

รายการอ้างอิง

ภาษาไทย

- ชาญ ฉันทอุดมพร. การบริหารงานโลจิสติกส์. เอกสารประกอบการสัมมนาวิชาการ, กรุงเทพมหานคร :
ดับเบิลดีเจ็ดส์. 2539. (อัดสำเนา)
- เนาวรัตน์ บุญไฟโล. เทคนิคการออกแบบคลังสินค้า การสร้างระบบการจัดการอย่างมีประสิทธิภาพ. กรุงเทพฯ
มหานคร : อัลฟา แมเนจเม้นท์ คอนซัลแตนท์ จำกัด, 2539. (อัดสำเนา)
- สุทธิ ศรีเพชรदानนท์. แบบจำลองการจัดการเส้นทางเดินรถสำหรับกระบวนการขนส่งสินค้า วิทยานิพนธ์ปริญญาโทบัณฑิต
ภาควิชาชีพวิศวกรรมอุตสาหกรรม บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2536
- ศิริจันทร์ ทองประเสริฐ. ระบบพัสดุคงคลัง. กรุงเทพมหานคร : จุฬาลงกรณ์มหาวิทยาลัย, 2538.

ภาษาอังกฤษ

- Blanchard, B., S. Logistics Engineering and Management. New Jersey : Prentice-Hall, 1992.
- Bookbinder, J.H., and Kathleen E. Reece Vehicle routing Considerations in distribution
system design. European Journal of Operation Research. 37 (1988) : 204-213.
- Goetschalckx, M. Storage and Retrieval Policies for Efficient Order Picking Operation.
Doctoral dissertation, Georgia Institute of Technology, 1983.
- Goetschalckx, M., and Donald Rattiff Order Picking In An Aisle. IIE Transactions 20 (March
1988) : 53-62.
- Hatton, G. Order Picking. The Gower Handbook of Logistics and Distribution Management,
pp. 245-266. London : Gower, 1992.
- Jarvis, J.M., and McDowell, F. D., Optimal Product Layout in an Order Picking Warehouse.
IIE Transactions 23 (1991) : 93-102.
- Klibbua, V., Warehouse Design and Vehicle Routing of Distribution in Northern Thailand.
Master's Thesis AIT, 1990.
- Lambert, D.M., and Jameo R. Stock. Strategic Logistics Management. Boston : Richard D.
Irwin, 1993.

- Malmberg, C.I., Optimization of Cube-Pre-Order Index Warehouse Layout with Zoning Constraint. International Journal of Production Research 33 (1995) : 465-482.
- Malmberg, C.I., and Deutsch, S.I., A Stock Location Model for Dual Address Order Picking System. IIE Transactions 4 (1988) : 40-52.
- Malmberg, C.I., and Krish N.B., On the Optimality of the Cube-Pre-Order Index for Conventional Warehouse with Dual Command Cycles. Material Flow 4 (1987) : 169-175.
- MD, A. Amin., Economic Planning and Scheduling of Distribution System, Considering Transportation and Inventory Cost. Master's Thesis AIT, 1989.
- Peterson, H., Logistics, Transportation and Distribution : Engineering Service and Profits. Industrial Engineering 25 (1993) : 103-120.
- Ratliff, H.D., and Rosenthal, A.S.; Order Picking in a Rectangular Warehouse : A Solvable case of Traveling Salesman Problem. Operation Research, pp.507-521. New York : McGraw-Hill, 1983.
- Strahan, B.A., Logistics Necessary for Success. IE News, Logistics, Transportation & Distribution Division 1 (1996) No.2.
- Sustin, C., Examination of 10-years Trends in Logistics Information system, Industrial Engineering 25 (1993) No.1.
- Yoon, C.S., and Sharp G.P., A Structured Procedure for analysis and Design of Order Pick System IIE Transaction 28 (1991) : 379-389.

ภาคผนวก ก.
✓ ปริมาตรและน้ำหนักสินค้า

รายการ	รหัสสินค้า	ขนาดสินค้า(ซม.)			ปริมาตร (ลิตร)	น้ำหนัก (กก.)
		ยาว	กว้าง	สูง		
1	120124	27.6	21.9	29.2	17.69	3.0
2	120048	28.9	21.6	21.3	13.28	4.4
3	120308	21.0	28.9	31.8	19.23	7.4
4	120096	28.6	22.2	25.7	16.34	6.0
5	120294	21.6	29.2	25.4	16.02	5.2
6	130072	48.3	37.8	19.7	35.91	14.0
7	120062	28.9	21.6	21.3	13.28	4.6
8	120232	28.9	21.6	21.3	13.28	4.4
9	126002	36.2	27.6	17.1	17.15	1.8
10	120246	28.9	21.6	25.7	16.06	6.0
11	120260	28.9	21.6	29.2	18.23	7.6
12	120110	28.6	22.2	25.7	16.34	6.0
13	120158	27.6	21.9	29.2	17.69	7.8
14	120356	28.9	21.6	21.3	13.28	4.4
15	122076	28.9	21.6	25.7	16.06	6.2
16	122028	28.9	21.6	21.3	13.28	4.8
17	122104	28.9	21.6	26.7	16.64	8.0
18	120370	28.9	21.6	25.7	16.06	6.0
19	122042	28.9	21.6	21.3	13.28	5.0
20	120348	28.9	21.6	29.2	18.23	7.4
21	130024	48.3	37.8	19.7	35.91	14.0

รายการ	รหัสสินค้า	ขนาดสินค้า(ซม.)			ปริมาตร (ลิตร)	น้ำหนัก (กก.)
		ยาว	กว้าง	สูง		
22	104010	14.5	9.7	9.8	18.57	6.2
23	104086	16.0	12.2	9.0	28.9	13.2
24	104100	40.6	31.1	23.5	29.71	13.6
25	104148	28.3	21	25.1	14.87	7.0
26	104058	36.8	24.8	20.3	18.53	6.4
27	104162	28.6	21.6	22.2	13.71	6.5
28	104024	34.3	24.8	20.3	18.53	6.4
29	104208	29.2	31.8	15.2	16.59	4.9
30	104222	40.6	24.8	15.9	11.48	4.8
31	104256	34.9	31.1	23.5	29.71	14.7
32	010112	39.4	20.3	29.3	20.73	7.8
33	010098	34.9	25.1	25.4	25.1	10.0
34	010140	30.3	20.7	24.8	17.86	7.6
35	008528	27.1	24.6	33.2	24.73	12.6
36	008344	36.6	18.7	25.8	13.09	6.6
37	008056	30.3	24.4	29.9	26.64	13.0
38	008480	30.3	24.6	33.2	24.73	12.6
39	008514	30.3	24.6	33.2	24.73	12.6
40	008008	36.6	24.4	29.9	26.64	13.0
41	008774	30.3	24.6	33.2	24.73	12.6
42	008022	36.6	24.4	29.9	26.64	13.0
43	008268	27.1	18.7	25.8	13.09	6.6
44	008316	27.1	18.7	25.8	13.09	6.6

รายการ	รหัสสินค้า	ขนาดสินค้า(ซม.)			ปริมาตร (ลิตร)	น้ำหนัก (กก.)
		ยาว	กว้าง	สูง		
45	008740	36.6	24.4	29.9	26.64	12.8
46	008378	27.1	18.7	25.2	13.09	6.6
47	008726	27.1	18.7	25.8	13.09	6.3
48	008282	27.1	18.7	25.8	13.09	6.6
49	008330	27.1	18.7	25.8	13.09	6.6
50	014100	32.5	32.5	29.0	30.61	14.0
51	014010	24.0	22.9	22.6	17.52	8.0
52	014024	33.0	19.5	26.4	17.02	7.0
53	014086	33.8	25.4	24.4	20.95	10.0
54	014072	34.5	26.4	29.9	25.5	9.6
55	014406	42.5	31.8	28.6	35.6	16.0
56	014058	33.0	19.5	26.5	17.02	7.0
57	014208	34.0	22.9	22.6	17.52	7.8
58	014482	33.8	25.4	24.4	20.95	9.8
59	014148	33.0	19.5	26.4	17.02	7.2
60	014134	34.0	22.9	22.6	17.52	8.0
61	002020	28.6	21.6	25.7	15.87	6.6
62	072072	20.0	21.9	22.4	9.81	1.8
63	080038	31.4	24.8	31.1	24.23	11.1
64	080052	29.3	24.6	34.2	24.63	11.0
65	080004	26.7	20.7	26.4	14.52	5.7
66	080080	37.8	25.7	31.8	30.88	17.8
67	056000	50.0	36.8	18.1	33.46	11.0

รายการ	รหัสสินค้า	ขนาดสินค้า(ซม.)			ปริมาตร (ลิตร)	น้ำหนัก (กก.)
		ยาว	กว้าง	สูง		
68	056034	27.9	21.6	20.3	12.26	5.2
69	062034	38.4	34.0	25.4	33.17	14.7
70	056062	48.3	29.2	27.0	38.06	17.4
71	062048	38.4	34.0	25.4	33.17	14.9
72	062000	38.4	34.0	25.4	33.17	4.0
73	072010	18.4	19.1	17.1	6.01	1.6
74	066206	25.4	21.6	10.5	5.75	1.5
75	066022	21.6	25.1	11.6	6.3	1.0
76	066220	25.4	21.6	10.5	5.75	1.5
77	066008	21.6	25.1	11.6	6.3	1.0
78	072024	18.4	19.1	17.1	6.01	1.6
79	072058	18.4	19.1	17.1	6.01	1.6
80	066330	25.4	21.6	10.5	5.75	1.5
81	801760	26.7	20.7	26.7	14.52	5.8
82	066316	21.6	25.1	11.6	6.3	1.0
83	008014	31.4	24.8	31.0	24.13	11.1
84	062158	38.4	34.0	25.4	33.17	14.6
85	066268	21.6	25.1	11.6	6.3	1.0
86	066282	25.4	21.6	10.5	5.75	1.5
87	062246	38.4	34.0	25.4	33.17	14.9
88	560124	50.2	36.8	18.1	33.46	11.4
89	062218	38.4	34.0	25.4	33.17	14.3
90	062232	38.4	34.0	25.4	33.17	14.7

รายการ	รหัสสินค้า	ขนาดสินค้า(ซม.)			ปริมาตร (ลิตร)	น้ำหนัก (กก.)
		ยาว	กว้าง	สูง		
91	030384	24.5	16.5	18.4	7.44	1.8
92	030370	24.5	16.5	18.4	7.44	1.8
93	029016	18.4	26.7	18.1	8.90	2.0
94	036288	17.1	22.6	14.0	5.9	1.6
95	036140	17.1	22.6	14.0	5.4	1.6
96	022128	23.5	17.8	21.0	8.75	2.9
97	036200	17.1	22.6	14.0	5.4	1.6
98	022080	23.5	17.8	21.0	8.75	2.9
99	022114	23.5	17.8	21.0	8.75	2.9
100	023226	28.6	21.0	21.6	12.93	5.0
101	023608	23.5	17.8	21.0	8.75	2.8
102	029044	18.4	26.7	18.1	8.89	2.0
103	030416	24.5	16.5	18.4	7.44	1.8
104	030430	24.5	16.5	18.4	7.44	1.8
105	028054	23.5	29.8	25.4	17.81	1.1
106	022066	23.5	17.8	21.0	8.75	2.9
107	036644	27.9	23.5	14.0	9.17	1.0
108	030034	24.5	16.5	18.4	7.44	2.0
109	030000	24.5	16.5	18.4	7.44	2.0
110	029078	18.4	26.7	18.1	8.89	2.0
111	022052	23.5	17.8	21.0	8.75	2.9
112	036678	27.9	23.5	14.0	9.17	1.0
113	036262	17.1	22.6	14.0	5.4	1.6

รายการ	รหัสสินค้า	ขนาดสินค้า(ซม.)			ปริมาตร (ลิตร)	น้ำหนัก (กก.)
		ยาว	กว้าง	สูง		
114	036630	27.9	23.5	14.0	9.17	1.0
115	023868	23.5	17.8	21.0	8.75	2.8
116	036036	28.6	37.5	12.1	12.92	4.5
117	030370	24.5	26.5	18.4	7.44	1.8
118	030384	24.5	16.5	18.4	7.44	1.8
119	028024	23.5	29.8	25.4	17.81	1.1
120	029030	18.4	26.7	18.1	8.90	2.0
121	030062	24.5	16.5	18.4	7.44	2.0
122	036248	17.1	22.6	14.0	5.40	1.6
123	028068	23.5	29.8	25.4	17.81	1.1
124	022038	23.5	17.8	21.0	8.75	2.9
125	036002	28.6	37.5	12.1	12.92	4.5
126	036556	14.0	16.5	14.6	3.37	0.5
127	036522	14.0	16.5	14.6	3.37	0.5
128	022510	17.8	23.5	26.0	3.80	3.8
129	030048	24.5	16.5	18.4	7.44	2.0
130	022052	23.5	17.8	21.0	8.75	2.9
131	022142	23.5	17.8	21.0	8.75	2.9
132	036372	22.39	33.0	17.1	12.94	4.5
133	036050	28.6	37.5	21.1	12.92	4.56
134	030416	24.5	16.5	18.4	7.44	1.8
135	036234	17.1	22.6	14.0	5.40	1.6
136	023028	29.8	21.6	21.9	14.12	5.7

