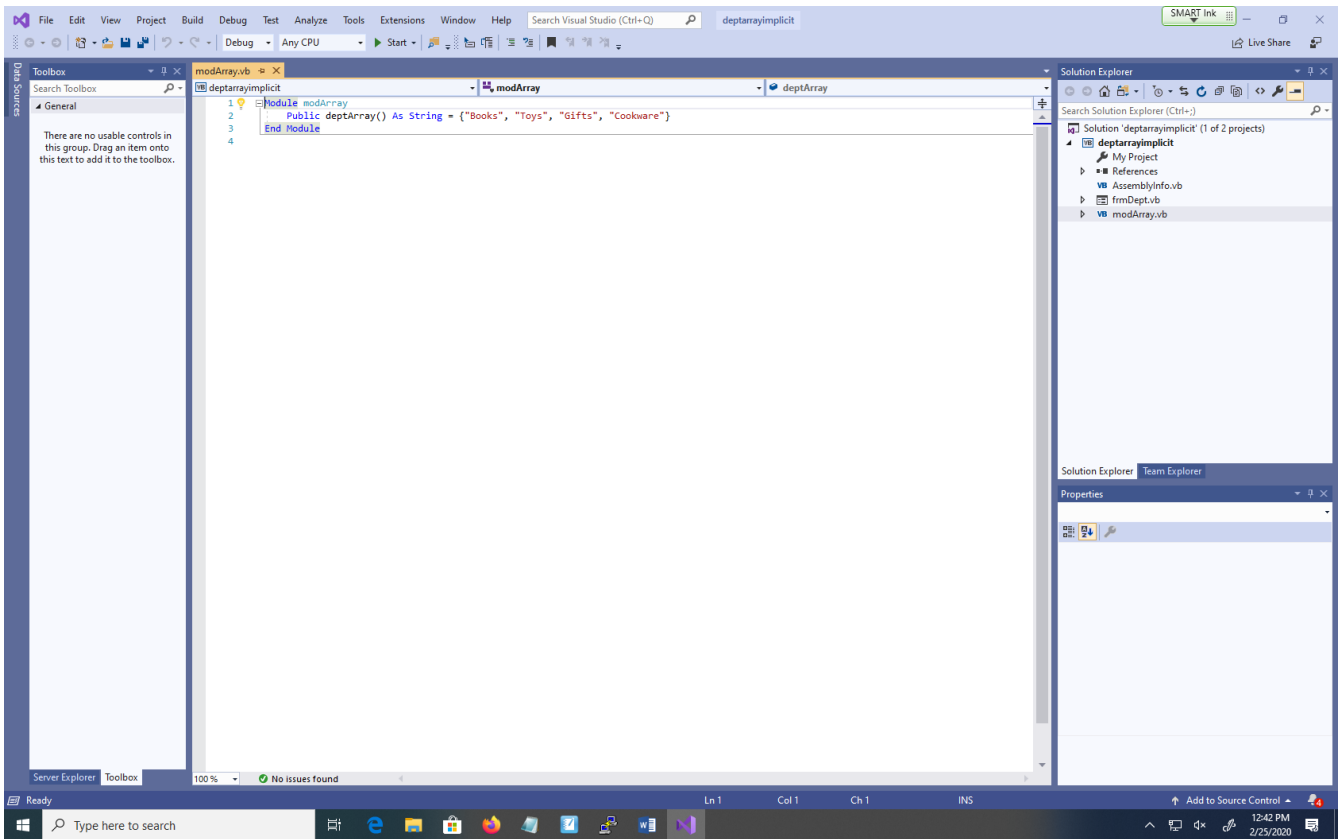
The application window 'Dept Array' is running, showing a text input field with '1', two buttons ('Get Department Name' and 'Get Department Number with Check'), and a text box containing 'Toys'.