รายการ	รหัสสินค้า	ขนาดสินค้า(ซม.)			ปริมาตร (ลิตร)	น้ำหนัก (กก.)
		ยาว	กว้าง	สูง		
137	022462	17.8	23.5	26	10.88	3.8
138	030430	24.5	16.5	18.4	7.44	1.8
139	036570	14.0	16.5	14.6	3.37	0.5
140	036064	28.6	37.5	12.1	12.92	4.5
141	022496	17.8	23.5	26	10.88	3.8
142	036706	38.1	11.8	12.4	5.55	0.5
143	036720	38.1	11.8	12.4	5.55	0.5
144	036460	28.6	1.8	22.9	1.16	1.6
145	036692	38.1	11.8	12.4	5.55	0.5
146	028600	21.9	17.8	16.2	6.32	2.9
147	023240	28.6	21.0	21.6	12.93	5.0
148	036494	28.6	1.8	22.9	1.16	1.6
149	022004	23.5	17.8	21.0	8.75	2.9
150	023212	28.6	21.0	21.6	12.93	5.0
151	028402	28.3	23.2	14.6	9.57	1.0
152	023746	23.5	17.8	21.0	8.75	2.8
153	029092	18.4	26.7	18.1	8.89	2.0
154	036616	14.0	16.5	14.6	3.37	0.5
155	030158	27.9	17.8	23.5	11.67	1.1
156	036324	22.9	33.0	17.1	12.94	4.5
157	023930	23.5	17.8	21.0	8.75	2.8
158	023760	23.5	17.8	21.0	8.75	2.8
159	036386	22.9	33.0	17.1	12.94	4.5

รายการ	รหัสสินค้า	ขนาดสินค้า(ซม.)			ปริมาตร (ลิตร)	น้ำหนัก (กก.)
		ยาว	กว้าง	สูง		
160	023732	23.5	17.8	21.0	8.75	2.8
161	036508	28.6	1.8	22.9	1.16	1.6
162	036296	17.1	22.6	14.0	5.4	1.6
163	030096	24.5	16.5	18.4	7.44	2.0
164	028238	20.3	18.7	20.3	7.74	1.0
165	023916	23.5	17.8	21.0	8.75	2.8
166	023882	23.5	17.8	21.0	8.75	2.8
167	023944	23.5	17.8	21.0	8.75	2.8
168	023622	23.5	17.8	21.0	8.75	2.8
169	023014	23.8	17.8	21.6	9.15	4.1
170	025008	23.5	17.8	21.0	8.75	2.9
171	036584	28.6	1.8	22.9	1.16	1.6
172	025056	23.5	17.8	21.0	8.75	2.9
173	025022	23.5	17.8	21.0	8.75	2.9
174	025070	23.5	17.8	21.0	8.75	2.9
175	036358	22.9	33.0	17.1	12.94	4.5
176	028006	23.5	29.8	25.4	17.81	1.1
177	030172	27.9	17.8	23.5	11.67	1.1
178	022434	17.8	23.5	26.0	10.88	3.8
179	036310	28.6	1.8	22.9	1.16	1.6
180	028204	20.3	18.7	20.3	7.74	1.0
181	022448	17.8	23.5	26.0	10.88	3.8

ภาคผนวก ข.
ปริมาณและน้ำหนักรถบรรทุก

ปริมาณ และน้ำหนัก บรรทุก (กิโลกรัม)	รถบรรทุก หมายเลข	ความกว้าง (ซม.)	ความยาว (ซม.)	ความสูง (ซม.)	ปริมาณ (ลิตร)
2500	1	206	494	175	17808.7
2500	2	209	483	182	18372.3
2500	3	159	312	173	8582.2
1000	4	148	205	147	4459.9
1000	5	148	205	147	4459.9
1000	6	148	205	147	4459.9
1000	7	148	205	147	4459.9
1000	8	148	205	147	4459.9
1000	9	148	205	147	4459.9
1000	10	148	205	147	4459.9
1000	11	148	205	147	4459.9
1000	12	148	205	147	4459.9
1000	13	148	205	147	4459.9
1000	14	148	205	147	4459.9
1000	15	148	205	147	4459.9
1000	16	148	205	147	4459.9
1000	17	148	205	147	4459.9
1000	18	148	205	147	4459.9

ภาคผนวก ค.
การจัดเก็บตามเขต ABC

รายการ	รหัสสินค้า	จำนวนกล่อง ต่อเดือน	เขต ABC	ข้อจำกัด การจัดเก็บ	ตำแหน่ง สินค้า
1	120124	3937	A	R1	041-E001
2	120048	3123	A	R1	041-D001
3	120308	3032	A	R1	021-C001
4	120096	2781	A	R1	020-B001
5	120294	1633	A	R1	042-E002
6	130072	1461	A	R1	039-D002
7	120062	442	B	R1	033-D008
8	120232	432	B	R1	028-C008
9	126002	405	B	F1	123-F012
10	120246	310	B	R1	084-C012
11	120260	315	B	F1	065-A015
12	120110	411	B	R1	029-C009
13	120158	370	B	R1	102-D011
14	120356	1294	A	R1	022-C002
15	122076	302	B	R1	100-D013
16	122028	281	B	R1	105-E013
17	122104	1184	A	R1	019-B002
18	120370	1140	A	R1	043-E003
19	122042	1106	A	R1	038-D003
20	120348	1003	A	R1	023-C003
21	130024	50	C	F1	177-F034

รายการ	รหัสสินค้า	จำนวนกล่อง ต่อเดือน	เขต ABC	ข้อจำกัด การจัดเก็บ	ตำแหน่ง สินค้า
22	104010	225	C	R1	107-E015
23	104086	351	B	R1	104-E012
24	104100	447	B	F1	008-A008
25	104148	497	A	R1	026-C006
26	104058	999	A	R1	044-E004
27	104162	718	A	R1	036-D005
28	104024	335	B	R1	101-D012
29	104208	374	B	R1	103-E011
30	104222	367	B	R1	082-B011
31	104256	405	B	R1	032-D009
32	010112	904	A	R1	037-D004
33	010098	768	A	R1	024-C004
34	010140	95	C	R1	155-C029
35	008528	423	B	F1	009-A009
36	008344	410	B	F1	124-F011
37	008056	387	B	F1	121-F014
38	008480	55	C	F1	178-F033
39	008514	448	B	F1	053-F008
40	008008	212	C	F1	066-A016
41	008774	201	C	F1	118-F017
42	008022	190	C	F1	067-A017
43	008268	154	C	F1	068-A018
44	008316	1264	A	F1	003-A003

รายการ	รหัสสินค้า	จำนวนกล่อง ต่อเดือน	เขต ABC	ข้อจำกัด การจัดเก็บ	ตำแหน่ง สินค้า
45	008740	416	B	F1	010-A010
46	008378	381	B	F1	064-A014
47	008726	349	B	F1	120-F015
48	008282	42	B	F1	052-F009
49	008330	461	B	F1	054-F007
50	014100	2367	A	F1	001-A001
51	014010	1429	A	F1	058-F003
52	014024	965	A	F1	004-A004
53	014086	831	A	F1	005-A005
54	014072	145	C	F1	097-D016
55	014406	116	C	F1	186-F025
56	014058	119	C	F1	126-A024
57	014208	15	C	F1	127-A025
58	014482	111	C	F1	185-F026
59	014148	111	C	F1	128-A026
60	014134	95	C	F1	182-F029
61	002020	17	C	F1	069-A019
62	072072	18	C	F1	184-F027
63	080038	3159	A	F1	060-F001
64	080052	1758	A	F1	059-F002
65	080004	1705	A	F1	002-A002
66	080080	456	B	F1	007-A007
67	056000	419	B	F1	051-F010

รายการ	รหัสสินค้า	จำนวนกล่อง ต่อเดือน	เขต ABC	ข้อจำกัด การจัดเก็บ	ตำแหน่ง สินค้า
68	056034	408	B	F1	061-A011
69	062034	890	A	F1	056-F005
70	056062	60	C	F1	179-F032
71	062048	1231	A	F1	057-F004
72	062000	828	A	F1	055-F006
73	072010	695	A	F1	006-A006
74	066206	65	C	F1	180-F031
75	066022	404	B	F1	062-A012
76	066220	400	B	F1	122-F013
77	066008	338	B	F1	063-A013
78	072024	358	B	F1	083-C011
79	072058	145	C	R1	165-D022
80	066330	95	C	R1	158-D029
81	801760	94	C	R1	138-B029
82	066316	86	C	R1	156-C030
83	008014	73	C	F1	181-F030
84	062158	158	C	F1	117-F018
85	066268	136	C	F1	115-F020
86	066282	129	C	F1	070-A020
87	062246	127	C	F1	072-A022
88	0560124	127	C	F1	188-F023
89	062218	101	C	F1	129-A027
90	062232	100	C	F1	183-F028

รายการ	รหัสสินค้า	จำนวนกล่อง ต่อเดือน	เขต ABC	ข้อจำกัด การจัดเก็บ	ตำแหน่ง สินค้า
91	030384	95	C	R1	175-E029
92	030370	99	C	R1	154-C028
93	029016	157	C	R1	146-B021
94	036288	108	C	R1	173-E027
95	036140	95	C	R1	139-B028
96	022128	139	C	R1	144-B023
97	036200	466	B	R1	014-B077
98	022080	473	B	R1	034-D007
99	022114	449	B	R1	048-E008
100	023226	428	B	R1	013-B008
101	023608	409	B	R1	049-E009
102	029044	523	A	R1	016-B005
103	030416	502	A	R1	035-D006
104	030430	486	B	R1	047-E007
105	028054	473	B	R1	027-C007
106	022066	106	C	R1	160-D027
107	036644	105	C	R1	163-C027
108	030034	89	C	R1	157-D030
109	030000	128	C	R1	162-D025
110	029078	117	C	R1	161-D026
111	022052	144	C	R1	145-B022
112	036678	143	C	R1	169-E023
113	036262	485	A	R1	015-B006



รายการ	รหัสสินค้า	จำนวนกล่อง ต่อเดือน	เขต ABC	ข้อจำกัด การจัดเก็บ	ตำแหน่ง สินค้า
114	036630	508	A	R1	046-E006
115	023868	602	A	R1	025-C005
116	036036	742	A	R1	017-B004
117	030370	1002	A	R1	018-B003
118	030384	720	A	R1	045-E005
119	028024	395	B	R1	031-D010
120	029030	394	B	R1	030-C010
121	030062	385	B	R1	011-D010
122	036248	282	B	R1	085-C013
123	028068	141	C	R1	164-D023
124	022038	125	C	R1	142-B025
125	036002	112	C	R1	152-C026
126	036556	94	C	R1	176-E030
127	036522	80	C	R1	137-B030
128	022510	122	C	R1	172-E026
129	030048	158	C	R1	166-D021
130	022052	247	B	R1	080-B013
131	022142	312	B	R1	081-B012
132	036372	402	B	R1	012-D009
133	036050	401	B	R1	050-E010
134	030416	139	C	R1	149-C023
135	036234	136	C	R1	170-E024
136	023028	135	C	R1	150-C024

รายการ	รหัสสินค้า	จำนวนกล่อง ต่อเดือน	เขต ABC	ข้อจำกัด การจัดเก็บ	ตำแหน่ง สินค้า
137	022462	235	C	R1	106-E014
138	030430	234	C	R1	086-C014
139	036570	232	C	R1	099-D014
140	036064	228	C	R1	079-D014
141	022496	222	C	R1	098-D015
142	036706	215	C	R1	087-C015
143	036720	214	C	R1	078-B015
144	036460	214	C	R1	108-E016
145	036692	210	C	R1	116-F019
146	028600	210	C	R1	088-C016
147	023240	205	C	R1	077-B016
148	036494	203	C	R1	109-E017
149	022004	202	C	R1	096-D017
150	023212	201	C	R1	089-C017
151	028402	200	C	R1	076-B017
152	023746	197	C	R1	110-E018
153	029092	196	C	R1	095-D018
154	036616	188	C	R1	090-C018
155	030158	182	C	R1	075-D018
156	036324	181	C	R1	111-E019
157	023930	177	C	R1	094-D019
158	023760	177	C	R1	091-C019
159	036386	176	C	R1	074-B019

รายการ	รหัสสินค้า	จำนวนกล่อง ต่อเดือน	เขต ABC	ข้อจำกัด การจัดเก็บ	ตำแหน่ง สินค้า
160	023732	109	C	R1	141-B026
161	036508	169	C	R1	112-E020
162	036296	169	C	R1	093-D020
163	030096	166	C	R1	092-C020
164	028238	104	C	R1	174-E028
165	023916	159	C	R1	073-B020
166	023882	159	C	R1	167-E021
167	023944	157	C	R1	147-C021
168	023622	105	C	R1	140-D027
169	023014	100	C	R1	159-D028
170	025008	146	C	R1	168-E022
171	036584	144	C	R1	148-C022
172	025056	127	C	R1	151-C025
173	025022	136	C	R1	163-D024
174	025070	134	C	R1	143-B024
175	036358	134	C	R1	171-E025
176	028006	128	C	F1	113-F022
177	030172	128	C	F1	071-A021
178	022434	218	C	F1	119-F016
179	036310	128	C	F1	114-F021
180	028204	126	C	F1	187-F024
181	022448	123	C	F1	125-A023

ภาคผนวก ง.

เขตการขนส่ง

1. เขตการขนส่งกรุงเทพมหานคร

รหัสเขตการขนส่ง	ชื่อเขตการขนส่ง
001	บางรัก คลองเตย พระราม 4 ทรอกจันทน์
002	ถนนตก พระราม 3 สาทร ลีพระยา
003	ประตูน้ำ เพลินจิต ร่มโพธิ์ ราชเทวี รัชดาภิเษก
004	สุทธิสาร พระราม 9 หัวขวง สะพานควาย
005	มหานาค งามวงศ์วาน แจ้งวัฒนะ
006	นนทบุรี ปากเกร็ด เทเวศร์ หลักสี่
007	เขาวราช พหลโยธิน ปากคลองตลาด เจริญผล
008	พระราม 6 นางเลิ้ง บางลำพู บางซื่อ
009	พหลโยธิน จตุจักร ดอนเมือง รังสิต
010	ปทุมธานี ลำลูกกา ปากทางลาดพร้าว
011	สุขุมวิท อโศก-ดินแดง อุดมสุข อ่อนนุช บางจาก
012	บางนา สมุทรปราการ ลำโพง บางพลี
013	หัวหมาก คลองตัน พัฒนาการ บางกะปิ โชคชัย 4
014	รามอินทรา มีนบุรี รามคำแหง
015	ฝั่งขวา บุคคโล ท่าพระ บางขุนเทียน หนองแขม
016	พุทธมณฑล ดลิ่งชัน ปิ่นเกล้า เพชรเกษม

2. เขตการขนส่งภาคกลาง

เขตการขนส่ง	ชื่อเขตการขนส่ง	เขตการขนส่ง	ชื่อเขตการขนส่ง
501	อ่างทอง	527	ท่ายาง
502	อยุธยา	528	ปราจีนบุรี
503	ฉะเชิงเทรา	529	กบินทร์บุรี
504	พนมสารคาม	530	ประจวบคีรีขันธ์
505	บางคล้า	531	ทับสะแก
506	บางปะกง	532	บางสะพาน
507	ชัยนาท	533	ราชบุรี
508	จันทบุรี	534	บ้านโป่ง
509	ท่าใหม่	535	โพธาราม
510	ชลบุรี	536	ดำเนินสะดวก
511	ศรีราชา	537	ระยอง
512	พัตยา	538	บ้านฉาง
513	สัตหีบ	539	แก่ง
514	บ้านบึง	540	สมุทรสาคร
515	กาญจนบุรี	540	สมุทรสงคราม
516	ท่ามะกา	542	สระบุรี
517	ลพบุรี	543	พระพุทธบาท
518	ชัยบาดาล	544	สิงห์บุรี
519	โคกสำโรง	545	สุพรรณบุรี
520	ลำนาทรายณ์	546	อุทอง
521	นครนายก	547	ดอนเจดีย์
522	นครปฐม	548	สุพรรณบุรี
523	กำแพงแสน	549	ตราด
524	เพชรบุรี	550	สระแก้ว
525	ปราณบุรี	551	อรัญประเทศ
526	กุยบุรี	552	อุทัยธานี



ภาคผนวก จ.

ชุดคำสั่งภาษา Visual Basics

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

File Name : GetDate.Frm

Option Explicit

Global Const MAX_ORDERLINE As Integer = 26

Global Const MAX_TRANSFER_PER_PAGE As Integer = 6

Global Const MAX_PRODUCT_PER_PAGE As Integer = 10

Global Const MAX_INVOICE_PER_PAGE As Integer = 12

Global Const DATABASE_FILE As String = "Prodinfo.mdb"

Global Const MAXINVOICE_PER_PAGE As Integer = 12

Public ProgramDir As String

Public OrderDir As String

Public DateAns As String

Public IsCancel As Boolean

Public Type OrderStruc

CustomerID As String

CustomerName As String

CustomerAddr As String

DistrictID As String

ProductID(25) As String

ProductName(25) As String

ProductCount(25) As String

End Type

Public Type ValidProductStruc

CustomerID As String

CustomerName As String

CustomerAddr As String

DistrictID As String

```
ProductID() As String
ProductName() As String
ProductCount() As String
ProductCnt As Integer
End Type
```

```
Public Type InvoiceStruc
```

```
OrderNo As Integer
InvoiceID As String
InvoiceDate As String
TransferID As String
TransferName As String
CustomerName As String
TotalVolume As String
TotalMass As String
End Type
```

```
Public Type TransferStruc
```

```
InvoiceID As String
TransferID As String
OrderNo() As Integer
TransferName As String
CustomerName As String
TotalVolume As String
TransferCount As Integer
End Type
```

```
Public Type TakeProductStruc
```

```
DriverName As String
TruckID As String
```



```
TruckNumber As String
OrderNo() As Integer
OrderCount As Integer
InvoiceCount As Integer
InvoiceID() As String
TakeProductID As String
ProductPosition() As String
ProductID() As String
ProductName() As String
ProductCount() As String
ProductCnt As Integer
End Type
```

```
Function CThaiDate(ByVal mDate As Date) As String
    Dim mDay As Integer
    Dim mMonth As Integer
    Dim mYear As Integer

    mYear = Year(mDate) + 543
    mMonth = Month(mDate)
    mDay = Day(mDate)
    CThaiDate = Format(DateSerial(mYear, mMonth, mDay), "Long Date")
End Function
```

File Name: GetDate.Frm

Option Explicit

Private Sub Combo1_DblClick()

 Command1_Click

End Sub

Private Sub Command1_Click()

 Dim TmpDate As Date

 Dim mDay As Integer, mMonth As Integer, mYear As Integer

 TmpDate = CDate(Combo1.List(Combo1.ListIndex))

 mYear = Year(TmpDate)

 mMonth = Month(TmpDate)

 mDay = Day(TmpDate)

 TmpDate = DateSerial(mYear, mMonth, mDay)

 DateAns = Format(TmpDate, "dd/mm/yy")

 IsCancel = False

 Unload Form2

End Sub

Private Sub Command2_Click()

 IsCancel = True

 Unload Form2

End Sub

Private Sub Form_Load()

```
Dim i As Integer
Dim FileName As String
Dim FName As String, temp As String
Combo1.Clear
FileName = Dir(ProgramDir & "*.ord")
Do Until FileName = ""

    FName = Right(FileName, 10)
    FName = Left(FName, 6)
    temp = Mid(FName, 3, 2)
    temp = temp & "/" & Left(FName, 2)
    temp = temp & "/" & Right(FName, 2)
    FName = CThaiDate(CDate(temp))
    Combo1.AddItem FName
    FileName = Dir()
Loop
Combo1.ListIndex = 1
End Sub
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

File Name : Order.frm

Option Explicit

Dim OrderAry() As OrderStruc

Dim InvoiceAry() As InvoiceStruc

Dim SortByDistrictInvAry() As InvoiceStruc

Dim TransferAry() As TransferStruc

Dim TakeProductAry() As TakeProductStruc

Dim ValidProductAry() As ValidProductStruc

Dim TruckID As String

Dim TruckNumber As String

Dim DriverName As String

Dim CurTransferCount As Integer

Dim ValidProductCnt As Integer

Dim CurTakeProductID As Long

Dim InvoiceCount As Integer

Dim TransferCount As Integer

Dim TakeProductCount As Integer

Dim OrderCount As Integer

Dim CurTakeProduct As Integer

Dim CurrentTransfer As Integer

Dim CurrentOrder As Integer

Dim CurRowInvoice As Integer

Dim CurTruckVol As Double

Dim CurProductVol As Double

Dim CurTruckMass As Double

```
Dim CurProductMass As Double
```

```
Dim DateOfOrder As Date
```

```
Function AllVolumeOfOrder(OrderID As Integer, AllWeight As Double) As Double
```

```
    Dim db As Database
```

```
    Dim ProductRset As Recordset
```

```
    Dim SQLStr As String
```

```
    Dim ListProduct As String
```

```
    Dim AllVol As Double
```

```
    Dim i As Integer
```

```
    ' Store product which valid data
```

```
    ReDim Preserve ValidProductAry(OrderID + 1) As ValidProductStruc
```

```
    ValidProductAry(OrderID).ProductCnt = 0
```

```
    ' Initialize volume
```

```
    AllWeight = 0
```

```
    AllVol = 0
```

```
    Set db = Workspaces(0).OpenDatabase(ProgramDir & "prodinfo.mdb", False, False)
```

```
    For i = 0 To MAX_ORDERLINE - 1
```

```
        ' check data of product valid
```

```
        If Len(OrderAry(OrderID).ProductID(i)) = 6 And _
```

```
            IsNumeric(OrderAry(OrderID).ProductID(i)) And _
```

```
            IsNumeric(OrderAry(OrderID).ProductCount(i)) Then
```

```
            SQLStr = "SELECT ProductID, Volume, Weight FROM ProductInfo " & _
```

```
                " WHERE ProductID = " & _
```

```
                OrderAry(OrderID).ProductID(i) & "' " & _
```

```
                " ORDER BY ProductID"
```

```
            Set ProductRset = db.OpenRecordset(SQLStr, dbOpenDynaset, dbReadOnly)
```

```
' not found product id in database
```

```
If ProductRset.EOF = True Then
```

```
    ' keep product in string which display on message box
```

```
    ListProduct = ListProduct & OrderAry(OrderID).ProductID(i) & Chr$(13) & Chr$(10)
```

```
    GoTo NextFor
```

```
End If
```

```
' calculate all volume of product
```

```
AllVol = AllVol + ProductRset.Fields(1).Value * OrderAry(OrderID).ProductCount(i)
```

```
AllWeight = AllWeight + ProductRset.Fields(2).Value *
```

```
OrderAry(OrderID).ProductCount(i)
```

```
ProductRset.Close
```

```
' store in array
```

```
ReDim Preserve ValidProductAry(OrderID).ProductID( _
```

```
ValidProductAry(OrderID).ProductCnt + 1) As String
```

```
ReDim Preserve ValidProductAry(OrderID).ProductName( _
```

```
ValidProductAry(OrderID).ProductCnt + 1) As String
```

```
ReDim Preserve ValidProductAry(OrderID).ProductCount( _
```

```
ValidProductAry(OrderID).ProductCnt + 1) As String
```

```
ValidProductAry(OrderID).ProductID(ValidProductAry(OrderID).ProductCnt) = _
```

```
OrderAry(OrderID).ProductID(i)
```

```
ValidProductAry(OrderID).ProductName(ValidProductAry(OrderID).ProductCnt) = _
```

```
OrderAry(OrderID).ProductName(i)
```

```
ValidProductAry(OrderID).ProductCount(ValidProductAry(OrderID).ProductCnt) = _
```

```
OrderAry(OrderID).ProductCount(i)
```

```
ValidProductAry(OrderID).ProductCnt = ValidProductAry(OrderID).ProductCnt + 1
```

```
Elseif OrderAry(OrderID).ProductID(i) <> "" Then
```

```
    ListProduct = ListProduct & OrderAry(OrderID).ProductID(i) & Chr$(13) & Chr$(10)
```

```

End If

NextFor:
Next i

db.Close
AllVolumeOfOrder = AllVol
If ListProduct <> "" Then
    MsgBox "ไม่มีรหัสสินค้าต่อไปสำหรับ Invoice ลำดับที่ " & CStr(OrderID) & Chr$(13) &
    Chr$(10) & _
    ListProduct, vbOKOnly, "รหัสสินค้าที่ไม่มีอยู่ในฐานข้อมูลระบบ"
End If

End Function

Sub FillControlWithVariable(OrderNo As Integer)
    Dim i As Integer

    TxtCustomerID.Text = OrderAry(OrderNo).CustomerID
    TxtCustomerName.Text = OrderAry(OrderNo).CustomerName
    TxtCustomerAddr.Text = OrderAry(OrderNo).CustomerAddr
    TxtDistrictID.Text = OrderAry(OrderNo).DistrictID
    For i = 0 To MAX_ORDERLINE - 1
        Text5(i).Text = OrderAry(OrderNo).ProductID(i)
        Text6(i).Text = OrderAry(OrderNo).ProductName(i)
        Text7(i).Text = OrderAry(OrderNo).ProductCount(i)
    Next i