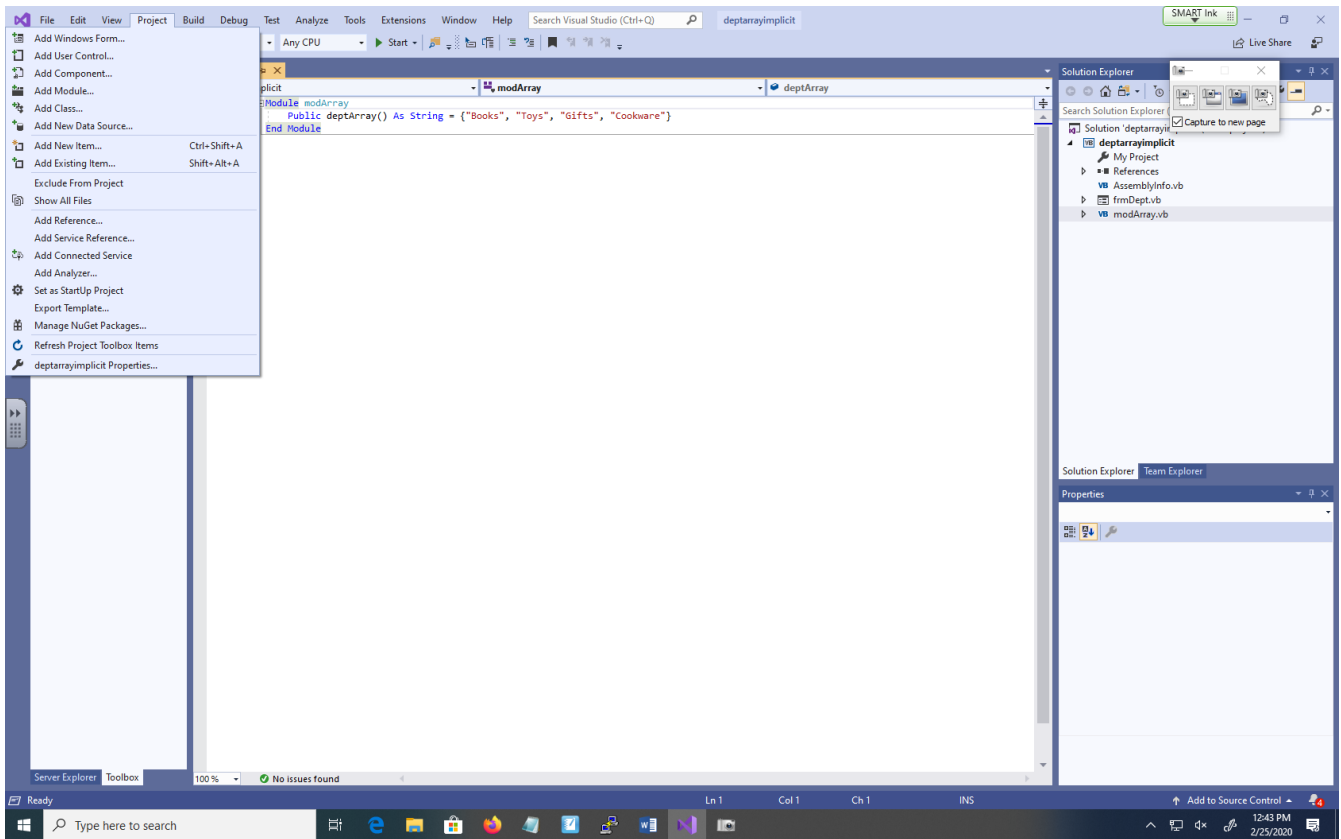
The screenshot displays the Visual Studio IDE with a VB.NET project named 'deptrarray'. The code in 'frmDept.vb' defines a class 'frmDept' that inherits from 'System.Windows.Forms.Form'. It includes a 'btnDept' control and a 'Click' event handler. The code defines an array 'deptArray' with three elements: 'Books', 'Toys', and 'Gifts'. The 'Click' event handler calls 'deptArray(CInt(txtDept.Text))' to update 'txtDeptName.Text'. A second event handler, 'btnCheckDept_Click', checks if the input is numeric and within the range 0-3, displaying an error message if not. A third event handler, 'btnDept_Click', is also present but its code is not fully visible.

```
1 reference
2 Public Class frmDept
3 Inherits System.Windows.Forms.Form
4 Dim deptArray(3) As String
5 Windows Form Designer generated code
102
103 Private Sub btnDept_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDept.Click
104 txtDeptName.Text = deptArray(CInt(txtDept.Text))
105 End Sub
106
107 Private Sub frmDept_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
108 deptArray(0) = "Books"
109 deptArray(1) = "Toys"
110 deptArray(2) = "Gifts"
111 deptArray(3) = "Cookware"
112 End Sub
113
114 Private Sub btnCheckDept_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCheckDept.Click
115 If IsNumeric(txtDept.Text) Then
116 If CInt(txtDept.Text) <= 3 Then
117 txtDeptName.Text = deptArray(CInt(txtDept.Text))
118 Else
119 txtDeptName.Text = "Invalid Department Number"
120 End If
121 Else
122 txtDeptName.Text = "Non-numeric Department Number"
123 End If
124 End Sub
125 End Class
126
```

The application window 'Dept Array' is running and shows a dialog box with the title 'Dept Array'. It contains a text input field with the value '1', a label 'Enter Dept # (Valid values 0, 1, 2, 3):', and two buttons: 'Get Department Name' and 'Get Department Number with Check'. Below the buttons is a text box containing the word 'Toys'.





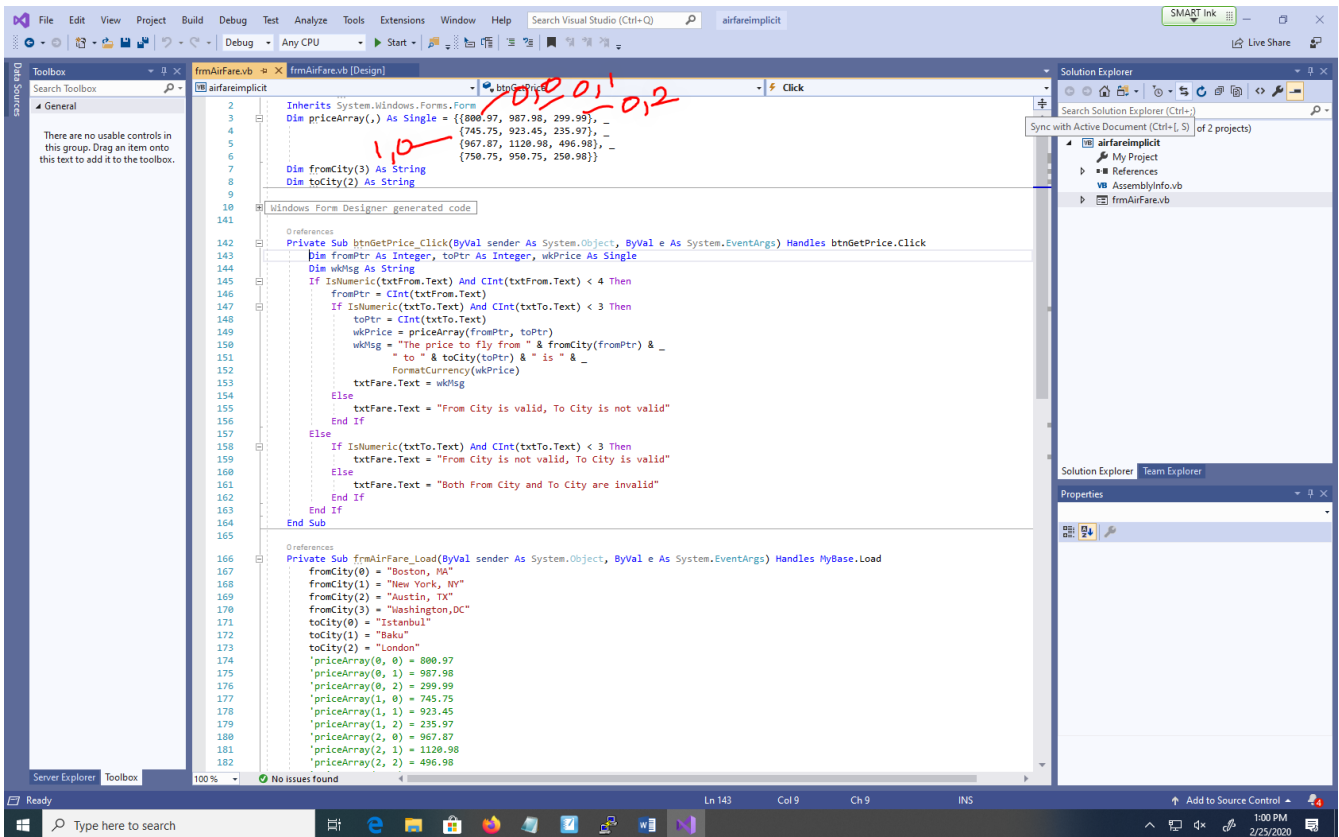


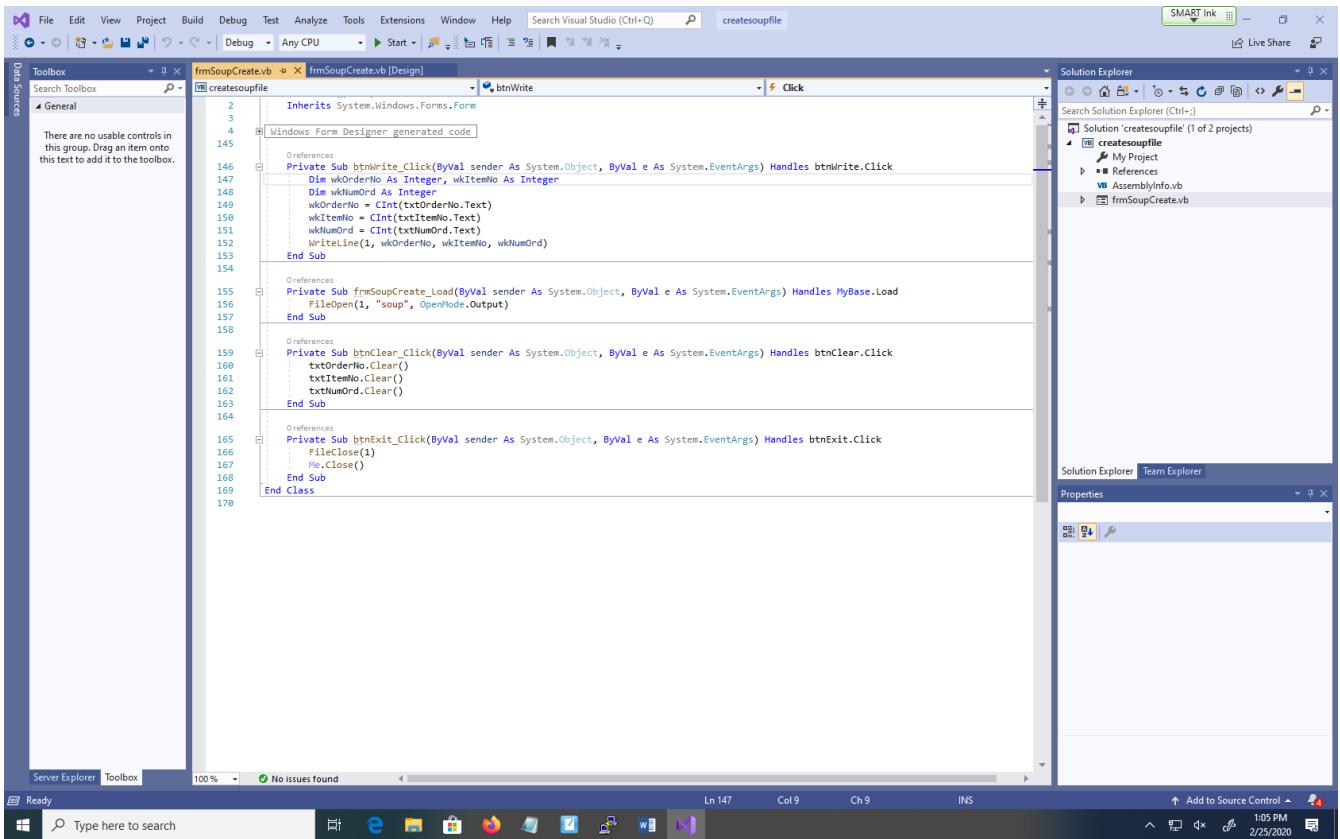
The screenshot shows the Visual Studio IDE with a VB.NET project named 'airfare'. The main window displays the code for the 'btnGetPrice_Click' event handler. The code includes logic to validate 'from' and 'to' city inputs and calculate the fare based on a price array. Handwritten blue annotations include 'From city' and 'to city' with arrows pointing to the respective variables in the code. A handwritten table is also present, showing the mapping of city indices to prices.

```
Private Sub btnGetPrice_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnGetPrice.Click
    Dim fromPtr As Integer, toPtr As Integer, wkPrice As Single
    Dim wkMsg As String
    If IsNumeric(txtFrom.Text) And CInt(txtFrom.Text) < 4 Then
        fromPtr = CInt(txtFrom.Text)
        If IsNumeric(txtTo.Text) And CInt(txtTo.Text) < 3 Then
            toPtr = CInt(txtTo.Text)
            wkPrice = priceArray(fromPtr, toPtr)
            wkMsg = "The price to fly from " & fromCity(fromPtr) & _
                " to " & toCity(toPtr) & " is " & _
                FormatCurrency(wkPrice)
            txtFare.Text = wkMsg
        Else
            txtFare.Text = "From City is valid, To City is not valid"
        End If
    Else
        If IsNumeric(txtTo.Text) And CInt(txtTo.Text) < 3 Then
            txtFare.Text = "From City is not valid, To City is valid"
        Else
            txtFare.Text = "Both From City and To City are invalid"
        End If
    End If
End Sub
```

from city \ to city	0	1	2	3