End Sub

```

Sub FillTakeProductControl(InvoiceNo As Integer)

On Error GoTo Err_Handler

If CurRowInvoice < MAX_TRANSFER_PER_PAGE And _

CurRowInvoice >= 0 Then

txtInvoiceID(CurRowInvoice).Text = InvoiceAry(InvoiceNo).InvoiceID

txtTransferID(CurRowInvoice).Text = InvoiceAry(InvoiceNo).TransferID

txtTransferName(CurRowInvoice).Text = InvoiceAry(InvoiceNo).TransferName

txtCustName(CurRowInvoice).Text = InvoiceAry(InvoiceNo).CustomerName

txtVolume(CurRowInvoice).Text = InvoiceAry(InvoiceNo).TotalVolume

End If

' Store in Transfer Array

VScroll2.Max = (Int((TransferCount - 1) / MAX_TRANSFER_PER_PAGE))

TransferAry(TransferCount - 1).InvoiceID = InvoiceAry(InvoiceNo).InvoiceID

ReDim Preserve TransferAry(TransferCount - 1).OrderNo(TransferCount + 1) As Integer

TransferAry(TransferCount - 1).OrderNo(TransferCount) = InvoiceAry(InvoiceNo).OrderNo

TransferAry(TransferCount - 1).TransferCount = TransferCount

TransferAry(TransferCount - 1).TransferID = InvoiceAry(InvoiceNo).TransferID

TransferAry(TransferCount - 1).TransferName = InvoiceAry(InvoiceNo).TransferName

TransferAry(TransferCount - 1).CustomerName = InvoiceAry(InvoiceNo).CustomerName

TransferAry(TransferCount - 1).TotalVolume = InvoiceAry(InvoiceNo).TotalVolume

Text11.Text = Format(CurProductVol + CDbI(InvoiceAry(InvoiceNo).TotalVolume), "##0.00")

Text15.Text = Format(CurProductMass + CDbI(InvoiceAry(InvoiceNo).TotalMass), "##0.00")

UpdateVolumeDiff

UpdateMassDiff

CurRowInvoice = CurRowInvoice + 1

Exit Sub

Err_Handler:

Text12.Text = ""


```
Text13.Text = ""
```

```
Resume Next
```

```
End Sub
```

```
Sub FillVariableWithControlValue(OrderNo As Integer)
```

```
Dim i As Integer
```

```
Dim a As Single
```

```
OrderAry(OrderNo).CustomerID = TxtCustomerID.Text
```

```
OrderAry(OrderNo).CustomerName = TxtCustomerName.Text
```

```
OrderAry(OrderNo).CustomerAddr = TxtCustomerAddr.Text
```

```
OrderAry(OrderNo).DistrictID = TxtDistrictID.Text
```

```
For i = 0 To MAX_ORDERLINE - 1
```

```
OrderAry(OrderNo).ProductID(i) = Text5(i).Text
```

```
OrderAry(OrderNo).ProductName(i) = Text6(i).Text
```

```
OrderAry(OrderNo).ProductCount(i) = Text7(i).Text
```

```
Next i
```

```
End Sub
```

```
Function FindProductPos(ProductID As String)
```

```
Dim db As Database
```

```
Dim ProdPosInf As Recordset
```

```
Dim SQLStr As String, temp As String
```

```
Dim i As Integer
```

```
Dim AStr As String
```

```
Set db = Workspaces(0).OpenDatabase(ProgramDir & DATABASE_FILE, False, False)
```

```
SQLStr = "SELECT ProductID, ProductPosition " & _
```

```
"FROM ProductPosition " & _
"WHERE ProductID = "" & ProductID & """
```

```
Set ProdPosInf = db.OpenRecordset(SQLStr)
```

```
If ProdPosInf.EOF = True Then
```

```
    Exit Function
```

```
End If
```

```
temp = ProdPosInf.Fields(1).Value
```

```
ProdPosInf.Close
```

```
db.Close
```

```
FindProductPos = temp
```

```
End Function
```

```
Function GetDateOfOrder() As String
```

```
Form2.Show vbModal
```

```
If IsCancel = True Then
```

```
    Exit Function
```

```
End If
```

```
If IsDate(DateAns) Then
```

```
    GetDateOfOrder = Format(CDate(DateAns), "dd/mm/yy")
```

```
Else
```

```
    MsgBox "ข้อมูลวันที่ไม่ถูกต้อง", vbOKOnly, "ข้อมูลผิดพลาด"
```

```
End If
```

```
End Function
```

```

Sub GetTruckInfo(TruckNo As Integer, _
                Weight As Double, _
                Volume As Double)

    Dim db As Database
    Dim TruckRset As Recordset
    Dim SQLStr As String
    On Error Resume Next
    Set db = Workspaces(0).OpenDatabase(ProgramDir & _
        DATABASE_FILE, False, False)
    SQLStr = "SELECT TruckID, Volume, Mass " & _
        "FROM TruckInfo " & _
        "WHERE TruckID = " & CStr(TruckNo)
    Set TruckRset = db.OpenRecordset(SQLStr, dbOpenDynaset, dbReadOnly)
    Weight = TruckRset.Fields(2).Value
    weight = truckrset.fields(
    Volume = TruckRset.Fields(1).Value

End Sub

Sub InitControl()
    Dim i As Integer
    SStab1.Tab = 0
    TxtCustomerID.Text = ""
    TxtCustomerName.Text = ""
    TxtCustomerAddr.Text = ""
    TxtDistrictID.Text = ""
    LblOrderDate.Caption = CThaiDate(Now)
    For i = 0 To 12
        Text1(i).Text = ""
        Text2(i).Text = ""
    
```

```
Text3(i).Text = ""
Text4(i).Text = ""
Text8(i).Text = ""
Text9(i).Text = ""
Next i
For i = 0 To MAX_ORDERLINE - 1
    Text5(i).Text = ""
    Text6(i).Text = ""
    Text7(i).Text = ""
Next i
Text10.Text = ""
Text11.Text = ""
Text12.Text = ""
Text13.Text = ""
Text14.Text = ""
Text15.Text = ""
Text16.Text = ""
Text17.Text = ""
Text18(0).Text = ""
Text18(1).Text = ""
Text19(0).Text = ""
Text19(1).Text = ""
Text20(0).Text = ""
Text20(1).Text = ""
For i = 0 To 5
    txtInvoiceID(i).Text = ""
    txtTransferID(i).Text = ""
    txtTransferName(i).Text = ""
    txtCustName(i).Text = ""
    txtVolume(i).Text = ""
Next i
```

End Sub

Sub InitMenu()

MnuNewOrder.Enabled = True
MnuOpenOrder.Enabled = True
MnuSaveOrder.Enabled = False
MnuCloseOrder.Enabled = False
MnuOpenOrder.Enabled = True
MnuSaveOrder.Enabled = False
MnuCloseOrder.Enabled = False
MnuPrintOrder.Enabled = False

End Sub

Sub InitMenuWhenOrderExist()

MnuNewOrder.Enabled = False
MnuOpenOrder.Enabled = False
MnuSaveOrder.Enabled = True
MnuCloseOrder.Enabled = True

MnuShowAllInvoice.Enabled = True
MnuShowInvoice.Enabled = True
MnuPrintInvoice.Enabled = True

MnuShowTransferProduct.Enabled = False
MnuShowTakeProduct.Enabled = False
MnuPrintProduct = False
MnuPrintTransfer.Enabled = False
MnuPrintTakeProduct.Enabled = False

End Sub

Sub InitOrderVariable()

Dim i As Integer, j As Integer

ReDim Preserve OrderAry(1) As OrderStruc

CurrentOrder = 0

OrderCount = 1

DateOfOrder = CDate(Now)

For i = 0 To OrderCount - 1

OrderAry(i).CustomerID = ""

OrderAry(i).CustomerName = ""

OrderAry(i).CustomerAddr = ""

OrderAry(i).DistrictID = ""

For j = 0 To MAX_ORDERLINE - 1

OrderAry(i).ProductID(j) = ""

OrderAry(i).ProductName(j) = ""

OrderAry(i).ProductCount(j) = ""

Next j

Next i

End Sub

Sub InitTransferControl()

Dim i As Integer

List2(0).Clear

List1(0).Clear

For i = 0 To InvoiceCount - 1

```
List2(0).AddItem InvoiceAry(i).InvoiceID
Next i
Text10.Text = ""
Text11.Text = ""
Text12.Text = ""
Text13.Text = ""
Text14.Text = ""
Text15.Text = ""
Text16.Text = ""
Text17.Text = ""
Text18(0).Text = ""
Text18(1).Text = ""
Text19(0).Text = ""
Text19(1).Text = ""
Text20(0).Text = ""
Text20(1).Text = ""
Text20(2).Text = ""
For i = 0 To 5
    txtInvoiceID(i).Text = ""
    txtTransferID(i).Text = ""
    txtTransferName(i).Text = ""
    txtCustName(i).Text = ""
    txtVolume(i).Text = ""
Next i
End Sub

Sub InitTakeProductControl()
    Dim i As Integer

    'List1(0).Clear
    'Text10.Text = ""
```

```
'Text11.Text = ""
'Text12.Text = ""
'Text13.Text = ""
'Text14.Text = ""
'Text15.Text = ""
'Text16.Text = ""
'Text17.Text = ""
'Text18(0).Text = ""
Text18(1).Text = ""
'Text19(0).Text = ""
Text19(1).Text = ""
'Text20(0).Text = ""
Text20(1).Text = ""
Text20(2).Text = ""
For i = 6 To 15
```

```
    txtTransferID(i).Text = ""
    txtTransferName(i).Text = ""
    txtCustName(i).Text = ""
    txtVolume(i).Text = ""
```

```
Next i
```

```
End Sub
```

```
Sub InitTransferVar()
```

```
    CurRowInvoice = 0
    TransferCount = 0
    CurTruckVol = 0#
    CurProductVol = 0#
    CurTruckMass = 0#
    CurProductMass = 0#
```


End Sub

Sub InitTakeProductVar()

TakeProductCount = 0

CurTakeProduct = 0

End Sub

Sub MakeAllControlNotVisible()

Dim i As Integer

For i = 0 To 1

Label1(i).Visible = False

Label2(i).Visible = False

Label3(i).Visible = False

Label4(i).Visible = False

Next i

Label5.Visible = False

Label6.Visible = False

Label7.Visible = False

Label8.Visible = False

Label9.Visible = False

Label10.Visible = False

Label11.Visible = False

LblOrderDate.Visible = False

TxtCustomerID.Visible = False

TxtCustomerName.Visible = False

TxtCustomerAddr.Visible = False

```
TxtDistrictID.Visible = False
```

```
BtnInsert(0).Visible = False
```

```
BtnNext(0).Visible = False
```

```
BtnPrevious(0).Visible = False
```

```
For i = 0 To MAX_ORDERLINE - 1
```

```
Text5(i).Visible = False
```

```
Text6(i).Visible = False
```

```
Text7(i).Visible = False
```

```
Next i
```

```
SSTab1.Visible = False
```

```
End Sub
```

```
Sub MakeAllControlVisible()
```

```
Dim i As Integer
```

```
For i = 0 To 1
```

```
Label1(i).Visible = True
```

```
Label2(i).Visible = True
```

```
Label3(i).Visible = True
```

```
Label4(i).Visible = True
```

```
Next i
```

```
Label5.Visible = True
```

```
Label6.Visible = True
```

```
Label7.Visible = True
```

```
Label8.Visible = True
```

```
Label9.Visible = True
```

```
Label10.Visible = True
```

```
Label11.Visible = True
```

```
LblOrderDate.Visible = True
```

```

TxtCustomerID.Visible = True
TxtCustomerName.Visible = True
TxtCustomerAddr.Visible = True
TxtDistrictID.Visible = True

BtnInsert(0).Visible = True
BtnNext(0).Visible = True
BtnPrevious(0).Visible = True

For i = 0 To MAX_ORDERLINE - 1
    Text5(i).Visible = True
    Text6(i).Visible = True
    Text7(i).Visible = True
Next i
End Sub

Function OpenOrder(DateOfOrder As String) As Boolean

    Dim FileNo As Integer
    Dim LineChr As String
    Dim i As Integer, j As Integer
    Dim OrdCount As Integer

    On Error GoTo Err_Handler

    FileNo = FreeFile(0)
    Open OrderDir & DateOfOrder & ".ord" For Input As #FileNo

    Do Until EOF(FileNo)

        Line Input #FileNo, LineChr

```

```

OrdCount = CInt(Mid(LineChr, InStr(LineChr, "=") + 1))
i = OrdCount
ReDim Preserve OrderAry(OrdCount + 1) As OrderStruc

Line Input #FileNo, LineChr
OrderAry(i).CustomerID = Trim(Mid(LineChr, InStr(LineChr, "=") + 1))
Line Input #FileNo, LineChr
OrderAry(i).CustomerName = Trim(Mid(LineChr, InStr(LineChr, "=") + 1))
Line Input #FileNo, LineChr
OrderAry(i).CustomerAddr = Trim(Mid(LineChr, InStr(LineChr, "=") + 1))
Line Input #FileNo, LineChr
OrderAry(i).DistrictID = Trim(Mid(LineChr, InStr(LineChr, "=") + 1))

For j = 0 To MAX_ORDERLINE - 1
    Line Input #FileNo, LineChr
    OrderAry(i).ProductID(j) = Trim(Mid(LineChr, InStr(LineChr, "=") + 1))
    Line Input #FileNo, LineChr
    OrderAry(i).ProductName(j) = Trim(Mid(LineChr, InStr(LineChr, "=") + 1))
    Line Input #FileNo, LineChr
    OrderAry(i).ProductCount(j) = Trim(Mid(LineChr, InStr(LineChr, "=") + 1))
Next j

Loop

OrderCount = OrdCount + 1
Close #FileNo
OpenOrder = True
Exit Function
Err_Handler:
    OpenOrder = False