0	800.97	987.98		
1	987.98	299.99	923.45	
2	299.99	923.45	235.97	

```
Private Sub frmAirfare_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    fromCity(0) = "Boston, MA"
    fromCity(1) = "New York, NY"
    fromCity(2) = "Austin, TX"
    fromCity(3) = "Washington, DC"
    toCity(0) = "Istanbul"
    toCity(1) = "Baku"
    toCity(2) = "London"
    priceArray(0, 0) = 800.97
    priceArray(0, 1) = 987.98
    priceArray(0, 2) = 299.99
    priceArray(1, 0) = 745.75
    priceArray(1, 1) = 923.45
    priceArray(1, 2) = 235.97
    priceArray(2, 0) = 967.87
    priceArray(2, 1) = 1120.98
    priceArray(2, 2) = 496.98
    priceArray(3, 0) = 750.75
    priceArray(3, 1) = 950.75
    priceArray(3, 2) = 250.98
End Sub
```





The screenshot shows a Visual Studio IDE with a VB.NET code file named 'frmSoup.vb' open. The code defines a 'frmSoup' class that inherits from 'System.Windows.Forms.Form'. It includes several arrays for 'itemNo' and 'itemName', and event handlers for 'btnRead_Click', 'btnClear_Click', and 'btnExit_Click'. The 'btnRead_Click' handler reads data from a file named 'soup.txt' and displays it in a Notepad window.

The Notepad window, titled 'soup - Notepad', displays the following data:

```
1111, 12, 1
1212, 27, 2
1224, 45, 4
1225, 28, 2
2003, 17, 1
2027, 28, 4
3003, 3, 3
3010, 34, 3
4012, 15, 5
4078, 50, 2
5000, 19, 1
5012, 24, 4
```

A 'Search Soup' dialog box is also visible, showing the following fields:

- Order #: 1111
- Item #: 12
- Number Ordered: 1
- Item Name: CORN CHOWDER

The dialog has 'Read', 'Clear', and 'Exit' buttons.

The screenshot shows a Visual Studio IDE with a VB.NET code file named 'frmSoup.vb' open. The code defines a 'frmSoup' class that inherits from 'System.Windows.Forms.Form'. It includes several arrays for 'itemNo' and 'itemName'. The 'itemName' array lists various soup types: SEAFOOD CHOWDER, CORN CHOWDER, CLAM CHOWDER, TOMATO SOUP, CHICKEN SOUP, VEGETABLE SOUP, ONION SOUP, GREEN PEA SOUP, and MONTON SOUP. The code also contains event handlers for 'btnRead_Click', 'btnClear_Click', and 'btnExit_Click'. A 'Notepad' window is overlaid on the code, displaying a list of order data with columns for Order #, Item #, Number Ordered, and Item Name. The data includes entries like '1111, 12, 1' for 'SEAFOOD CHOWDER' and '2027, 28, 4' for 'ONION SOUP'. The Notepad window also has a 'Search Sol' dialog box open, showing search criteria for '1212', '27', '2', and 'ONION SOUP'.

```
Public Class frmSoup
    Inherits System.Windows.Forms.Form
    Dim itemNoArray(8) As Integer
    Dim itemNameArray(8) As String
    ' Windows Form Designer generated code
    Private Sub frmSoupCreate_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        itemNoArray(0) = 3
        itemNoArray(1) = 12
        itemNoArray(2) = 15
        itemNoArray(3) = 17
        itemNoArray(4) = 24
        itemNoArray(5) = 25
        itemNoArray(6) = 27
        itemNoArray(7) = 28
        itemNoArray(8) = 45
        itemNameArray(0) = "SEAFOOD CHOWDER"
        itemNameArray(1) = "CORN CHOWDER"
        itemNameArray(2) = "CLAM CHOWDER"
        itemNameArray(3) = "TOMATO SOUP"
        itemNameArray(4) = "CHICKEN SOUP"
        itemNameArray(5) = "VEGETABLE SOUP"
        itemNameArray(6) = "ONION SOUP"
        itemNameArray(7) = "GREEN PEA SOUP"
        itemNameArray(8) = "MONTON SOUP"
        FileOpen(1, "soup", OpenMode.Input)
    End Sub
    Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClear.Click
        txtOrderNo.Clear()
        txtItemNo.Clear()
        txtNumOrd.Clear()
    End Sub
    Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnExit.Click
        FileClose(1)
        Me.Close()
    End Sub
    Private Sub btnRead_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnRead.Click
        Dim wkOrderNo As Integer
        Dim wkNumOrd As Integer
        If Not EOF(1) Then
            Input(1, wkOrderNo)
            Input(1, wkItemNo)
            Input(1, wkNumOrd)
        End If
    End Sub
End Class
```

Notepad Content:

```
1111, 12, 1
1212, 27, 2
1224, 45, 4
1225, 28, 2
2003, 17, 1
2027, 28, 4
3003, 3, 3
3010, 34, 3
4012, 15, 5
4078, 50, 2
5000, 19, 1
5012, 24, 4
```

Search Sol Dialog:

```
Order #: 1212
Item #: 27
Number Ordered: 2
Item Name: ONION SOUP
```

The screenshot shows a Visual Studio IDE with a code editor for `frmSoup.vb` and a Notepad window. The code defines a `frmSoup` class with arrays for `itemNo` and `itemName`. Handwritten red notes in the code editor say "Find 27 Index 6" and "Take nam with index 6". The Notepad window displays a list of order data with columns for Order #, Item #, Number Ordered, and Item Name. A small dialog box titled "Search Soup" is overlaid on the Notepad window, showing search criteria: Order #: 1212, Item #: 27, Number Ordered: 2, and Item Name: ONION SOUP. The dialog has "Read", "Clear", and "Exit" buttons.

```
1 Public Class frmSoup
2 Inherits System.Windows.Forms.Form
3 Dim itemNoArray(8) As Integer
4 Dim itemNameArray(8) As String
5
6 Windows Form Designer generated code
7
170
171 Private Sub frmSoupCreate_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
172 itemNoArray(0) = 3
173 itemNoArray(1) = 12
174 itemNoArray(2) = 15
175 itemNoArray(3) = 17
176 itemNoArray(4) = 24
177 itemNoArray(5) = 25
178 itemNoArray(6) = 27
179 itemNoArray(7) = 28
180 itemNoArray(8) = 45
181 itemNameArray(0) = "SEAFOOD CHOWDER"
182 itemNameArray(1) = "CORN CHOWDER"
183 itemNameArray(2) = "CLAM CHOWDER"
184 itemNameArray(3) = "TOMATO SOUP"
185 itemNameArray(4) = "CHICKEN SOUP"
186 itemNameArray(5) = "VEGETABLE SOUP"
187 itemNameArray(6) = "ONION SOUP"
188 itemNameArray(7) = "GREEN PEA SOUP"
189 itemNameArray(8) = "MONTON SOUP"
190 FileOpen(1, "soup", OpenMode.Input)
191 End Sub
192
193 Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClear.Click
194 txtOrderNo.Clear()
195 txtItemNo.Clear()
196 txtNumOrd.Clear()
197 End Sub
198
199 Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnExit.Click
200 FileClose(1)
201 Me.Close()
202 End Sub
203
204 Private Sub btnRead_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnRead.Click
205 Dim wkOrderNo As Integer
206 Dim wkNumOrd As Integer
207 If Not EOF(1) Then
208 Input(1, wkOrderNo)
209 Input(1, wkItemNo)
210 Input(1, wkNumOrd)
```