End Function

```

```
Function SaveOrder(OrderFile As String) As Boolean
```

```
    Dim FileNo As Integer
```

```
    Dim i As Integer
```

```
    Dim j As Integer
```

```
    On Error GoTo Err_Handler
```

```
    FileNo = FreeFile(0)
```

```
    Open OrderDir & OrderFile For Output As #FileNo
```

```
    For i = 0 To OrderCount - 1
```

```
        Print #FileNo, "Order = " & CStr(i)
```

```
        Print #FileNo, "CustomerID = " & OrderAry(i).CustomerID
```

```
        Print #FileNo, "CustomerName = " & OrderAry(i).CustomerName
```

```
        Print #FileNo, "CustomerAddr = " & OrderAry(i).CustomerAddr
```

```
        Print #FileNo, "DistrictID = " & OrderAry(i).DistrictID
```

```
        For j = 0 To MAX_ORDERLINE - 1
```

```
            Print #FileNo, "ProductID = " & OrderAry(i).ProductID(j)
```

```
            Print #FileNo, "ProductName = " & OrderAry(i).ProductName(j)
```

```
            Print #FileNo, "ProductCount = " & OrderAry(i).ProductCount(j)
```

```
        Next j
```

```
    Next i
```

```
    Close #FileNo
```

```
    SaveOrder = True
```

```
    Exit Function
```

```
Err_Handler:
```

```
    SaveOrder = False
```

```
End Function
```

```
Sub UpdateVolumeDiff()
```

```
    On Error GoTo Err_Handler
```

```
    If CurTruckVol = 0 Or CurProductVol = 0 Then
```

```
        Exit Sub
```

```
    End If
```

```
    Text12.Text = Format((CurTruckVol - CurProductVol), "##0.00")
```

```
    If CDbI(Text12.Text) < 0# Then
```

```
        Text12.ForeColor = &HFF
```

```
    Else
```

```
        Text12.ForeColor = &HFF0000
```

```
    End If
```

```
    Text13.Text = Format((CurTruckVol - CurProductVol) * 100 / CurTruckVol, "##0.00")
```

```
    If CDbI(Text13.Text) < 0# Then
```

```
        Text13.ForeColor = &HFF
```

```
    Else
```

```
        Text13.ForeColor = &HFF0000
```

```
    End If
```

```
    Exit Sub
```

```
Err_Handler:
```

```
    Text12.Text = ""
```

```
    Text13.Text = ""
```

```
    Resume Next
```

```
End Sub
```

```
Sub UpdateMassDiff()
```

```
    On Error GoTo Err_Handler
```

```
    If CurTruckMass = 0 Or CurProductMass = 0 Then
```

```

Exit Sub
End If
Text16.Text = Format((CurTruckMass - CurProductMass), "##0.00")
If CDb1(Text16.Text) < 0# Then
    Text16.ForeColor = &HFF
Else
    Text16.ForeColor = &HFF0000
End If

Text17.Text = Format((CurTruckMass - CurProductMass) * 100 / CurTruckMass, "##0.00")
If CDb1(Text17.Text) < 0# Then
    Text17.ForeColor = &HFF
Else
    Text17.ForeColor = &HFF0000
End If
Exit Sub
Err_Handler:
Text16.Text = ""
Text17.Text = ""
Resume Next
End Sub

Private Sub BtnInsert_Click(Index As Integer)
    Dim i As Integer, j As Integer

    If Index = 0 Then
        FillVariableWithControlValue (CurrentOrder)
        OrderCount = OrderCount + 1
        ReDim Preserve OrderAry(OrderCount) As OrderStruc
        CurrentOrder = OrderCount - 1
        InitControl
        TxtCustomerID.SetFocus
    
```

Elseif Index = 1 Then

With TakeProductAry(CurTakeProduct)

For i = 0 To .InvoiceCount - 1

List2(0).AddItem .InvoiceID(i)

Next i

End With

For i = CurTakeProduct To TakeProductCount - 2

TakeProductAry(i).DriverName = TakeProductAry(i + 1).DriverName

TakeProductAry(i).TruckID = TakeProductAry(i + 1).TruckID

TakeProductAry(i).TruckNumber = TakeProductAry(i + 1).TruckNumber

TakeProductAry(i).OrderCount = TakeProductAry(i + 1).OrderCount

TakeProductAry(i).InvoiceCount = TakeProductAry(i + 1).InvoiceCount

TakeProductAry(i).TakeProductID = TakeProductAry(i + 1).TakeProductID

TakeProductAry(i).ProductCnt = TakeProductAry(i + 1).ProductCnt

ReDim Preserve TakeProductAry(i).OrderNo(TakeProductAry(i).OrderCount) As Integer

ReDim Preserve TakeProductAry(i).InvoiceID(TakeProductAry(i).InvoiceCount) As String

ReDim Preserve TakeProductAry(i).ProductID(TakeProductAry(i).ProductCnt) As String

ReDim Preserve TakeProductAry(i).ProductName(TakeProductAry(i).ProductCnt) As

String

ReDim Preserve TakeProductAry(i).ProductPosition(TakeProductAry(i).ProductCnt) As

String

ReDim Preserve TakeProductAry(i).ProductCount(TakeProductAry(i).ProductCnt) As

String

For j = 0 To TakeProductAry(i).OrderCount - 1

TakeProductAry(i).OrderNo(j) = TakeProductAry(i + 1).OrderNo(j)

Next j

For j = 0 To TakeProductAry(i).InvoiceCount - 1

TakeProductAry(i).InvoiceID(j) = TakeProductAry(i + 1).InvoiceID(j)

Next j



```
For j = 0 To TakeProductAry(i).ProductCnt - 1
    TakeProductAry(i).ProductID(j) = TakeProductAry(i + 1).ProductID(j)
    TakeProductAry(i).ProductName(j) = TakeProductAry(i + 1).ProductName(j)
    TakeProductAry(i).ProductPosition(j) = TakeProductAry(i + 1).ProductPosition(j)
    TakeProductAry(i).ProductCount(j) = TakeProductAry(i + 1).ProductCount(j)
Next j

Next i

ReDim Preserve TakeProductAry(TakeProductCount - 1) As TakeProductStruc
ReDim Preserve TakeProductAry(i).OrderNo(TakeProductAry(i).OrderCount) As Integer
ReDim Preserve TakeProductAry(i).InvoiceID(TakeProductAry(i).InvoiceCount) As String
ReDim Preserve TakeProductAry(i).ProductID(TakeProductAry(i).ProductCnt) As String
ReDim Preserve TakeProductAry(i).ProductName(TakeProductAry(i).ProductCnt) As
String
    ReDim Preserve TakeProductAry(i).ProductPosition(TakeProductAry(i).ProductCnt) As
String
    ReDim Preserve TakeProductAry(i).ProductCount(TakeProductAry(i).ProductCnt) As
String
    TakeProductCount = TakeProductCount - 1
    CurTakeProduct = 0
    InitTakeProductControl
    FillControlWithTransferVariable (CurTakeProduct)
    VScroll3.Max = Int((TakeProductAry(CurTakeProduct).ProductCnt - 1) _
        / MAX_PRODUCT_PER_PAGE)
    VScroll3.Value = 0
End If

End Sub

Private Sub BtnNext_Click(Index As Integer)
```

```

If Index = 0 Then
    If CurrentOrder >= OrderCount - 1 Then
        MsgBox "ไม่มีรายการต่อไป", vbOKOnly + vbInformation, "ข้อผิดพลาด"
    Exit Sub
    End If
    FillVariableWithControlValue (CurrentOrder)
    CurrentOrder = CurrentOrder + 1
    FillControlWithVariable (CurrentOrder)

```

```

Elseif Index = 1 Then
    If CurTakeProduct >= TakeProductCount - 1 Then
        MsgBox "ไม่มีรายการต่อไป", vbOKOnly + vbInformation, "ข้อผิดพลาด"
    Exit Sub
    End If
    InitTakeProductControl
    CurTakeProduct = CurTakeProduct + 1
    FillControlWithTransferVariable (CurTakeProduct)
    VScroll3.Max = Int((TakeProductAry(CurTakeProduct).ProductCnt - 1) _
        / MAX_PRODUCT_PER_PAGE)
    VScroll3.Value = 0
    End If

```

```
End Sub
```

```

Sub FillVariableWithTransferControlValue(TransferNo As Integer)
    Dim i As Integer
    TakeProductAry(CurTakeProductID).TruckID = Text18(1).Text
    TakeProductAry(CurTakeProductID).DriverName = Text20(1).Text
    TakeProductAry(CurTakeProductID).TakeProductID = Text20(2).Text
End Sub

```

```
Sub FillControlWithTransferVariable(TakeProductNo As Integer)
```

```
    Dim i As Integer
```

```
    Dim j As Integer
```

```
    With TakeProductAry(TakeProductNo)
```

```
        Text18(1).Text = .TruckID
```

```
        Text19(1).Text = .TruckNumber
```

```
        Text20(1).Text = .DriverName
```

```
    Text18(1).Text = .TruckID
```

```
    Text19(1).Text = .TruckNumber
```

```
    Text20(1).Text = .DriverName
```

```
    Text20(2).Text = .TakeProductID
```

```
    If .ProductCnt <= 10 Then
```

```
        For j = 6 To .ProductCnt + 5
```

```
            txtTransferName(j).Text = .ProductID(j - 6)
```

```
            txtCustName(j).Text = .ProductName(j - 6)
```

```
            txtVolume(j).Text = .ProductCount(j - 6)
```

```
            .ProductPosition(j - 6) = FindProductPos(.ProductID(j - 6))
```

```
            txtTransferID(j).Text = .ProductPosition(j - 6)
```

```
        Next j
```

```
    Else
```

```
        For j = 6 To 15
```

```
            txtTransferName(j).Text = .ProductID(j - 6)
```

```
            txtCustName(j).Text = .ProductName(j - 6)
```

```
            txtVolume(j).Text = .ProductCount(j - 6)
```

```
            .ProductPosition(j - 6) = FindProductPos(.ProductID(j - 6))
```

```
            txtTransferID(j).Text = .ProductPosition(j - 6)
```

```
        Next j
```

```
    End If
```

```
    List1(1).Clear
```

```
    For i = 0 To .InvoiceCount - 1
```

```

        List1(1).AddItem ,InvoiceID(i)
    Next i
End With
End Sub

Private Sub BtnPrevious_Click(Index As Integer)

    If Index = 0 Then
        If CurrentOrder <= 0 Then
            MsgBox "ไม่มีรายการต่อไป", vbOKOnly + vbInformation, "ข้อผิดพลาด"
            Exit Sub
        End If

        FillVariableWithControlValue (CurrentOrder)
        CurrentOrder = CurrentOrder - 1
        FillControlWithVariable (CurrentOrder)

    ElseIf Index = 1 Then
        If CurTakeProduct <= 0 Then
            MsgBox "ไม่มีรายการต่อไป", vbOKOnly + vbInformation, "ข้อผิดพลาด"
            Exit Sub
        End If
        InitTakeProductControl
        CurTakeProduct = CurTakeProduct - 1
        FillControlWithTransferVariable (CurTakeProduct)
        VScroll3.Max = Int((TakeProductAry(CurTakeProduct).ProductCnt - 1) _
            / MAX_PRODUCT_PER_PAGE)

        VScroll3.Value = 0

    End If

End Sub

```

```
Private Sub Command1_Click()
```

```
    Dim i As Integer
```

```
    Dim CurInvoiceID As String
```

```
    On Error Resume Next
```

```
    If List2(0).ListCount = 0 Then
```

```
        Exit Sub
```

```
    End If
```

```
    i = 0
```

```
    List1(0).AddItem List2(0).List(List2(0).ListIndex)
```

```
    CurInvoiceID = List2(0).List(List2(0).ListIndex)
```

```
    List2(0).RemoveItem List2(0).ListIndex
```

```
    ReDim Preserve TransferAry(TransferCount + 1) As TransferStruc
```

```
    TransferCount = TransferCount + 1
```

```
    Do
```

```
        If InvoiceAry(i).InvoiceID = CurInvoiceID Then
```

```
            Exit Do
```

```
        End If
```

```
        i = i + 1
```

```
    Loop Until i >= InvoiceCount - 1
```

```
    If i < InvoiceCount Then
```

```
        FillTakeProductControl (i)
```

```
    End If
```

```
    List2(0).ListIndex = 0
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
    Dim i As Integer
```

```
    Dim CurInvoiceID As String
```

```
    On Error Resume Next
```

```
    If List1(0).ListCount <= 0 Then
```

```
        Exit Sub
```

```
    End If
```

```
    i = 0
```

```
    List2(0).AddItem List1(0).List(List1(0).ListIndex)
```

```
    CurInvoiceID = List1(0).List(List1(0).ListIndex)
```

```
    List1(0).RemoveItem List1(0).ListIndex
```

```
    Do
```

```
        If InvoiceAry(i).InvoiceID = CurInvoiceID Then
```

```
            Exit Do
```

```
        End If
```

```
        i = i + 1
```

```
    Loop Until i >= InvoiceCount - 1
```

```
    If i < InvoiceCount Then
```

```
        DeleteProductControl (i)
```

```
    End If
```

```
End Sub
```

```
Private Sub DeleteProductControl(InvoiceNo As Integer)
```

```
    Dim j, k, m As Integer
```

```

'On Error GoTo Err_Handler
' Fill textbox table with Blank
For j = 0 To 5
    txtInvoiceID(j).Text = ""
    txtTransferID(j).Text = ""
    txtTransferName(j).Text = ""
    txtCustName(j).Text = ""
    txtVolume(j).Text = ""
Next j