Find 27
Index 6

Take nam
with index 6

1111, 12, 1
1212, 27, 2
1224, 45, 4
1225, 28, 2
2003, 17, 1
2027, 28, 4
3003, 3, 3
3010, 34, 3
4012, 15, 5
4078, 50, 2
5000, 19, 1
5012, 24, 4

Search Soup SMART Ink

Order #: 1212
Item #: 27
Number Ordered: 2
Item Name: ONION SOUP

Read Clear Exit

```
202 End Sub
203
204 Private Sub btnRead_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnRead.Click
205     Dim wkOrderNo As Integer, wkItemNo As Integer
206     Dim wkNumOrd As Integer
207     If Not EOF(1) Then
208         Input(1, wkOrderNo)
209         Input(1, wkItemNo)
210         Input(1, wkNumOrd)
211         txtOrderNo.Text = wkOrderNo
212         txtItemNo.Text = wkItemNo
213         txtNumOrd.Text = wkNumOrd
214         txtItemName.Text = SearchArray(wkItemNo)
215         txtItemName.Text = SearchWhile(wkItemNo)
216     Else
217         MessageBox.Show("EOF reached")
218         btnRead.Visible = False
219     End If
220 End Sub
221
222 Function SearchArray(ByVal wkItemNo)
223     Dim itemSub As Integer = 0
224     Dim matchInd As String = "NO"
225     Do Until itemSub > 8 Or matchInd = "YES"
226         If wkItemNo = itemNoArray(itemSub) Then
227             matchInd = "YES"
228         Else
229             itemSub = itemSub + 1
230         End If
231     Loop
232     If matchInd = "YES" Then
233         Return itemNameArray(itemSub)
234     Else
235         Return "Match Not Found"
236     End If
237 End Function
238
239 Function SearchWhile(ByVal wkItemNo)
240     Dim itemSub As Integer = 0
241     Dim matchInd As String = "NO"
242     Do While itemSub < 9 And matchInd = "NO"
243         If wkItemNo = itemNoArray(itemSub) Then
244             matchInd = "YES"
245         Else
246             itemSub = itemSub + 1
247         End If
248     Loop
249     If matchInd = "YES" Then
250         Return itemNameArray(itemSub)
251     Else
252         Return "Match Not Found"
253     End If
254 End Function
```

Handwritten annotations in blue ink:

- A fraction $\frac{\text{Item Sub}}{0}$ is written to the left of the code.
- The text "match End NO" is written to the right of the code.
- The number "27" is written with two arrows pointing to lines 225 and 226.

Item Sub match Ind
NO
YES

27

Onion Soup

```
itemNoArray(0) = 3  
itemNoArray(1) = 12  
itemNoArray(2) = 15  
itemNoArray(3) = 17  
itemNoArray(4) = 24  
itemNoArray(5) = 25  
itemNoArray(6) = 27  
itemNoArray(7) = 28  
itemNoArray(8) = 45  
itemNameArray(0) = "SEAFOOD CHOWDER"  
itemNameArray(1) = "CORN CHOWDER"  
itemNameArray(2) = "CLAM CHOWDER"  
itemNameArray(3) = "TOMATO SOUP"  
itemNameArray(4) = "CHICKEN SOUP"  
itemNameArray(5) = "VEGETABLE SOUP"  
itemNameArray(6) = "ONION SOUP"  
itemNameArray(7) = "GREEN PEA SOUP"  
itemNameArray(8) = "WONTON SOUP"
```

Order #: 1212
Item #: 27
Number Ordered: 2
Item Name: ONION SOUP

Read Clear Exit

```
202 End Sub
203
204 Private Sub btnRead_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnRead.Click
205     Dim wkOrderNo As Integer, wkItemNo As Integer
206     Dim wkNumOrd As Integer
207     If Not EOF(1) Then
208         Input(1, wkOrderNo)
209         Input(1, wkItemNo)
210         Input(1, wkNumOrd)
211         txtOrderNo.Text = wkOrderNo
212         txtItemNo.Text = wkItemNo
213         txtNumOrd.Text = wkNumOrd
214         txtItemName.Text = SearchArray(wkItemNo)
215         txtItemName.Text = SearchWhile(wkItemNo)
216     Else
217         MessageBox.Show("EOF reached")
218         btnRead.Visible = False
219     End If
220 End Sub
221
222 Function SearchOrderArray(ByVal wkItemNo)
223     Dim itemSub As Integer = 0
224     Dim matchInd As String = "NO"
225     Do Until itemSub > 8 Or matchInd = "YES"
226         If wkItemNo = itemNoArray(itemSub) Then
227             matchInd = "YES"
228         Else
229             itemSub = itemSub + 1
230         End If
231     Loop
232     If matchInd = "YES" Then
233         Return itemNameArray(itemSub)
234     Else
235         Return "Match Not Found"
236     End If
237 End Function
238
239 Function SearchWhile(ByVal wkItemNo)
240     Dim itemSub As Integer = 0
241     Dim matchInd As String = "NO"
242     Do While itemSub < 9 And matchInd = "NO"
243         If wkItemNo = itemNoArray(itemSub) Then
244             matchInd = "YES"
245         Else
246             itemSub = itemSub + 1
247         End If
248     Loop
249     If matchInd = "YES" Then
250         Return itemNameArray(itemSub)
251     Else
252         Return "Match Not Found"
253     End If
254 End Function
```

SMART Ink

Order #: 1224
Item #: 45
Number Ordered: 4
Item Name: WONTON SOUP

Read Clear Exit

SMART Ink

```
1111,12,1
1212,27,2
1224,45,4
1225,28,2
2003,17,1
2027,28,4
3003,3,3
3010,34,3
4012,15,5
4078,50,2
5000,19,1
5012,24,4
```

SMART Ink

```
itemNoArray(0) = 3
itemNameArray(1) = "CORN CHOWDER"
itemNoArray(1) = 12
itemNameArray(2) = "CLAM CHOWDER"
itemNoArray(2) = 15
itemNameArray(3) = 17
itemNoArray(3) = 17
itemNameArray(4) = "TOMATO SOUP"
itemNoArray(4) = 24
itemNameArray(4) = "CHICKEN SOUP"
itemNoArray(5) = 25
itemNameArray(5) = "VEGETABLE SOUP"
itemNoArray(6) = 27
itemNameArray(6) = "ONION SOUP"
itemNoArray(7) = 28
itemNameArray(7) = "GREEN PEA SOUP"
itemNoArray(8) = 45
itemNameArray(8) = "WONTON SOUP"
```

The screenshot displays a Visual Studio IDE with the following components:

- Code Editor (frmSoup.vb):** Contains a private sub procedure `btnRead_Click` and two functions: `SearchOrderArray` and `SearchWhile`. The code handles user input for order number, item number, and item name, and performs a search for the item name in a list.
- SMART Ink Window:** A dialog box with input fields for "Order #:" (1225), "Item #:" (28), "Number Ordered:" (2), and "Item Name:" (GREEN PEA SOUP). It includes "Read", "Clear", and "Exit" buttons.
- Notepad Window:** Displays the output of the search, showing item numbers and names in an array format:

```
itemNoArray(0) = 3
itemNoArray(1) = 12
itemNoArray(2) = 15
itemNoArray(3) = 17
itemNoArray(4) = 24
itemNoArray(5) = 25
itemNoArray(6) = 27
itemNoArray(7) = 28
itemNoArray(8) = 45
itemNameArray(0) = "SEAFOOD CHOWDER"
itemNameArray(1) = "CORN CHOWDER"
itemNameArray(2) = "CLAM CHOWDER"
itemNameArray(3) = "TOMATO SOUP"
itemNameArray(4) = "CHICKEN SOUP"
itemNameArray(5) = "VEGETABLE SOUP"
itemNameArray(6) = "ONION SOUP"
itemNameArray(7) = "GREEN PEA SOUP"
itemNameArray(8) = "WONTON SOUP"
```


The screenshot displays a Visual Studio IDE with the following components:

- Code Editor (frmSoup.vb):** Contains a private sub `btnRead_Click` and two functions: `SearchOrderArray` and `SearchWhile`. The code handles user input for order and item numbers and returns arrays of item numbers and names.
- SMART Ink Search Window:** A dialog box with input fields for Order # (3010), Item # (34), and Number Ordered (3). The Item Name field shows "Match Not Found". Buttons for Read, Clear, and Exit are present.
- SMART Ink Results Window:** A Notepad window displaying the output of the search, showing item numbers and names in an array format.

```
1111,12,1
1212,27,2
1224,45,4
1225,28,2
2003,17,1
2027,28,4
3003,3,3
3010,34,3
4012,15,5
4078,50,2
5000,19,1
5012,24,4

itemNoArray(0) = 3
itemNoArray(1) = 12
itemNoArray(2) = 15
itemNoArray(3) = 17
itemNoArray(4) = 24
itemNoArray(5) = 25
itemNoArray(6) = 27
itemNoArray(7) = 28
itemNoArray(8) = 45
itemNameArray(0) = "SEAFOOD CHOWDER"
itemNameArray(1) = "CORN CHOWDER"
itemNameArray(2) = "CLAM CHOWDER"
itemNameArray(3) = "TOMATO SOUP"
itemNameArray(4) = "CHICKEN SOUP"
itemNameArray(5) = "VEGETABLE SOUP"
itemNameArray(6) = "ONION SOUP"
itemNameArray(7) = "GREEN PEA SOUP"
itemNameArray(8) = "WONTON SOUP"
```

```
202 End Sub
203
204 Private Sub btnRead_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnRead.Click
205     Dim wkOrderNo As Integer, wkItemNo As Integer
206     Dim wkNumOrd As Integer
207     If Not EOF(1) Then
208         Input(1, wkOrderNo)
209         Input(1, wkItemNo)
210         Input(1, wkNumOrd)
211         txtOrderNo.Text = wkOrderNo
212         txtItemNo.Text = wkItemNo
213         txtNumOrd.Text = wkNumOrd
214         txtItemName.Text = SearchArray(wkItemNo)
215         txtItemName.Text = SearchWhile(wkItemNo)
216     Else
217         MessageBox.Show("EOF reached")
218         btnRead.Visible = False
219     End If
220 End Sub
221
222 Private Function SearchOrderArray(ByVal wkItemNo)
223     Dim itemSub As Integer = 0
224     Dim matchInd As String = "NO"
225     Do Until itemSub > 8 Or matchInd = "YES"
226         If wkItemNo = itemNoArray(itemSub) Then
227             matchInd = "YES"
228         Else
229             itemSub = itemSub + 1
230         End If
231     Loop
232     If matchInd = "YES" Then
233         Return itemNameArray(itemSub)
234     Else
235         Return "Match Not Found"
236     End If
237 End Function
238
239 Private Function SearchWhile(ByVal wkItemNo)
240     Dim itemSub As Integer = 0
241     Dim matchInd As String = "NO"
242     Do While itemSub < 9 And matchInd = "NO"
243         If wkItemNo = itemNoArray(itemSub) Then
244             matchInd = "YES"
245         Else
246             itemSub = itemSub + 1
247         End If
248     Loop
249     If matchInd = "YES" Then
250         Return itemNameArray(itemSub)
251     Else
252         Return "Match Not Found"
253     End If
254 End Function
```

Search Soup

Order #: 5012
Item #: 24
Number Ordered: 4
Item Name: CHICKEN SOUP

Read Clear Exit

EOF reached

OK

SMART Ink

```
1111,12,1
1212,27,2
1224,45,4
1225,28,2
2003,17,1
2027,28,4
3003,3,3
3010,34,3
4012,15,5
4078,50,2
5000,19,1
5012,24,4

itemNoArray(0) = 3
itemNoArray(1) = 12
itemNoArray(2) = 15
itemNoArray(3) = 17
itemNoArray(4) = 24
itemNoArray(5) = 25
itemNoArray(6) = 27
itemNoArray(7) = 28
itemNoArray(8) = 45
itemNameArray(0) = "SEAFOOD CHOWDER"
itemNameArray(1) = "CORN CHOWDER"
itemNameArray(2) = "CLAM CHOWDER"
itemNameArray(3) = "TOMATO SOUP"
itemNameArray(4) = "CHICKEN SOUP"
itemNameArray(5) = "VEGETABLE SOUP"
itemNameArray(6) = "ONION SOUP"
itemNameArray(7) = "GREEN PEA SOUP"
itemNameArray(8) = "WONTON SOUP"
```

The image shows a screenshot of the Visual Studio code editor with a VB.NET code file open. The code contains two functions: `SearchArray` and `SearchWhile`. Handwritten annotations in red and blue ink explain the loop conditions.

```
221 Function SearchArray(ByVal wkItemMo)
222     Dim itemSub As Integer = 0
223     Dim matchInd As String = "NO"
224     Do Until itemSub > 8 Or matchInd = "YES"
225         If wkItemMo = itemMoArray(itemSub) Then
226             matchInd = "YES"
227         Else
228             itemSub = itemSub + 1
229         End If
230     Loop
231     If matchInd = "YES" Then
232         Return itemNameArray(itemSub)
233     Else
234         Return "Match Not Found"
235     End If
236 End Function
237
238 'Function SearchWhile(ByVal wkItemMo)
239     Dim itemSub As Integer = 0
240     Dim matchInd As String = "NO"
241     Do While itemSub < 9 And matchInd = "NO"
242         If wkItemMo = itemMoArray(itemSub) Then
243             matchInd = "YES"
244         Else
245             itemSub = itemSub + 1
246         End If
247     Loop
248     If matchInd = "YES" Then
249         Return itemNameArray(itemSub)
250     Else
251         Return "Match Not Found"
252     End If
253 End Function
254
255 End Class
```

Annotations:

- Red:** "get me out of loop" with an arrow pointing to the condition `Or matchInd = "YES"` in the `SearchArray` function.
- Blue:** "either can get me out" with an arrow pointing to the same condition.
- Red:** "keep me in loop" with an arrow pointing to the condition `And matchInd = "NO"` in the `SearchWhile` function.
- Blue:** "both true to stay in either not true out" with an arrow pointing to the same condition.