'search match InvoiceID
For j = 0 To TransferCount - 1
    If TransferAry(j).InvoiceID = InvoiceAry(InvoiceNo).InvoiceID Then
        Exit For
    End If
Next j

'Delete with fill TransferAry
For k = j To TransferCount - 2
    TransferAry(k).InvoiceID = TransferAry(k + 1).InvoiceID
    TransferAry(k).TransferID = TransferAry(k + 1).TransferID
    ReDim Preserve TransferAry(k).OrderNo(TransferAry(k + 1).TransferCount) As Integer
    For m = 0 To TransferAry(k).TransferCount
        TransferAry(k).OrderNo(m) = TransferAry(k + 1).OrderNo(m)
    Next m
    TransferAry(k).TransferName = TransferAry(k + 1).TransferName
    TransferAry(k).CustomerName = InvoiceAry(k + 1).CustomerName
    TransferAry(k).TotalVolume = TransferAry(k + 1).TotalVolume
Next k

'TransferAry(k).TransferCount = TransferAry(k).TransferCount - 1
TransferCount = TransferCount - 1

```

```
If TransferCount < 6 Then
```

```
    CurRowInvoice = TransferCount
```

```
End If
```

```
ReDim Preserve TransferAry(TransferCount) As TransferStruc
```

```
If TransferCount > 6 Then
```

```
    For j = 0 To 5
```

```
        txtInvoiceID(j).Text = TransferAry(j).InvoiceID
```

```
        txtTransferID(j).Text = TransferAry(j).TransferID
```

```
        txtTransferName(j).Text = TransferAry(j).TransferName
```

```
        txtCustName(j).Text = TransferAry(j).CustomerName
```

```
        txtVolume(j).Text = TransferAry(j).TotalVolume
```

```
    Next j
```

```
Else
```

```
    For j = 0 To TransferCount - 1
```

```
        txtInvoiceID(j).Text = TransferAry(j).InvoiceID
```

```
        txtTransferID(j).Text = TransferAry(j).TransferID
```

```
        txtTransferName(j).Text = TransferAry(j).TransferName
```

```
        txtCustName(j).Text = TransferAry(j).CustomerName
```

```
        txtVolume(j).Text = TransferAry(j).TotalVolume
```

```
    Next j
```

```
End If
```

```
Text11.Text = Format(CurProductVol - CDbI(InvoiceAry(InvoiceNo).TotalVolume), "##0.00")
```

```
Text15.Text = Format(CurProductMass - CDbI(InvoiceAry(InvoiceNo).TotalMass), "##0.00")
```

```
UpdateVolumeDiff
```

```
UpdateMassDiff
```

```
VScroll2.Max = Int((TransferCount - 1) / MAX_PRODUCT_PER_PAGE)
```

```
VScroll2.Value = 0
```

```
Exit Sub
```


Err_Handler:

Text12.Text = ""

Text13.Text = ""

Resume Next

End Sub

Private Sub Form_Load()

SSTab1.Visible = False

InitTakeProductControl

InitTakeProductVar

InitMenu

InitOrderVariable

InitControl

MakeAllControlNotVisible

ProgramDir = App.Path & "\

OrderDir = App.Path & "\

End Sub

Private Sub MnuAllTransfer_Click()

SSTab1.Tab = 4

InitTransferControl

InitTransferVar

InitTakeProductControl

InitTakeProductVar

List1(1).Clear

End Sub

Private Sub MnuCloseOrder_Click()

```

Dim i As Integer, j As Integer
InitMenu
For i = 0 To OrderCount - 1
    OrderAry(i).CustomerID = ""
    OrderAry(i).CustomerName = ""
    OrderAry(i).CustomerAddr = ""
    OrderAry(i).DistrictID = ""
    For j = 0 To MAX_ORDERLINE - 1
        OrderAry(i).ProductID(j) = ""
        OrderAry(i).ProductName(j) = ""
        OrderAry(i).ProductCount(j) = ""
    Next j
Next i

MakeAllControlNotVisible
SSTab1.Visible = False
End Sub

Private Sub MnuExit_Click()
    SSTab1.Tab = 0
    Unload Form1
End Sub

Private Sub MnuNewOrder_Click()
    SSTab1.Visible = True
    InitMenuWhenOrderExist
    InitControl
    InitOrderVariable
    MakeAllControlVisible
    InitTakeProductVar
    InitTransferVar
    InitTakeProductControl

```

```

InitTransferControl
End Sub

```

```

Private Sub MnuOpenOrder_Click()

```

```

    Dim OpenFlag As Boolean
    Dim TempDate As Date
    On Error Resume Next
    InitControl
    InitTransferVar
    InitTakeProductVar
    InitTransferControl
    InitTakeProductControl
    DateOfOrder = GetDateOfOrder
    If IsCancel = True Then
        Exit Sub
    End If
    TempDate = CDate(DateOfOrder)
    OpenFlag = OpenOrder(Format(TempDate, "ddmmyy"))
    LblOrderDate.Caption = CThaiDate(TempDate)
    If OpenFlag = False Then
        MsgBox "ไม่มีข้อมูลการสั่งซื้อในวันนี้", vbOKOnly, "ข้อผิดพลาด"
    Else
        SSTab1.Visible = True
        SSTab1.Tab = 0
        MakeAllControlVisible
        InitMenuWhenOrderExist
        CurrentOrder = 0
        FillControlWithVariable (CurrentOrder)
        MnuShowAllInvoice_Click
        MnuAllTransfer_Click
    End If
End Sub

```

End If

End Sub

Private Sub MnuPrintInvoice_Click()

Dim db As Database

Dim SQLStr As String

Dim i As Integer

On Error Resume Next

Set db = Workspaces(0).OpenDatabase(ProgramDir & DATABASE_FILE, False, False)

SQLStr = "DELETE * FROM AllInvoice"

db.Execute SQLStr, dbFailOnError

For i = 0 To InvoiceCount - 1

SQLStr = "INSERT INTO AllInvoice (ID,InvoiceID,TransferID," & _
 "TransferName,CustomerName>TotalVolume,InvoiceDate) " & _
 "VALUES (" & CStr(i) & ", "" & InvoiceAry(i).InvoiceID & "" "" & _
 InvoiceAry(i).TransferID & "" "" & _
 InvoiceAry(i).TransferName & "" "" & _
 InvoiceAry(i).CustomerName & "" "" & _
 InvoiceAry(i).TotalVolume & "" "" & _
 CThaiDate(InvoiceAry(i).InvoiceDate) & """);"

db.Execute SQLStr, dbFailOnError

Next i

db.Close

CrystalReport1.ReportFileName = ProgramDir & "allinvoi.rpt"

CrystalReport1.Action = 1

End Sub

```
Private Sub MnuPrintTakeProduct_Click()
```

```
    Dim db As Database
```

```
    Dim TakeProdInf As Recordset
```

```
    Dim SQLStr As String
```

```
    Dim i As Integer
```

```
    'On Error Resume Next
```

```
    Set db = Workspaces(0).OpenDatabase(ProgramDir & DATABASE_FILE, False, False)
```

```
    SQLStr = "DELETE * FROM TakeProduct"
```

```
    db.Execute SQLStr, dbFailOnError
```

```
    SQLStr = "DELETE * FROM TakeProductHeader"
```

```
    db.Execute SQLStr, dbFailOnError
```

```
    SQLStr = "DELETE * FROM TakeProdInv"
```

```
    db.Execute SQLStr, dbFailOnError
```

```
    With TakeProductAry(CurTakeProduct)
```

```
        SQLStr = "INSERT INTO TakeProductHeader ( ID, TakeProductID, " & _
```

```
            "TruckID, TruckNumber, DriverName ) " & _
```

```
            "VALUES ( 0, " & """" & .TakeProductID & """, """" & _
```

```
                .TruckID & """, """" & .TruckNumber & """, """" & .DriverName & """);"
```

```
        db.Execute SQLStr, dbFailOnError
```

```
    For i = 0 To .InvoiceCount - 1
```

```
        SQLStr = "INSERT INTO TakeProdInv ( ID, TakeProductID, InvoiceID) " & _
```

```
            "VALUES ( " & CStr(i) & ", """" & .TakeProductID & """, """" & .InvoiceID(i) &
```

```
            """);"
```

```
        db.Execute SQLStr, dbFailOnError
```

```
    Next i
```

```
    For i = 0 To .ProductCnt - 1
```

```
        SQLStr = "INSERT INTO TakeProduct ( ID, TakeProductID, ProductPosition, " & _
```

```
            " ProductID, ProductName, ProductCount ) " & _
```

```

        "VALUES ( " & CStr(i) & ", "" & .TakeProductID & "", "" & _
        .ProductPosition(i) & "", "" & .ProductID(i) & _
        "", "" & .ProductName(i) & _
        "", " & .ProductCount(i) & ");"

    db.Execute SQLStr, dbFailOnError

Next i
End With

db.Close
CrystalReport1.ReportFileName = ProgramDir & "takeprod.rpt"
CrystalReport1.Action = 1
CrystalReport1.ReportFileName = ProgramDir & "TakeInv.rpt"
CrystalReport1.Action = 1
End Sub

Private Sub MnuPrintTransfer_Click()

    Dim db As Database
    Dim SQLStr As String
    Dim i As Integer
    On Error Resume Next

    Set db = Workspaces(0).OpenDatabase(ProgramDir & DATABASE_FILE, False, False)
    SQLStr = "DELETE * FROM TransferStruc"
    db.Execute SQLStr, dbFailOnError
    SQLStr = "DELETE * FROM TransferHeader"
    db.Execute SQLStr, dbFailOnError

    SQLStr = "INSERT INTO TransferHeader (ID, TruckID, TruckNumber, DriverName) " & _
        "VALUES ( 0, "" & TruckID & "", " & "" & TruckNumber & "", " & _
        "" & DriverName & "");"
    db.Execute SQLStr, dbFailOnError

```

```

For i = 0 To TransferCount - 1
    SQLStr = "INSERT INTO TransferStruc (ID,InvoiceID, TransferID, " & _
        "Transfername,customername,totalvolume) " & _
        "VALUES ( " & CStr(i) & " , " & "" & TransferAry(i).InvoiceID & "" & "
        , " & "" & _
        TransferAry(i).TransferID & "" & " , " & "" & _
        TransferAry(i).TransferName & "" & " , " & "" & _
        TransferAry(i).CustomerName & "" & " , " & "" & _
        TransferAry(i).TotalVolume & "" & ");"
    db.Execute SQLStr, dbFailOnError
Next i
db.Close
CrystalReport1.ReportFileName = ProgramDir & "transfer.rpt"
CrystalReport1.Action = 1

End Sub

Private Sub MnuSaveOrder_Click()
    Dim FileName As String
    Dim SaveFlag As Boolean

    FileName = Format(CDate(DateOfOrder), "ddmmyy") & ".ord"
    FillVariableWithControlValue (CurrentOrder)
    SaveFlag = SaveOrder(FileName)

    If SaveFlag = False Then ...
        MsgBox "ไม่สามารถจัดเก็บข้อมูลได้", vbOKOnly, "ข้อผิดพลาด"
    End If

End Sub

End Sub

```

```
Private Sub MnuShowAllInvoice_Click()
```

```
    Dim i As Integer
```

```
    Dim AllWeight As Double
```

```
    On Error Resume Next
```

```
    SSTab1.Tab = 2°
```

```
    InvoiceCount = OrderCount
```

```
    ReDim Preserve InvoiceAry(InvoiceCount) As InvoiceStruc
```

```
    For i = 0 To OrderCount - 1
```

```
        Text1(i).Text = Format(CDate(DateOfOrder), "ddmmyy") & Format(i, "000")
```

```
        InvoiceAry(i).OrderNo = i
```

```
        InvoiceAry(i).InvoiceID = Text1(i).Text
```

```
        Text2(i).Text = CThaiDate(DateOfOrder)
```

```
        InvoiceAry(i).InvoiceDate = DateOfOrder
```

```
        Text3(i).Text = OrderAry(i).DistrictID
```

```
        InvoiceAry(i).TransferID = Text3(i).Text
```

```
        Text4(i).Text = OrderAry(i).CustomerAddr
```

```
        InvoiceAry(i).TransferName = Text4(i).Text
```

```
        Text8(i).Text = OrderAry(i).CustomerName
```

```
        InvoiceAry(i).CustomerName = Text8(i).Text
```

```
        Text9(i).Text = Format(AllVolumeOfOrder(i, AllWeight), "##0.00")
```

```
        InvoiceAry(i).TotalVolume = Text9(i).Text
```

```
        InvoiceAry(i).TotalMass = Format(AllWeight, "##0.00")
```

```
        ValidProductAry(i).CustomerID = OrderAry(i).CustomerID
```

```
        ValidProductAry(i).CustomerName = OrderAry(i).CustomerName
```

```
        ValidProductAry(i).CustomerAddr = OrderAry(i).CustomerAddr
```

```
        ValidProductAry(i).DistrictID = OrderAry(i).DistrictID
```


Next i

VScroll1.Max = Int((InvoiceCount - 1) / MAX_INVOICE_PER_PAGE)

VScroll1.Value = 0

MnuShowTransferProduct.Enabled = True

MnuShowTakeProduct.Enabled = True

MnuPrintProduct = True

MnuPrintTransfer.Enabled = True

MnuPrintTakeProduct.Enabled = True

End Sub

Private Sub MnuShowInvoice_Click()

'Dim Month, Day, Year As String

'Dim DateFmt As String

Dim AllWeight As Double

SSTab1.Tab = 1

Text1(12).Text = Format(CDate(Now), "ddmmyy") & Format(CurrentOrder, "000")

'Day = Left(DateOfOrder, 2)

'Month = Mid(DateOfOrder, 3, 2)

'Year = Right(DateOfOrder, 2)

'DateFmt = Day & "/" & Month & "/" & Year

Text2(12).Text = CThaiDate(DateOfOrder)

Text3(12).Text = OrderAry(CurrentOrder).DistrictID

Text4(12).Text = OrderAry(CurrentOrder).CustomerAddr

Text8(12).Text = OrderAry(CurrentOrder).CustomerName

Text9(12).Text = Format(AllVolumeOfOrder(CurrentOrder, AllWeight), "##0.00")

End Sub

```

Private Sub MnuShowTakeProduct_Click()
    Dim i As Integer
    Dim j As Integer
    Dim k As Integer
    Dim l As Integer
    Dim m As Integer
    On Error Resume Next
    If List1(0).ListCount = 0 Then
        Exit Sub
    End If

    SSTab1.Tab = 3
    ReDim Preserve TakeProductAry(TakeProductCount + 1) As TakeProductStruc
    TakeProductAry(TakeProductCount).OrderCount = TransferCount
    ReDim Preserve TakeProductAry(TakeProductCount).OrderNo( _
    TakeProductAry(TakeProductCount).OrderCount) As Integer
    TakeProductAry(TakeProductCount).InvoiceCount = TransferCount
    ReDim Preserve TakeProductAry(TakeProductCount).InvoiceID( _
    TakeProductAry(TakeProductCount).InvoiceCount) As String

    List1(1).Clear
    InitTakeProductControl
    With TakeProductAry(TakeProductCount)

        For i = 0 To .InvoiceCount - 1
            .InvoiceID(i) = List1(0).List(i)
            List1(1).AddItem .InvoiceID(i)
        Next i

        k = 0

        For j = 0 To .InvoiceCount - 1

```

```

For i = 0 To InvoiceCount - 1
    If InvoiceAry(i).InvoiceID = .InvoiceID(j) Then
        .OrderNo(k) = InvoiceAry(i).OrderNo
        k = k + 1
    End If
Next i
Next j
' k = จำนวนใบ Order ในใบหยิบสินค้าที่จะสร้าง
l = k

Debug.Print "Begin In Loop transfer product"
' m = รายการสินค้าในใบ Order ทุกใบในใบหยิบสินค้าที่จะสร้าง โดยที่ยังไม่ได้ Group
m = 0
For i = 0 To .OrderCount - 1
    For k = 0 To OrderCount - 1
        If .OrderNo(i) = k Then
            For j = 0 To ValidProductAry(k).ProductCnt - 1
                ReDim Preserve .ProductID(m + 1) As String
                ReDim Preserve .ProductName(m + 1) As String
                ReDim Preserve .ProductCount(m + 1) As String
                ReDim Preserve .ProductPosition(m + 1) As String
                .ProductID(m) = ValidProductAry(k).ProductID(j)
                .ProductName(m) = ValidProductAry(k).ProductName(j)
                .ProductCount(m) = ValidProductAry(k).ProductCount(j)
                m = m + 1
            Next j
        End If
    Next k
Next i

.ProductCnt = m
.TruckID = TruckID

```

```

Text18(1).Text = .TruckID
.TruckNumber = TruckNumber
Text19(1).Text = .TruckNumber
.DriverName = DriverName
Text20(1).Text = .DriverName
CurTakeProductID = (Rnd() * 100)

.TakeProductID = Format(CurTakeProductID, "0000000")
Text20(2).Text = .TakeProductID

For i = 0 To .ProductCnt - 1
    .ProductPosition(i) = FindProductPos(.ProductID(i))
Next i
GroupTakeProduct
m = .ProductCnt
If .ProductCnt <= 10 Then
    For j = 6 To m + 5
        txtTransferName(j).Text = .ProductID(j - 6)
        txtCustName(j).Text = .ProductName(j - 6)
        txtVolume(j).Text = .ProductCount(j - 6)
        '.ProductPosition(j - 6) = FindProductPos(.ProductID(j - 6))
        txtTransferID(j).Text = .ProductPosition(j - 6)
    Next j
Else
    For j = 6 To 15
        txtTransferName(j).Text = .ProductID(j - 6)
        txtCustName(j).Text = .ProductName(j - 6)
        txtVolume(j).Text = .ProductCount(j - 6)
        '.ProductPosition(j - 6) = FindProductPos(.ProductID(j - 6))
        txtTransferID(j).Text = .ProductPosition(j - 6)
    Next j
End If

```

```
VScroll3.Max = Int((ProductCnt - 1) / MAX_PRODUCT_PER_PAGE)
```

```
VScroll3.Value = 0
```

```
End With
```

```
TakeProductCount = TakeProductCount + 1
```

```
CurTakeProduct = TakeProductCount - 1
```

```
ClearTransferControl
```

```
InitTransferVar
```

```
End Sub
```

```
Private Sub GroupTakeProduct()
```

```
Dim db As Database
```

```
Dim GrpTakeProduct As Recordset
```

```
Dim SQLStr As String
```

```
Dim i As Integer
```

```
'Dim TempProdInf As GrpProdStruc
```

```
Set db = Workspaces(0).OpenDatabase(ProgramDir & DATABASE_FILE, False, False)
```

```
SQLStr = "DELETE * FROM TakeProduct"
```

```
db.Execute SQLStr, dbFailOnError
```

```
With TakeProductAry(TakeProductCount)
```

```
For i = 0 To .ProductCnt - 1
```

```
SQLStr = "INSERT INTO TakeProduct ( TakeProductID, ProductID, ProductPosition, " &
```

```
ProductName, ProductCount ) " & _
```

```
"VALUES ( "" & .TakeProductID & "" ,"" & .ProductID(i) & "" ,"" & _
```

```
.ProductPosition(i) & "" ,"" & _
```

```

        .ProductName(i) & ""','" & _
        .ProductCount(i) & "");"
db.Execute SQLStr, dbFailOnError
Next i

SQLStr = "SELECT ProductID, ProductPosition, " & _
        " Max(ProductName) As ProductNameUni, " & _
        " Sum(ProductCount) As TotalProductCount " & _
        " FROM TakeProduct " & _
        " GROUP BY ProductID, ProductPosition" & _
        " ORDER BY ProductPosition "

Set GrpTakeProduct = db.OpenRecordset(SQLStr, dbOpenSnapshot)
i = 0
Do Until GrpTakeProduct.EOF
    ReDim Preserve .ProductID(i + 1) As String
    ReDim Preserve .ProductPosition(i + 1) As String
    ReDim Preserve .ProductName(i + 1) As String
    ReDim Preserve .ProductCount(i + 1) As String

    .ProductID(i) = GrpTakeProduct.Fields(0).Value
    .ProductPosition(i) = GrpTakeProduct.Fields(1).Value
    .ProductName(i) = GrpTakeProduct.Fields("ProductNameUni").Value
    .ProductCount(i) = CStr(GrpTakeProduct.Fields("TotalProductCount").Value)
    i = i + 1
    GrpTakeProduct.MoveNext
Loop

.ProductCnt = i
End With

GrpTakeProduct.Close
db.Close

```

```
End Sub
```

```
Private Sub ClearTransferControl()
```

```
Dim i As Integer
```

```
List1(0).Clear
```

```
Text10.Text = ""
```

```
Text11.Text = ""
```

```
Text12.Text = ""
```

```
Text13.Text = ""
```

```
Text14.Text = ""
```

```
Text15.Text = ""
```

```
Text16.Text = ""
```

```
Text17.Text = ""
```

```
Text18(0).Text = ""
```

```
Text19(0).Text = ""
```

```
Text20(0).Text = ""
```

```
For i = 0 To 5
```

```
txtInvoiceID(i).Text = ""
```

```
txtTransferID(i).Text = ""
```

```
txtTransferName(i).Text = ""
```

```
txtCustName(i).Text = ""
```

```
txtVolume(i).Text = ""
```

```
Next i
```

```
End Sub
```

```

Private Sub MnuSortByDistrictID_Click()
    Dim i As Integer
    Dim db As Database
    Dim InvoiceRset As Recordset
    Dim SQLStr As String
    On Error Resume Next
    MnuSortByInvoiceID.Checked = False
    MnuSortByDistrictID.Checked = True
    Set db = Workspaces(0).OpenDatabase(ProgramDir & DATABASE_FILE, False, False)
    SQLStr = "DELETE * FROM TempInvoice"
    db.Execute SQLStr, dbFailOnError

    For i = 0 To InvoiceCount - 1

        With InvoiceAry(i)
            SQLStr = "INSERT INTO TempInvoice ( OrderNo, InvoiceID, " & _
                "InvoiceDate, TransferID, TransferName, " & _
                "CustomerName, TotalVolume ) " & _
                "VALUES ( " & .OrderNo & ", " & .InvoiceID & ", " & _
                .InvoiceDate & ", " & .TransferID & ", " & _
                .TransferName & ", " & .CustomerName & ", " & _
                .TotalVolume & " );"
            db.Execute SQLStr
        End With
    Next i

    SQLStr = "SELECT * FROM TempInvoice ORDER BY TransferID, InvoiceID"
    Set InvoiceRset = db.OpenRecordset(SQLStr, dbOpenSnapshot)
    i = 0

    Do Until InvoiceRset.EOF
        If i > InvoiceCount - 1 Then

```



```

Exit Do
End If
With InvoiceAry(i)
    .OrderNo = InvoiceRset.Fields(0).Value
    .InvoiceID = InvoiceRset.Fields(1).Value
    .InvoiceDate = Format(InvoiceRset.Fields(2).Value, "Long Date")
    .TransferID = InvoiceRset.Fields(3).Value
    .TransferName = InvoiceRset.Fields(4).Value
    .CustomerName = InvoiceRset.Fields(5).Value
    .TotalVolume = InvoiceRset.Fields(6).Value
End With
i = i + 1
InvoiceRset.MoveNext
Loop

SQLStr = "DELETE * FROM TempInvoice"
db.Execute SQLStr, dbFailOnError

For i = 0 To InvoiceCount - 1

With InvoiceAry(i)
    SQLStr = "INSERT INTO TempInvoice ( OrderNo, InvoiceID, " & _
        "InvoiceDate, TransferID, TransferName, " & _
        "CustomerName, TotalVolume ) " & _
        "VALUES ( " & .OrderNo & ", "" & .InvoiceID & "" , "" & _
        .InvoiceDate & "" , "" & .TransferID & "" , "" & _
        .TransferName & "" , "" & .CustomerName & "" , "" & _
        .TotalVolume & "" );"

    db.Execute SQLStr
End With
Next i

```

```
InvoiceRset.Close
```

```
db.Close
```

```
For i = 0 To InvoiceCount - 1
```

```
  With InvoiceAry(i)
```

```
    Text1(i).Text = .InvoiceID
```

```
    Text2(i).Text = CThaiDate(CDate(.InvoiceDate))
```

```
    Text3(i).Text = .TransferID
```

```
    Text4(i).Text = .CustomerAddr
```

```
    Text8(i).Text = .CustomerName
```

```
    Text9(i).Text = .TotalVolume
```

```
  End With
```

```
Next i
```

```
End Sub
```

```
Private Sub MnuSortByInvoiceID_Click()
```

```
  Dim i As Integer
```

```
  Dim db As Database
```

```
  Dim InvoiceRset As Recordset
```

```
  Dim SQLStr As String
```

```
  On Error Resume Next
```

```
  MnuSortByInvoiceID.Checked = True
```

```
  MnuSortByDistrictID.Checked = False
```

```
  Set db = Workspaces(0).OpenDatabase(ProgramDir & DATABASE_FILE, False, False)
```

```
  SQLStr = "DELETE * FROM TempInvoice"
```

```
  db.Execute SQLStr, dbFailOnError
```

```
For i = 0 To InvoiceCount - 1
```

```
  With InvoiceAry(i)
```

```
    SQLStr = "INSERT INTO TempInvoice ( OrderNo, InvoiceID, " & _
```

```

    "InvoiceDate, TransferID, TransferName, " & _
    "CustomerName, TotalVolume ) " & _
    "VALUES ( " & .OrderNo & ", "" & .InvoiceID & "", "" & _
    .InvoiceDate & "", "" & .TransferID & "", "" & _
    .TransferName & "", "" & .CustomerName & "", "" & _
    .TotalVolume & "");"

    db.Execute SQLStr
End With
Next i
SQLStr = "SELECT * FROM TempInvoice ORDER BY InvoiceID"
Set InvoiceRset = db.OpenRecordset(SQLStr, dbOpenSnapshot)
i = 0

Do Until InvoiceRset.EOF
    If i > InvoiceCount - 1 Then
        Exit Do
    End If
    With InvoiceAry(i)
        .OrderNo = InvoiceRset.Fields(0).Value
        .InvoiceID = InvoiceRset.Fields(1).Value
        .InvoiceDate = Format(InvoiceRset.Fields(2).Value, "Long Date")
        .TransferID = InvoiceRset.Fields(3).Value
        .TransferName = InvoiceRset.Fields(4).Value
        .CustomerName = InvoiceRset.Fields(5).Value
        .TotalVolume = InvoiceRset.Fields(6).Value
    End With
    i = i + 1
    InvoiceRset.MoveNext
Loop

SQLStr = "DELETE * FROM TempInvoice"

```

```
db.Execute SQLStr, dbFailOnError
```

```
For i = 0 To InvoiceCount - 1
```

```
With InvoiceAry(i)
```

```
SQLStr = "INSERT INTO TempInvoice ( OrderNo, InvoiceID, " & _  
        "InvoiceDate, TransferID, TransferName, " & _  
        "CustomerName, TotalVolume ) " & _  
        "VALUES ( " & .OrderNo & ", " & .InvoiceID & ", " & _  
        .InvoiceDate & ", " & .TransferID & ", " & _  
        .TransferName & ", " & .CustomerName & ", " & _  
        .TotalVolume & ");"
```

```
db.Execute SQLStr
```

```
End With
```

```
Next i
```

```
InvoiceRset.Close
```

```
db.Close
```

```
For i = 0 To InvoiceCount - 1
```

```
With InvoiceAry(i)
```

```
Text1(i).Text = .InvoiceID
```

```
Text2(i).Text = CThaiDate(CDate(.InvoiceDate))
```

```
Text3(i).Text = .TransferID
```

```
Text4(i).Text = .CustomerAddr
```

```
Text8(i).Text = .CustomerName
```

```
Text9(i).Text = .TotalVolume
```

```
End With
```

```
Next i
```

```
End Sub
```

```
Private Sub Text10_Change()
```

```
If Not IsNumeric(Text10.Text) Then
```

```
Exit Sub
```

```
End If
```

```
CurTruckVol = CDbI(Text10.Text)
```

```
UpdateVolumeDiff
```

```
UpdateMassDiff
```

```
End Sub
```

```
Private Sub Text11_Change()
```

```
    If Not IsNumeric(Text11.Text) Then
```

```
        Exit Sub
```

```
    End If
```

```
    CurProductVol = CDbI(Text11.Text)
```

```
    UpdateVolumeDiff
```

```
    UpdateMassDiff
```

```
End Sub
```

```
Private Sub Text14_Change()
```

```
    If Not IsNumeric(Text14.Text) Then
```

```
        Exit Sub
```

```
    End If
```

```
    CurTruckMass = CDbI(Text14.Text)
```

```
    UpdateVolumeDiff
```

```
    UpdateMassDiff
```

```
End Sub
```

```
Private Sub Text15_Change()
```

```
    If Not IsNumeric(Text15.Text) Then
```

```
        Exit Sub
```

```
    End If
```

```
    CurProductMass = CDbI(Text15.Text)
```

```
UpdateVolumeDiff
```

```
UpdateMassDiff
```

```
End Sub
```

```
Private Sub Text18_KeyPress(Index As Integer, KeyAscii As Integer)
```

```
    If KeyAscii = 13 Then
```

```
        Text20(0).SetFocus
```

```
        KeyAscii = 0
```

```
    End If
```

```
End Sub
```

```
Private Sub Text18_LostFocus(Index As Integer)
```

```
    Dim Weight As Double, Volume As Double
```

```
    If Not IsNumeric(Text18(0).Text) Then
```

```
        Exit Sub
```

```
    End If
```

```
    GetTruckInfo CInt(Text18(0).Text), Weight, Volume
```

```
    Text10.Text = Format(Volume, "##0.00")
```

```
    Text14.Text = Format(Weight, "##0.00")
```

```
    UpdateVolumeDiff
```

```
    UpdateMassDiff
```

```
    If Index = 0 Then
```

```
        TruckID = Text18(0).Text
```

```
    End If
```

```
End Sub
```

```
Private Sub Text19_Change(Index As Integer)
```

```
    If Index = 0 Then
```

```
TruckNumber = Text19(Index).Text
End If
End Sub

Private Sub Text20_Change(Index As Integer)
    If Index = 0 Then
        DriverName = Text20(Index).Text
    End If
End Sub

Private Sub Text20_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 13 Then
        Text19(0).SetFocus
        KeyAscii = 0
    End If
End Sub

Private Sub Text5_Change(Index As Integer)
    OrderArray(CurrentOrder).ProductID(Index) = Text5(Index).Text
End Sub

Private Sub Text5_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 13 Then
        Text7(Index).SetFocus
        KeyAscii = 0
    End If
End Sub
```

```

Private Sub Text5_LostFocus(Index As Integer)
    Dim db As Database
    Dim Rs As Recordset
    Dim SQLStr As String

    On Error GoTo ErrHandler
    'If Not IsNumeric(Text5(Index).Text) And _
    ' Not (Len(Text5(Index).Text) = 6) Then
    ' MsgBox "ข้อมูลรหัสสินค้าจะต้องเป็นตัวเลข 6 หลักในรูปแบบ XXXXXX", _
    ' vbOKOnly + vbInformation, "ข้อผิดพลาด"
    ' Exit Sub
'End If

Set db = Workspaces(0).OpenDatabase(ProgramDir & DATABASE_FILE, False, False)
SQLStr = "SELECT ProductID, ProductName FROM ProductInfo " & _
        "WHERE ProductID = "" & Text5(Index).Text & """"
Set Rs = db.OpenRecordset(SQLStr, dbOpenDynaset)

Text6(Index).Text = Rs.Fields(1).Value
Rs.Close
db.Close
Exit Sub
ErrHandler:
Text6(Index).Text = ""
End Sub

Private Sub Text6_Change(Index As Integer)
    OrderAry(CurrentOrder).ProductName(Index) = Text6(Index).Text
End Sub

```



```
Private Sub Text7_Change(Index As Integer)
    OrderAry(CurrentOrder).ProductCount(Index) = Text7(Index).Text
End Sub
```

```
Private Sub Text7_KeyPress(Index As Integer, KeyAscii As Integer)
    On Error Resume Next
    If KeyAscii = 13 Then
        Text5(Index + 1).SetFocus
        KeyAscii = 0
    End If
End Sub
```

```
Private Sub Text7_LostFocus(Index As Integer)

    'If Not IsNumeric(Text7(Index).Text) Then
    ' MsgBox "จำนวนสินค้าต้องเป็นตัวเลข", _
    ' vbOKOnly + vbInformation, "ข้อผิดพลาด"
    ' End If
End Sub
```

```
Private Sub TxtCustomerAddr_Change()
    OrderAry(CurrentOrder).CustomerAddr = TxtCustomerAddr.Text
End Sub
```

```
Private Sub TxtCustomerAddr_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        TxtDistrictID.SetFocus
    End If
End Sub
```

```

    KeyAscii = 0
End If

End Sub

Private Sub TxtCustomerID_Change()
    OrderAry(CurrentOrder).CustomerID = TxtCustomerID.Text
End Sub

Private Sub TxtCustomerID_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        TxtCustomerName.SetFocus
        KeyAscii = 0
    End If
End Sub

Private Sub TxtCustomerID_LostFocus()
    Dim db As Database
    Dim Rs As Recordset
    Dim SQLStr As String
    On Error Resume Next
    Set db = Workspaces(0).OpenDatabase(ProgramDir & DATABASE_FILE, False, False)
    SQLStr = "SELECT CustomerID, CustomerName, TransAreaID, TransAreaName " & _
        " FROM CustomerInfo, TransferArea " & _
        " WHERE CustomerID = " & TxtCustomerID.Text & _
        " And CustomerInfo.TransferAreaID = TransferArea.TransAreaID "
    Set Rs = db.OpenRecordset(SQLStr, dbOpenSnapshot)

```

```
TxtCustomerName.Text = Rs.Fields(1).Value  
TxtDistrictID.Text = Rs.Fields(2).Value  
TxtCustomerAddr.Text = Rs.Fields(3).Value  
Rs.Close  
db.Close
```

```
End Sub
```

```
Private Sub TxtCustomerName_Change()  
    OrderAry(CurrentOrder).CustomerName = TxtCustomerName  
End Sub
```

```
Private Sub TxtCustomerName_KeyPress(KeyAscii As Integer)  
    If KeyAscii = 13 Then  
        TxtCustomerAddr.SetFocus  
        KeyAscii = 0  
    End If
```

```
End Sub
```

```
Private Sub TxtDistrictID_Change()  
    OrderAry(CurrentOrder).DistrictID = TxtDistrictID.Text  
End Sub
```

```
Private Sub TxtDistrictID_KeyPress(KeyAscii As Integer)  
    If KeyAscii = 13 Then  
        Text5(0).SetFocus
```

```
KeyAscii = 0
End If

End Sub

Private Sub VScroll1_Change()
    Dim i As Integer
    Dim j As Integer, k As Integer
    j = (VScroll1.Value * MAX_INVOICE_PER_PAGE)

    If j + 11 > InvoiceCount - 1 Then
        For i = j To j + 11
            If i <= InvoiceCount - 1 Then
                Text1(k).Text = InvoiceAry(i).InvoiceID
                Text2(k).Text = InvoiceAry(i).InvoiceDate
                Text3(k).Text = InvoiceAry(i).TransferID
                Text4(k).Text = InvoiceAry(i).TransferName
                Text8(k).Text = InvoiceAry(i).CustomerName
                Text9(k).Text = InvoiceAry(i).TotalVolume

                k = k + 1
            Else
                Text1(k).Text = ""
                Text2(k).Text = ""
                Text3(k).Text = ""
                Text4(k).Text = ""
                Text8(k).Text = ""
                Text9(k).Text = ""

                k = k + 1
            End If
        Next i
    End If
```

```

Next i

Else
  For i = j To j + 5
    Text1(k).Text = InvoiceAry(i).InvoiceID
    Text2(k).Text = InvoiceAry(i).InvoiceDate
    Text3(k).Text = InvoiceAry(i).TransferID
    Text4(k).Text = InvoiceAry(i).TransferName
    Text8(k).Text = InvoiceAry(i).CustomerName
    Text9(k).Text = InvoiceAry(i).TotalVolume
    k = k + 1
  Next i
End If

End Sub

Private Sub VScroll2_Change()
  Dim i As Integer
  Dim j As Integer, k As Integer
  j = (VScroll2.Value * MAX_TRANSFER_PER_PAGE)

  If j + 5 > TransferCount - 1 Then
    For i = j To j + 5
      If i <= TransferCount - 1 Then
        txtInvoiceID(k).Text = TransferAry(i).InvoiceID
        txtTransferID(k).Text = TransferAry(i).TransferID
        txtTransferName(k).Text = TransferAry(i).TransferName
        txtCustName(k).Text = TransferAry(i).CustomerName
        txtVolume(k).Text = TransferAry(i).TotalVolume
        k = k + 1
      Else
        txtInvoiceID(k).Text = ""

```

```

    txtTransferID(k).Text = ""
    txtTransferName(k).Text = ""
    txtCustName(k).Text = ""
    txtVolume(k).Text = ""
    k = k + 1
End If
Next i
Else
    For i = j To j + 5
        txtInvoiceID(k).Text = TransferAry(i).InvoiceID
        txtTransferID(k).Text = TransferAry(i).TransferID
        txtTransferName(k).Text = TransferAry(i).TransferName
        txtCustName(k).Text = TransferAry(i).CustomerName
        txtVolume(k).Text = TransferAry(i).TotalVolume
        k = k + 1
    Next i
End If

End Sub

Private Sub VScroll3_Change()
    Dim i As Integer
    Dim j As Integer, k As Integer

    j = (VScroll3.Value * MAX_PRODUCT_PER_PAGE)
    k = 6
    With TakeProductAry(CurTakeProduct)
        If j + 9 > TakeProductAry(CurTakeProduct).ProductCnt - 1 Then
            For i = j To j + 9
                If i <= TakeProductAry(CurTakeProduct).ProductCnt - 1 Then
                    txtTransferName(k).Text = .ProductID(i)
                    txtCustName(k).Text = .ProductName(i)
                End If
            Next i
        End If
    End With
End Sub

```

```
txtVolume(k).Text = .ProductCount(i)
txtTransferID(k).Text = .ProductPosition(i)
k = k + 1
Else
txtTransferName(k).Text = ""
txtCustName(k).Text = ""
txtVolume(k).Text = ""
txtTransferID(k).Text = ""
k = k + 1
End If
Next i
Else
For i = j To j + 9
txtTransferName(k).Text = .ProductID(i)
txtCustName(k).Text = .ProductName(i)
txtVolume(k).Text = .ProductCount(i)
txtTransferID(k).Text = .ProductPosition(i)
k = k + 1
Next i
End If
End With
End Sub
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



ประวัติผู้เขียน

นาย ยงยศ เกียรติกุล เกิดวันที่ 11 ธันวาคม พ.ศ. 2500 ที่กรุงเทพมหานคร สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต (สาขาวิศวกรรมอุตสาหการ) จากคณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น ในปีการศึกษา 2522 และสำเร็จการศึกษาปริญญาบริหารธุรกิจมหาบัณฑิต จากมหาวิทยาลัยธรรมศาสตร์ ในปีการศึกษา 2536 เข้าศึกษาต่อในระดับปริญญามหาบัณฑิต ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2537 เป็นสมาชิกอาวุโสของสถาบันวิศวกรอุตสาหการ ประเทศสหรัฐอเมริกา

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย