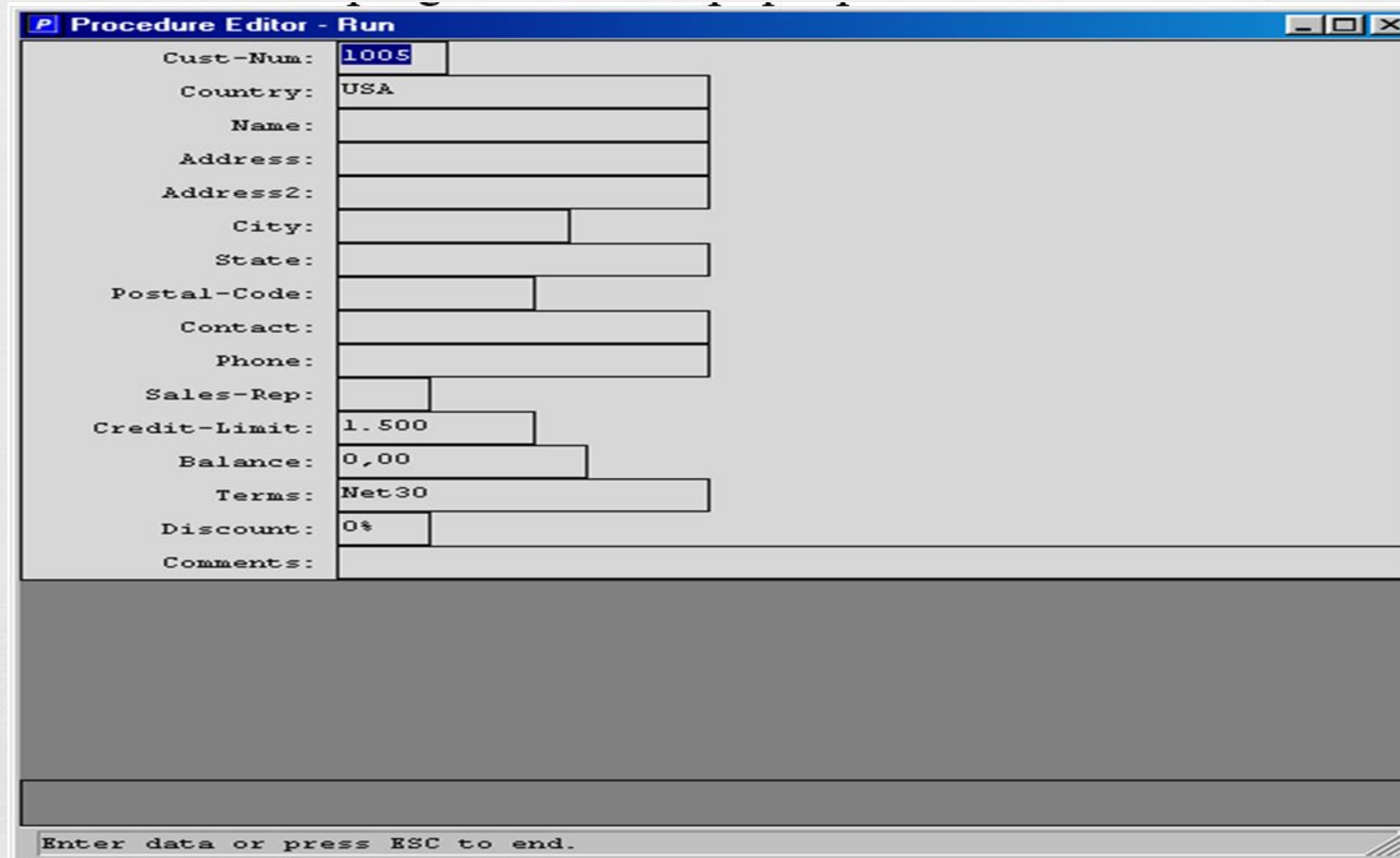


PROGRESS OPENEDGE

» Básico - Laboratórios

2025/ julho

1. Escreva um programa **l-add1.p** que permita incluir um cliente. A tela deve assemelhar-se a que segue:

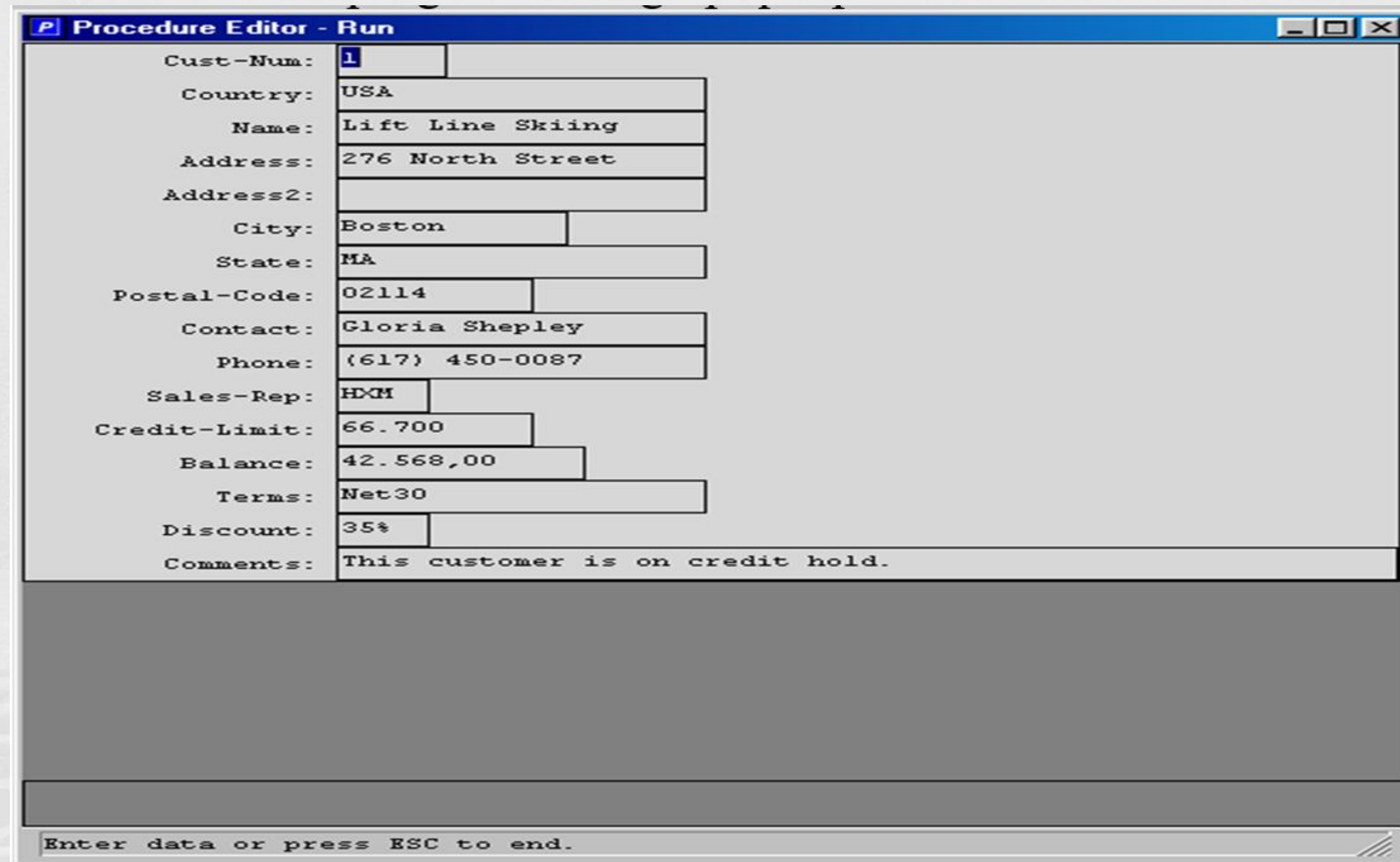


Cust-Num:	1005
Country:	USA
Name:	
Address:	
Address2:	
City:	
State:	
Postal-Code:	
Contact:	
Phone:	
Sales-Rep:	
Credit-Limit:	1.500
Balance:	0,00
Terms:	Net30
Discount:	0%
Comments:	

Enter data or press ESC to end.

2. Modifique o programa **l-add1.p**, chamando-o de **l-add2.p**, para continue incluindo um cliente, mas não utilize os comandos INSERT, UPDATE e SET.

1. Modifique o programa **l-add1.p**, chamando-o de **l-add3.p**, para que possibilite incluir vários clientes.
2. Escreva um programa **l-chg1.p** que permita alterar os clientes. A tela deve assemelhar-se a que segue:



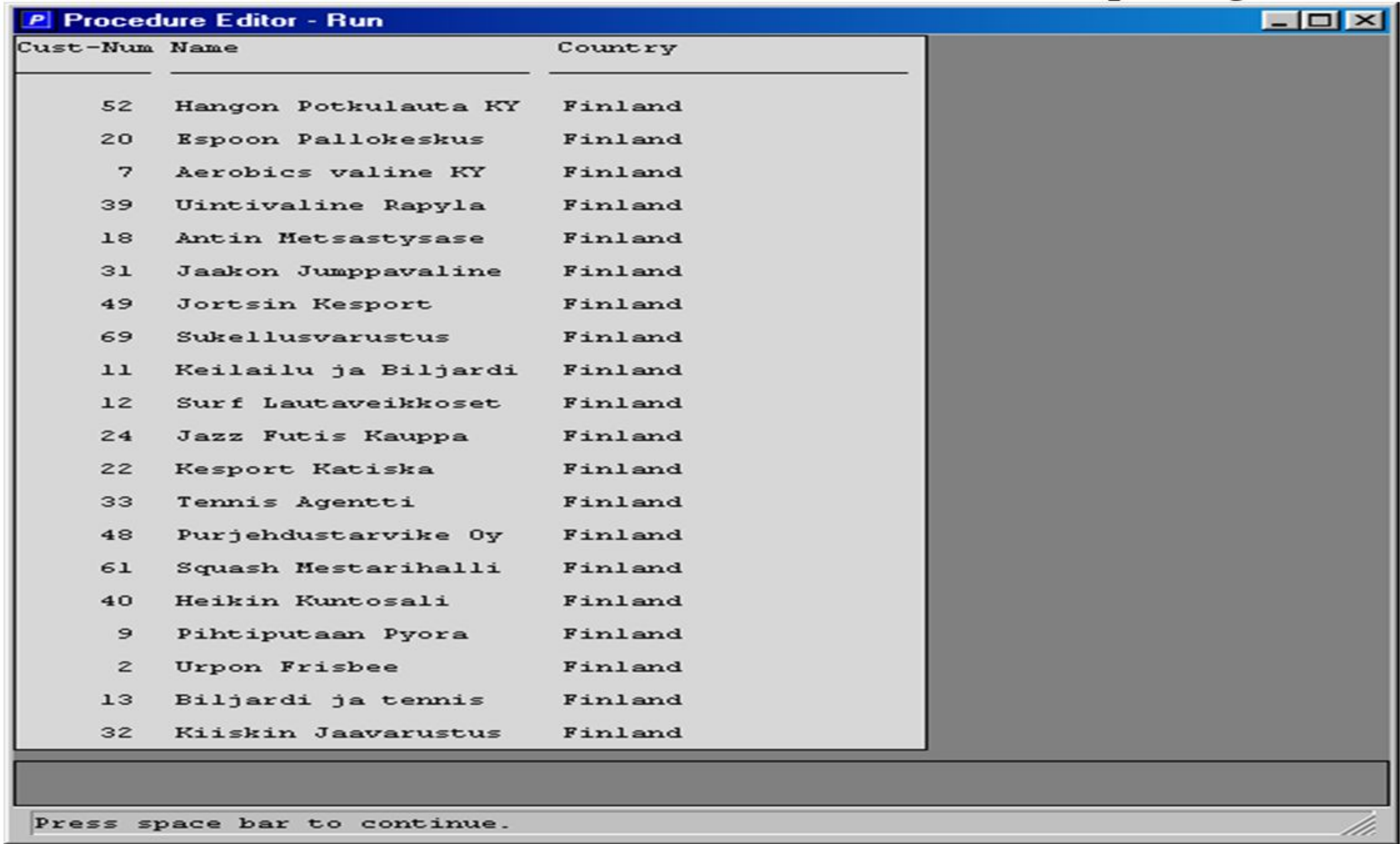
The screenshot shows a window titled "Procedure Editor - Run" with a form for entering customer data. The form fields are as follows:

Cust-Num:	1
Country:	USA
Name:	Lift Line Skiing
Address:	276 North Street
Address2:	
City:	Boston
State:	MA
Postal-Code:	02114
Contact:	Gloria Shepley
Phone:	(617) 450-0087
Sales-Rep:	HQM
Credit-Limit:	66.700
Balance:	42.568,00
Terms:	Net30
Discount:	35%
Comments:	This customer is on credit hold.

Enter data or press ESC to end.

1. Modifique o programa **l-chg1.p**, chamando-o de **l-chg2.p** para pedir o código do cliente e depois alterá-lo, utilize o comando UPDATE.
2. Modifique o programa **l-chg2.p**, chamando-o de **l-chg3.p** para fazer o mesmo que o anterior, mas sem utilizar o comando UPDATE, utilize o comando SET.
3. Modifique o programa **l-chg3.p**, chamando-o de **l-chg4.p** para fazer o mesmo que o anterior, mas sem utilizar os comandos UPDATE e SET.

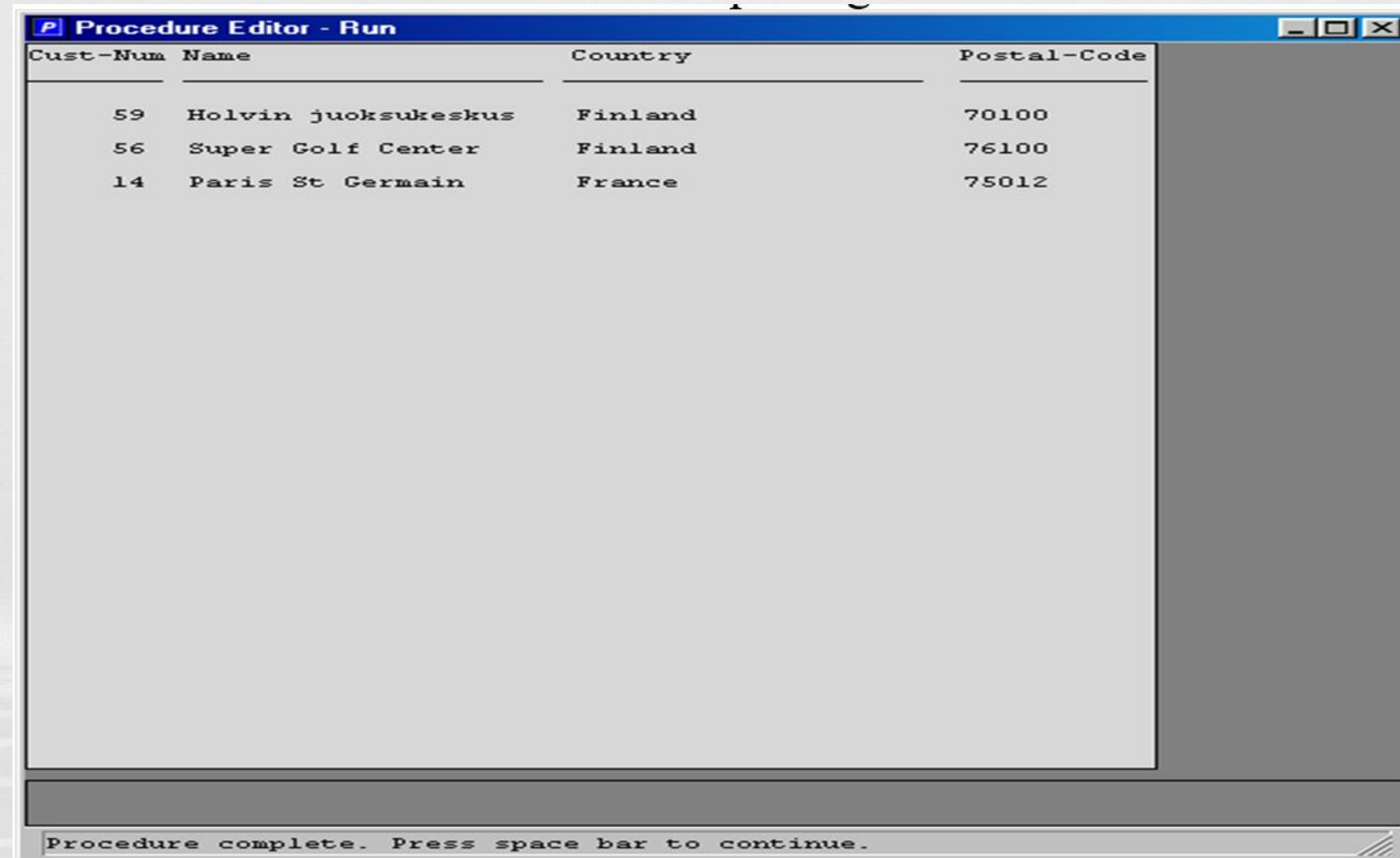
4. Escreva um programa **l-disp.p** que mostre o número, o nome e o país de todos os clientes cujo país começa com a letra "F". A tela deve se assemelhar a que segue:



The screenshot shows a window titled "P Procedure Editor - Run". Inside, there is a table with three columns: "Cust-Num", "Name", and "Country". The table lists 20 customers, all of whom are from Finland. The text is displayed in a monospaced font. At the bottom of the window, there is a status bar that says "Press space bar to continue."

Cust-Num	Name	Country
52	Hangon Potkulauta KY	Finland
20	Espoon Pallokeskus	Finland
7	Aerobics valine KY	Finland
39	Uintivaline Rapyta	Finland
18	Antin Metsastysase	Finland
31	Jaakon Jumppavaline	Finland
49	Jortsin Kesport	Finland
69	Sukellusvarustus	Finland
11	Keilailu ja Biljardi	Finland
12	Surf Lautaveikkoset	Finland
24	Jazz Futis Kauppa	Finland
22	Kesport Katiska	Finland
33	Tennis Agentti	Finland
48	Purjehdustarvike Oy	Finland
61	Squash Mestarihalli	Finland
40	Heikin Kuntosali	Finland
9	Pihtiputaan Pyora	Finland
2	Urpon Frisbee	Finland
13	Biljardi ja tennis	Finland
32	Kiiskin Jaavarustus	Finland

5. Modifique o programa **l-disp.p**, chamando-o de **l-disp1.p**, de modo que mostre o número, o nome, o país e o código postal de todos os clientes cujo país comece com a letra "F" e o código postal comece com "7". A tela deve ser assemelhar a que segue:

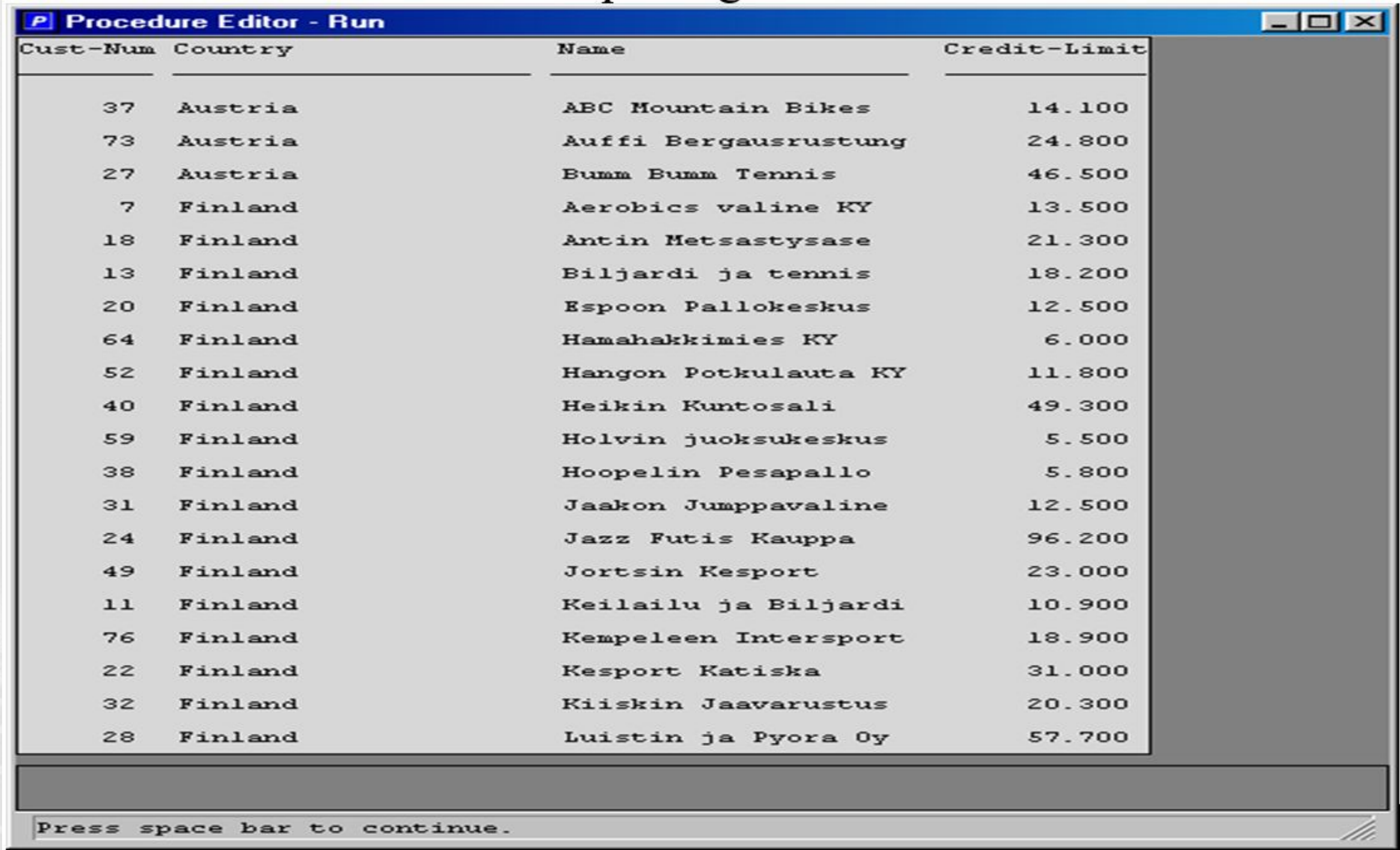


The screenshot shows a window titled "Procedure Editor - Run" with a table of customer data. The table has four columns: Cust-Num, Name, Country, and Postal-Code. The data is as follows:

Cust-Num	Name	Country	Postal-Code
59	Holvin juoksukeskus	Finland	70100
56	Super Golf Center	Finland	76100
14	Paris St Germain	France	75012

At the bottom of the window, a status bar reads: "Procedure complete. Press space bar to continue."

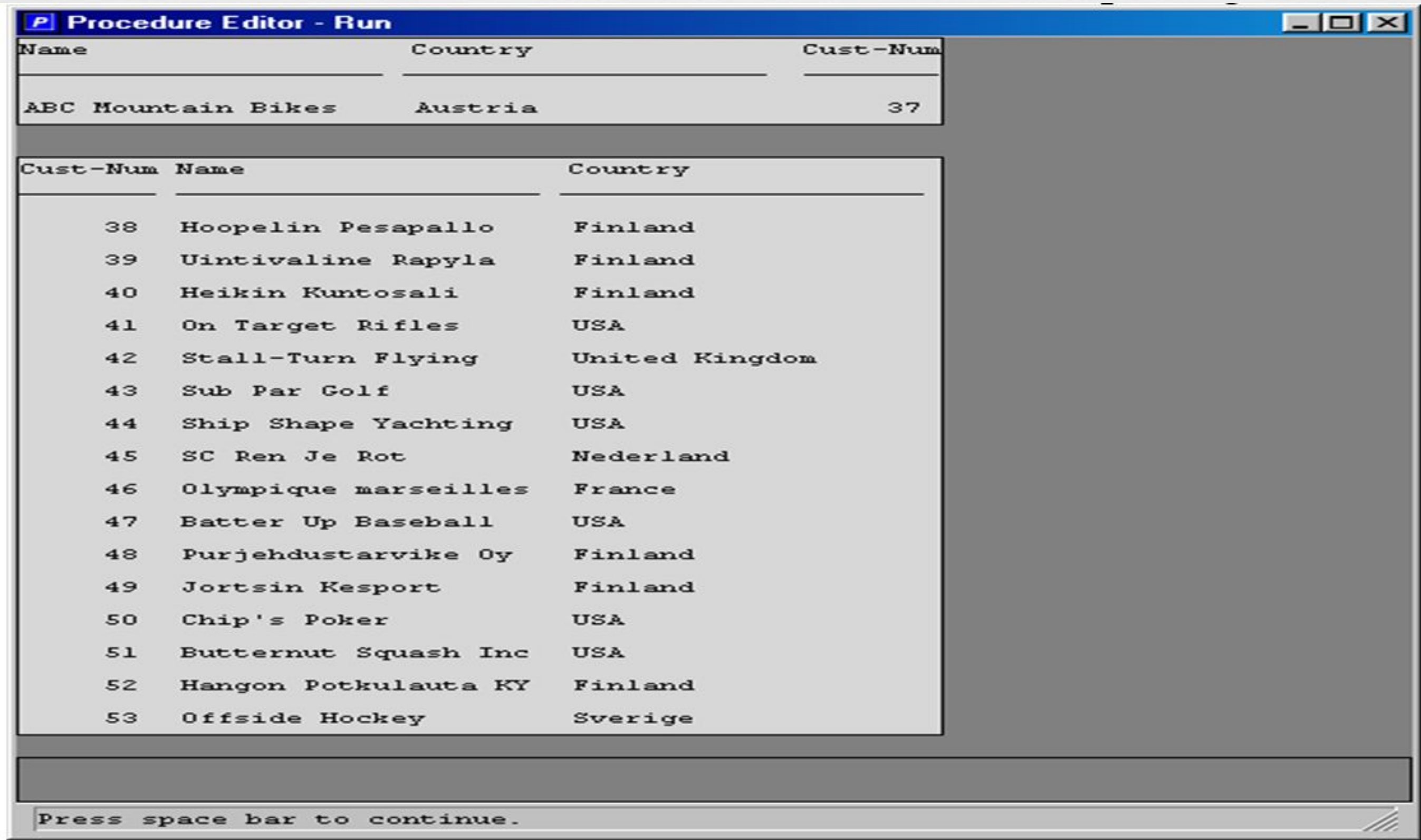
6. Escreva um programa **l-sort1.p** que mostre o número, o país, nome e o limite de crédito de cada cliente. Os dados deverão ser mostrados em ordem de país, e para os países iguais, deve ser classificado por nome. A tela deve se assemelhar a que segue:



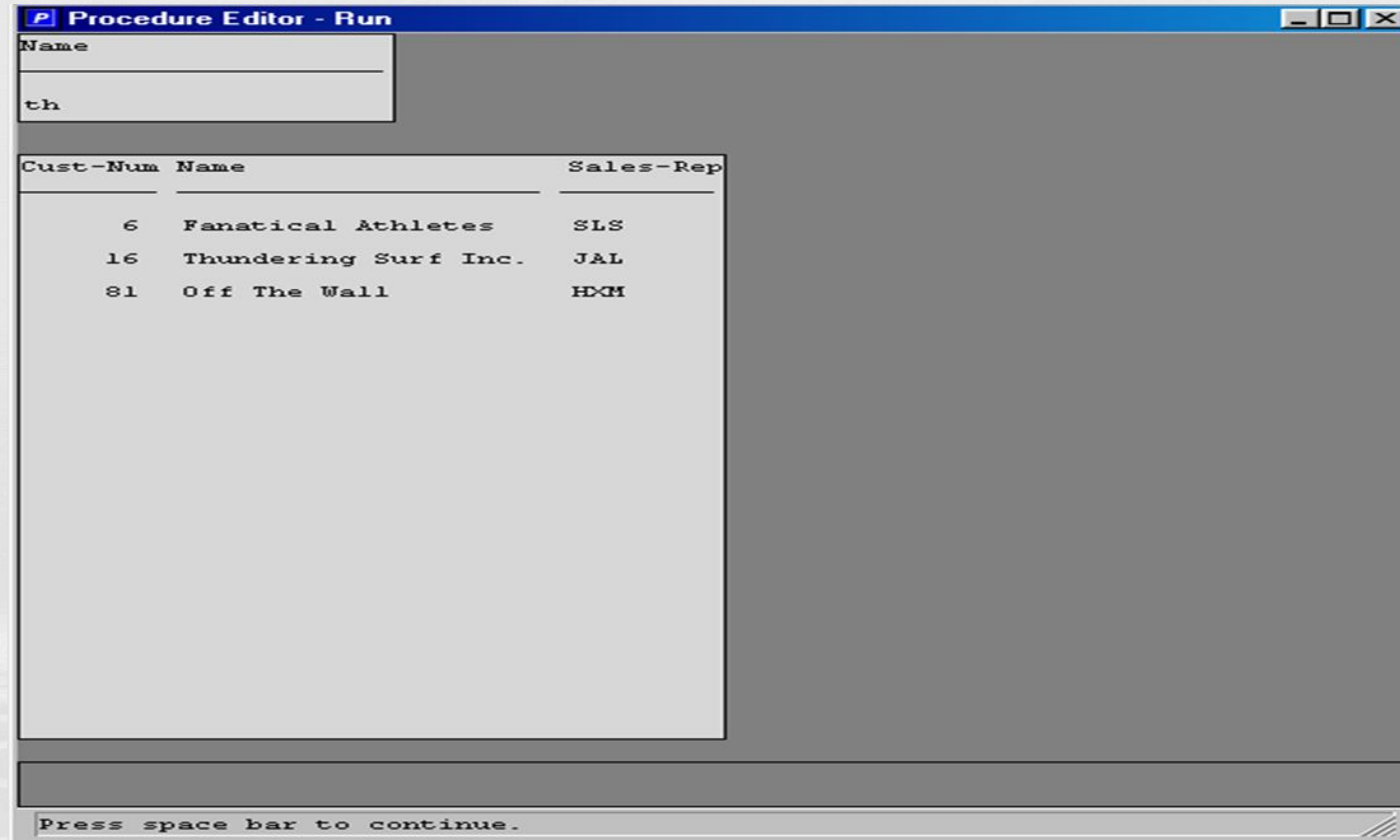
The screenshot shows a window titled "P Procedure Editor - Run" with a table of customer data. The table has four columns: "Cust-Num", "Country", "Name", and "Credit-Limit". The data is sorted by country, with Austria first and Finland second. Within each country, the entries are sorted alphabetically by name. The table contains 20 rows of data. At the bottom of the window, there is a status bar that says "Press space bar to continue."

Cust-Num	Country	Name	Credit-Limit
37	Austria	ABC Mountain Bikes	14.100
73	Austria	Auffi Bergausrustung	24.800
27	Austria	Bumm Bumm Tennis	46.500
7	Finland	Aerobics valine KY	13.500
18	Finland	Antin Metsastysase	21.300
13	Finland	Biljardi ja tennis	18.200
20	Finland	Espoon Pallokeskus	12.500
64	Finland	Hamahakkimies KY	6.000
52	Finland	Hangon Potkulauta KY	11.800
40	Finland	Heikin Kuntosali	49.300
59	Finland	Holvin juoksukeskus	5.500
38	Finland	Hoopelin Pesapallo	5.800
31	Finland	Jaakon Jumpsavoline	12.500
24	Finland	Jazz Futis Kauppa	96.200
49	Finland	Jortsin Kesport	23.000
11	Finland	Keilailu ja Biljardi	10.900
76	Finland	Kempeleen Intersport	18.900
22	Finland	Kesport Katiska	31.000
32	Finland	Kiiskin Jaavarustus	20.300
28	Finland	Luistin ja Pyora Oy	57.700

7. Escreva um programa **l-names.p** que peça o nome do cliente, encontre o primeiro cliente cujo nome inicie com o informado e mostre o código, o nome e o país. Então o programa deverá mostrar todos os outros clientes que tenha o código maior que o cliente encontrado. Tente não utilizar variáveis nem o comando FOR EACH. A tela deve se assemelhar a que segue:



8. Escreva um programa **l-match.p** que peça uma string (utilize o campo nome do cliente) e mostre o código, nome e representante de todo os clientes que possuam em qualquer posição do nome a string informada. A tela deve assemelhar-se a que segue:



The screenshot shows a window titled "Procedure Editor - Run". At the top, there is a text input field labeled "Name" containing the text "th". Below this, a table displays the results of a search. The table has three columns: "Cust-Num", "Name", and "Sales-Rep". The data rows are:

Cust-Num	Name	Sales-Rep
6	Fanatical Athletes	SLS
16	Thundering Surf Inc.	JAL
81	Off The Wall	HXM

At the bottom of the window, a status bar contains the text "Press space bar to continue."

1. Escreva um programa **l-cred2.p** que peça o código do cliente e mostre o número, o nome, e limite de crédito. Então o programa deverá mostrar os mesmos dados para todos os clientes cujo limite de crédito seja igual ou maior que o informado, classificado em ordem decrescente por limite de crédito.

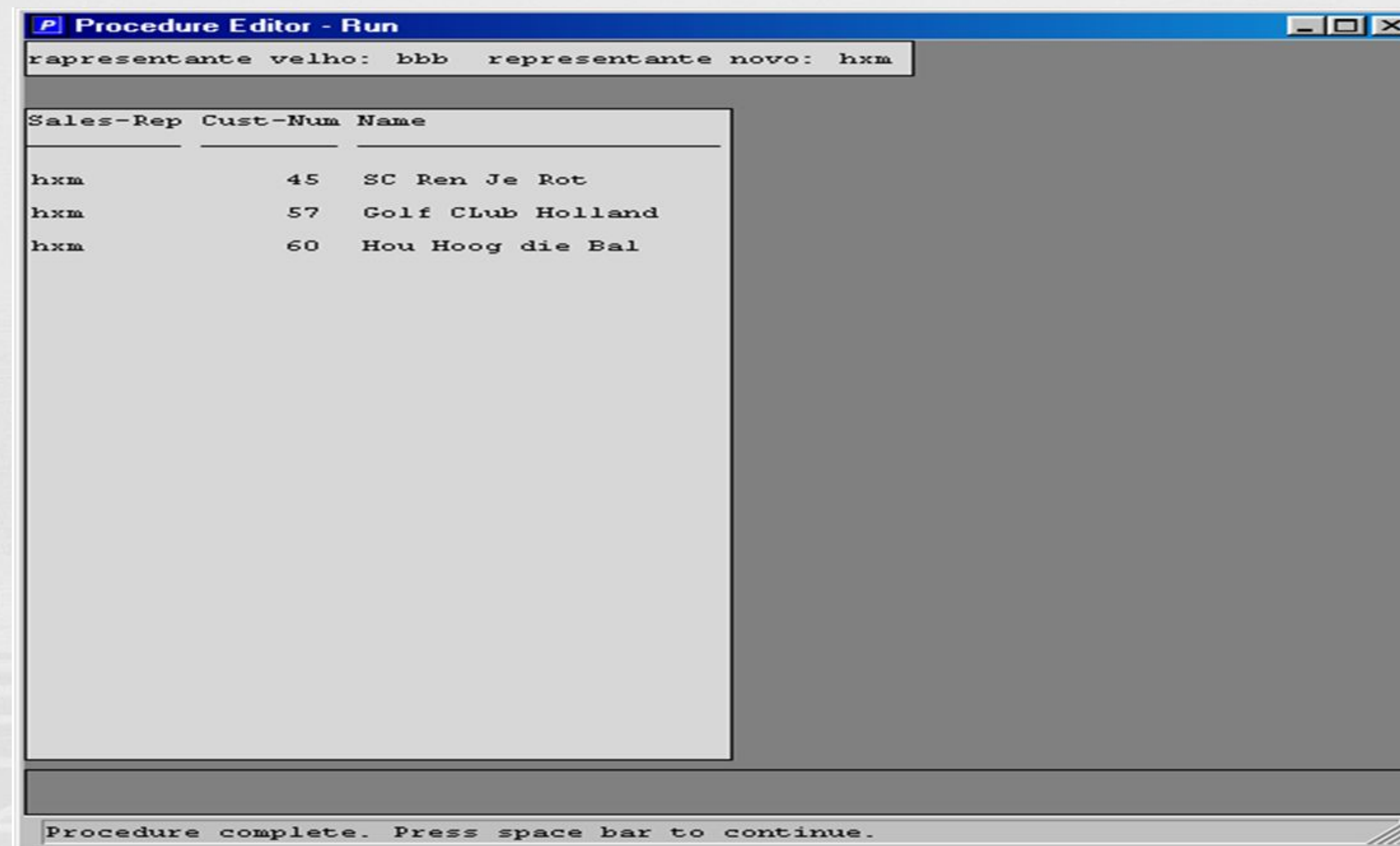
Dica: Depois de encontrar o cliente original, grave o limite de crédito em uma variável. Use esta variável quando for localizar os outros clientes.

P Procedure Editor - Run		
Cust-Num	Name	Credit-Limit
10	Just Joggers Limited	22.000
Cust-Num	Name	Credit-Limit
24	Jazz Futis Kauppa	96.200
66	First Down Football	82.200
3	Hoops Croquet Co.	75.000
83	Fallen Arch Running	73.500
1	Lift Line Skiing	66.700
41	On Target Rifles	58.000
34	Quick Toss Lacrosse	58.000
28	Luistin ja Pyora Oy	57.700
80	StickyWicket Cricket	54.800
30	Fast Flipper Pinball	53.400
72	Birdy's Badminton	52.900
65	Lagt Kort Ligger	51.700
40	Heikin Kuntosali	49.300
26	Bulls Eye Sports	47.700
27	Bumm Bumm Tennis	46.500
75	Hard Knocks Skating	46.400

Press space bar to continue.

2. Escreva um programa **l-xferj.p** que permita associar um novo representante a todos os clientes de um determinado representante. O programa deve pedir o representante atual e o novo representante e buscar todos os clientes que possuam o representante atual. A medida que você lê os clientes você deve mostrar o número e nome do cliente, associar o novo representante e mostrá-lo. A tela deve ser assemelhar a que segue:

Dica: Use duas variáveis, uma para o representante atual e outra para o novo. Lembre-se de usar UPDATE para entrar com os valores nas variáveis.



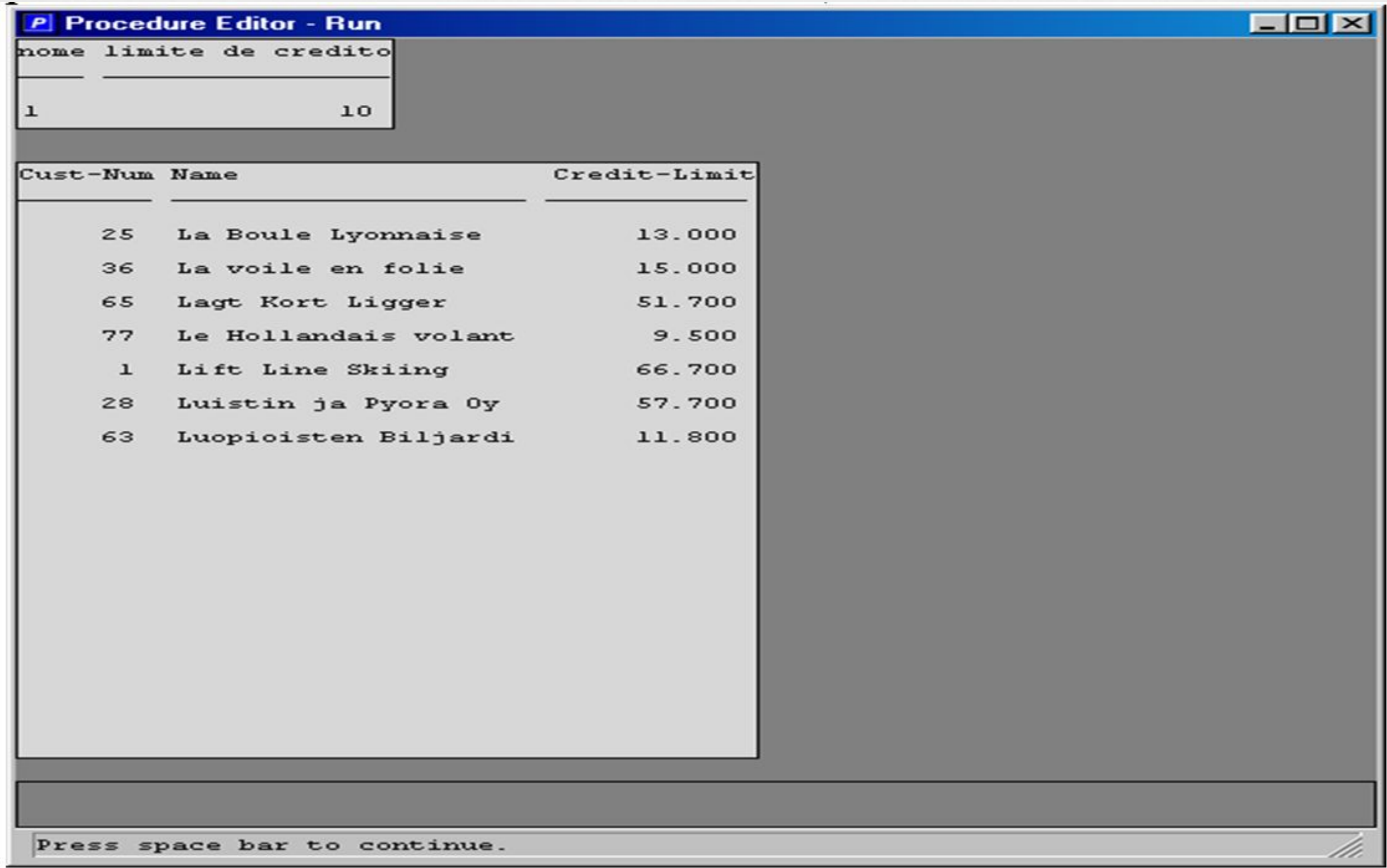
```
Procedure Editor - Run
representante velho: bbb representante novo: hxm

Sales-Rep  Cust-Num  Name
-----
hxm        45       SC Ren Je Rot
hxm        57       Golf CLub Holland
hxm        60       Hou Hoog die Bal

Procedure complete. Press space bar to continue.
```


3. Escreva um programa **l-disp2.p** que solicite o primeiro caracter do nome e o limite de crédito do cliente. Em seguida deve mostrar o número, o nome e o limite de crédito de todos os clientes cujo nome comece com a letra informada e que tenha o limite de crédito maior do que o informado. A tela deve se assemelhar a que segue:

Dica: Use duas variáveis, uma para caracter e outra para limite de crédito. Lembre-se de usar UPDATE para entrar com os valores nas variáveis.



4. Modifique o programa **l-chg2.p**, chamando-o de **l-chg5.p**, para usar uma variável que peça o código de um cliente e encontre o registro desse cliente. A tela deve assemelhar-se a que segue:

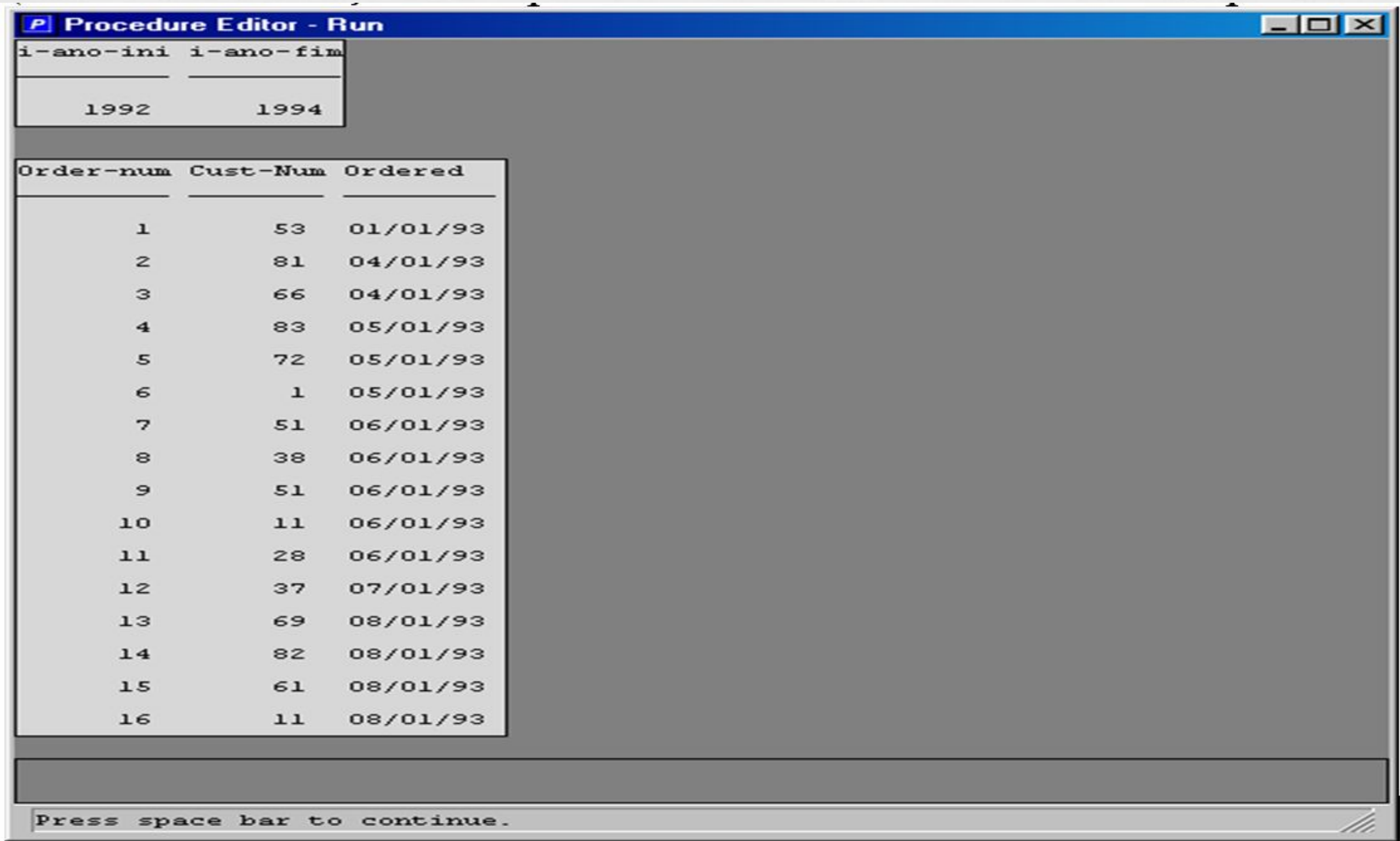
Procedure Editor - Run

Cust-Num:	12
Country:	Finland
Name:	Surf Lautaveikkoset
Address:	Venemestarinkatu 35
Address2:	
City:	Salo
State:	Varsinais-Suomi
Postal-Code:	24240
Contact:	Veikko Lehtisoini
Phone:	(51) 254 2180
Sales-Rep:	DKP
Credit-Limit:	6.500
Balance:	0,00
Terms:	Net30
Discount:	35%
Comments:	

Enter data or press ESC to end.

5. Escreva um programa **l-orders.p** que solicite ao usuário o valor de dois anos, e então mostre todos os pedidos cuja data do pedido esteja entre 1/1 do primeiro ano e 31/12 do segundo ano. Garanta que o usuário possa repetir a consulta tantas vezes quantas quiser sem sair do programa. A tela deve se assemelhar a que segue:

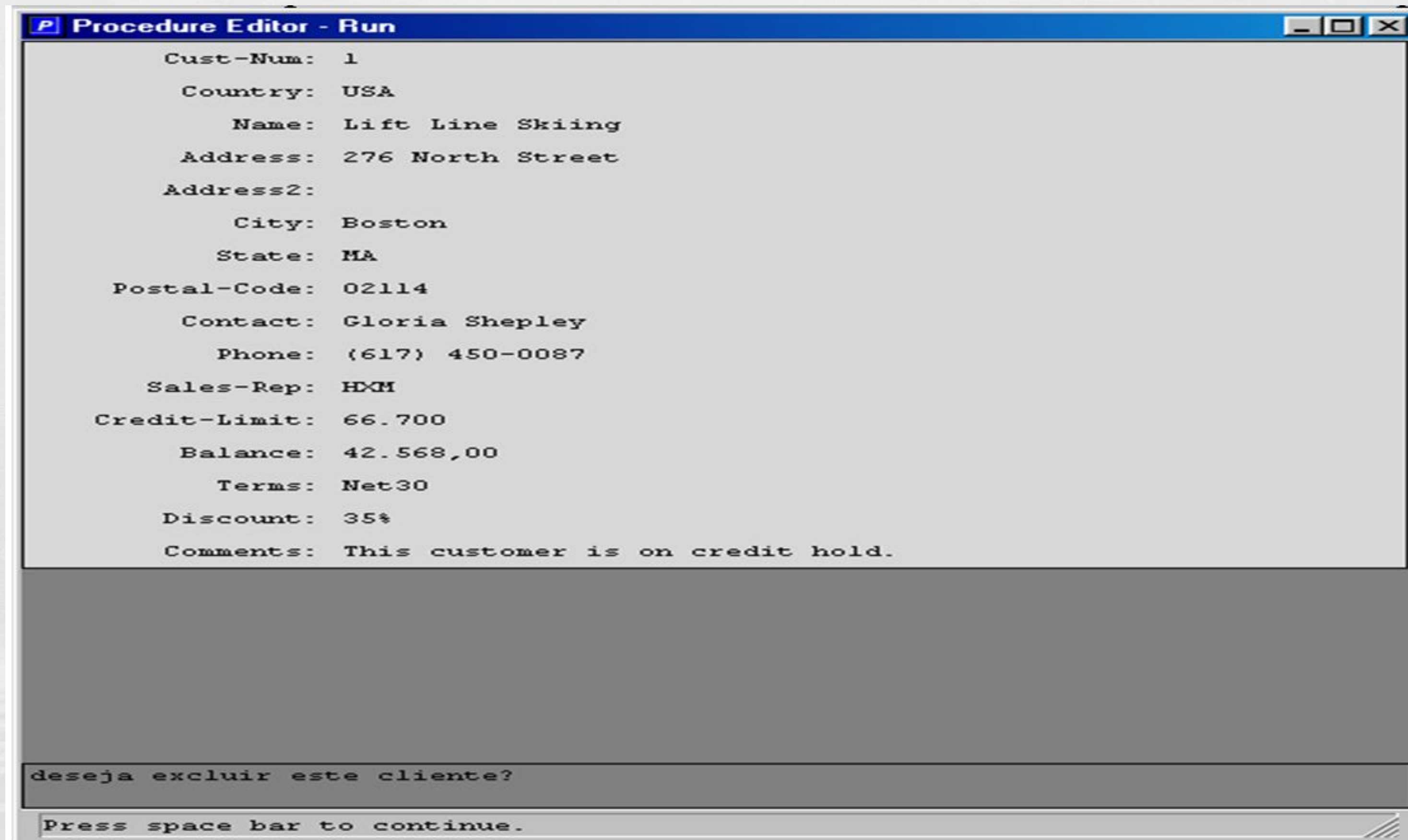
Dica: Use a função date para converter os valores inteiros para data.



The screenshot shows a window titled "Procedure Editor - Run". At the top, there are two input fields labeled "i-ano-ini" and "i-ano-fim" with the values "1992" and "1994" respectively. Below these fields is a table with three columns: "Order-num", "Cust-Num", and "Ordered". The table contains 16 rows of data. At the bottom of the window, there is a text box that says "Press space bar to continue."

Order-num	Cust-Num	Ordered
1	53	01/01/93
2	81	04/01/93
3	66	04/01/93
4	83	05/01/93
5	72	05/01/93
6	1	05/01/93
7	51	06/01/93
8	38	06/01/93
9	51	06/01/93
10	11	06/01/93
11	28	06/01/93
12	37	07/01/93
13	69	08/01/93
14	82	08/01/93
15	61	08/01/93
16	11	08/01/93

1. Escreva um programa **l-del.p** que mostre cada registro de cliente, um de cada vez, e que pergunte se deseja excluí-lo ou não. Se responder sim, o programa deve excluir o cliente, mostrar uma mensagem indicando o que cliente foi excluído. A tela deve assemelhar-se a que segue:

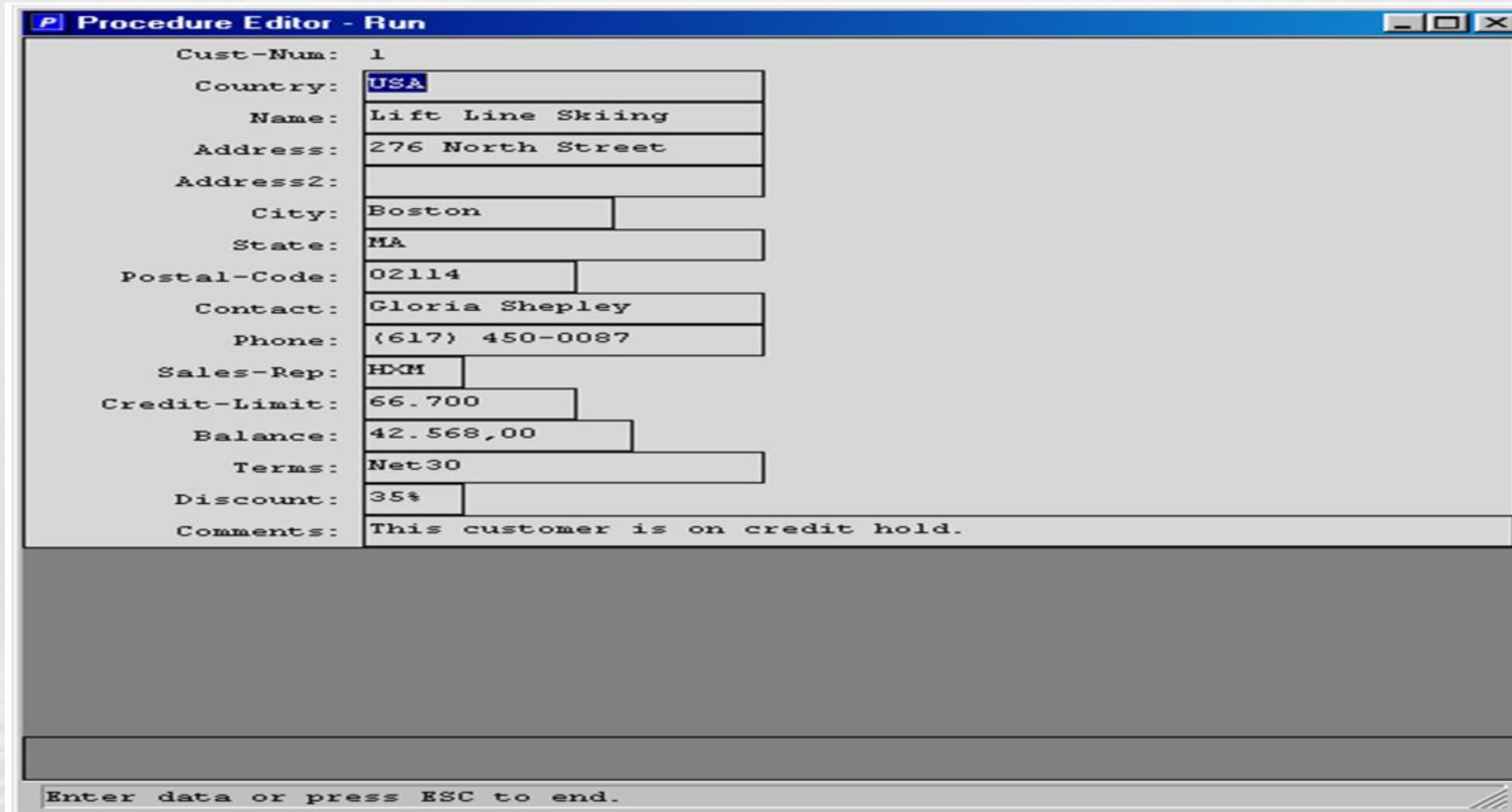


```
P Procedure Editor - Run

Cust-Num: 1
Country: USA
Name: Lift Line Skiing
Address: 276 North Street
Address2:
City: Boston
State: MA
Postal-Code: 02114
Contact: Gloria Shepley
Phone: (617) 450-0087
Sales-Rep: HCM
Credit-Limit: 66.700
Balance: 42.568,00
Terms: Net30
Discount: 35%
Comments: This customer is on credit hold.

deseja excluir este cliente?
Press space bar to continue.
```


2. Escreva um programa **l-addchg.p** que permita incluir um cliente ou alterar um registro de cliente existente. Esteja certo de que não possa alterar o código de um cliente existente. A tela deve assemelhar-se a que segue:

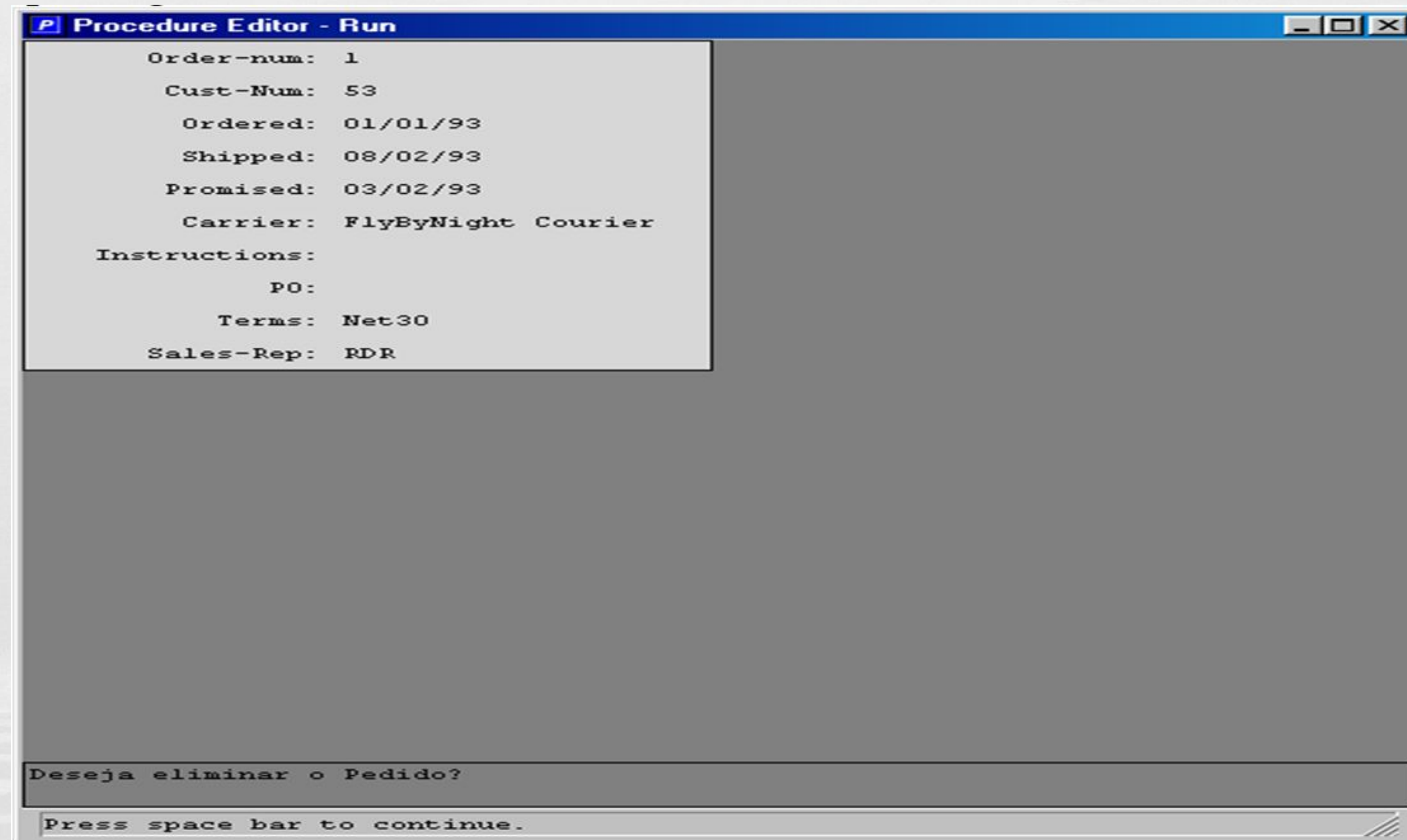


The screenshot shows a window titled "Procedure Editor - Run" with a form for entering customer data. The form fields are as follows:

Cust-Num:	1
Country:	USA
Name:	Lift Line Skiing
Address:	276 North Street
Address2:	
City:	Boston
State:	MA
Postal-Code:	02114
Contact:	Gloria Shepley
Phone:	(617) 450-0087
Sales-Rep:	HXM
Credit-Limit:	66.700
Balance:	42.568,00
Terms:	Net30
Discount:	35%
Comments:	This customer is on credit hold.

Enter data or press ESC to end.

3. Escreva um programa **l-orderdel.p** que deixe você eliminar pedidos. Garanta que o programa confirme a eliminação e, se for confirmada, elimina também todas as linhas do pedido. A tela deve assemelhar-se a que segue:

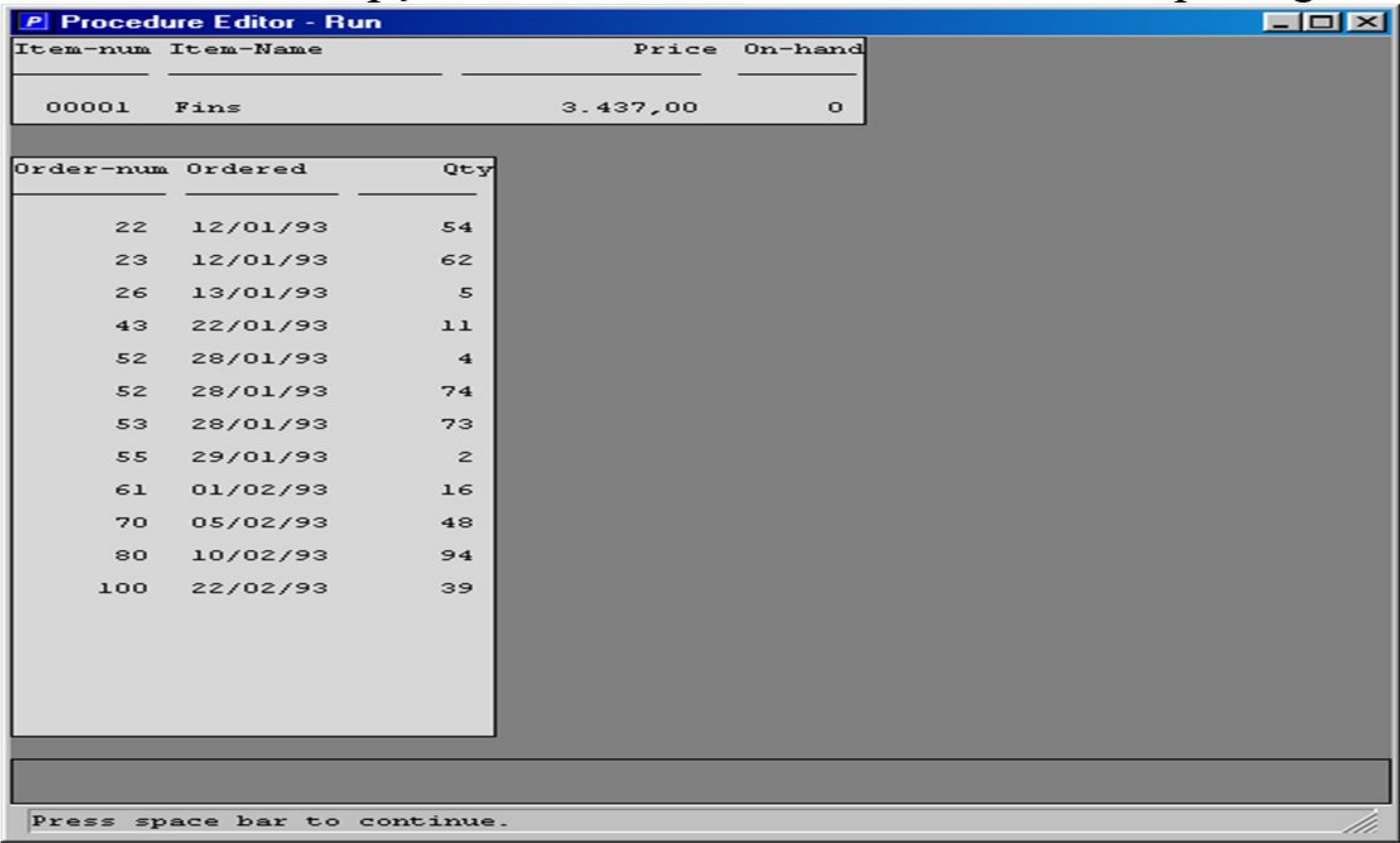


The screenshot shows a window titled "Procedure Editor - Run". Inside, there is a list of order details:

- Order-num: 1
- Cust-Num: 53
- Ordered: 01/01/93
- Shipped: 08/02/93
- Promised: 03/02/93
- Carrier: FlyByNight Courier
- Instructions:
- PO:
- Terms: Net30
- Sales-Rep: RDR

At the bottom of the window, there is a prompt: "Deseja eliminar o Pedido?" and a instruction: "Press space bar to continue."

4. Escreva um programa **l-disp3.p** que solicite o número do item e mostre o seu nome, preço e quantidade disponível. O programa deve também mostrar o número, data e quantidade de cada pedido que contenha este item. Use a opção **OF**. A tela deve assemelhar-se a que segue:



The screenshot shows a window titled "Procedure Editor - Run". Inside, there are two tables. The first table has columns "Item-num", "Item-Name", "Price", and "On-hand". The second table has columns "Order-num", "Ordered", and "Qty". At the bottom of the window, it says "Press space bar to continue."

Item-num	Item-Name	Price	On-hand
00001	Fins	3.437,00	0

Order-num	Ordered	Qty
22	12/01/93	54
23	12/01/93	62
26	13/01/93	5
43	22/01/93	11
52	28/01/93	4
52	28/01/93	74
53	28/01/93	73
55	29/01/93	2
61	01/02/93	16
70	05/02/93	48
80	10/02/93	94
100	22/02/93	39

Press space bar to continue.

- 5. Modifique o procedimento **l-disp3.p**, chamando-o de **l-disp4.p** de modo que tenha a mesma funcionalidade, mas sem usar a opção **OF**.
- 6. Modifique o procedimento **l-disp3.p**, chamando-o de **l-disp5.p** de modo que mostre os pedidos onde o item aparece classificados por quantidade (ordem decrescente). A tela deve assemelhar-se a que segue:

P Procedure Editor - Run			
Item-num	Item-Name	Price	On-hand
00001	Fins	3.437,00	0

Order-num	Ordered	Qty
80	10/02/93	94
52	28/01/93	74
53	28/01/93	73
23	12/01/93	62
22	12/01/93	54
70	05/02/93	48
100	22/02/93	39
61	01/02/93	16
43	22/01/93	11
26	13/01/93	5
52	28/01/93	4
55	29/01/93	2

Press space bar to continue.

7. Modifique o procedimento **l-disp5.p**, chamando-o de **l-disp6.p** de modo que dentro de cada nível de quantidade classifique por data do pedido.

Dica: use um único FOR EACH para ler os registros de order e orderline

8. Escreva um programa **l-disp7.p** que para cada representante mostre o nome do representante, para cada cliente do representante mostre o número e o nome do cliente, para cada pedido do cliente mostre a data do pedido e para cada item do pedido mostre a quantidade e o nome do item. A tela deve assemelhar-se a que segue:

Procedure Editor - Run

Rep-Name

Pitt , Dirk K.

Cust-Num Name

2 Urpon Frisbee

Ordered	Qty	Item-Name
16/02/93	98	Swimming trunks
16/02/93	75	Bowling Shoes
16/02/93	80	Volleyball
03/03/93	74	Hockey puck
03/03/93	98	Ski Gloves
03/03/93	93	Frisbee
26/03/93	22	Squash Glove
26/03/93	96	Leotard - Black
26/03/93	78	Tennis Shorts
26/03/93	1	Bowling Bag
26/03/93	44	Skis
26/03/93	34	Sneakers

Press space bar to continue.

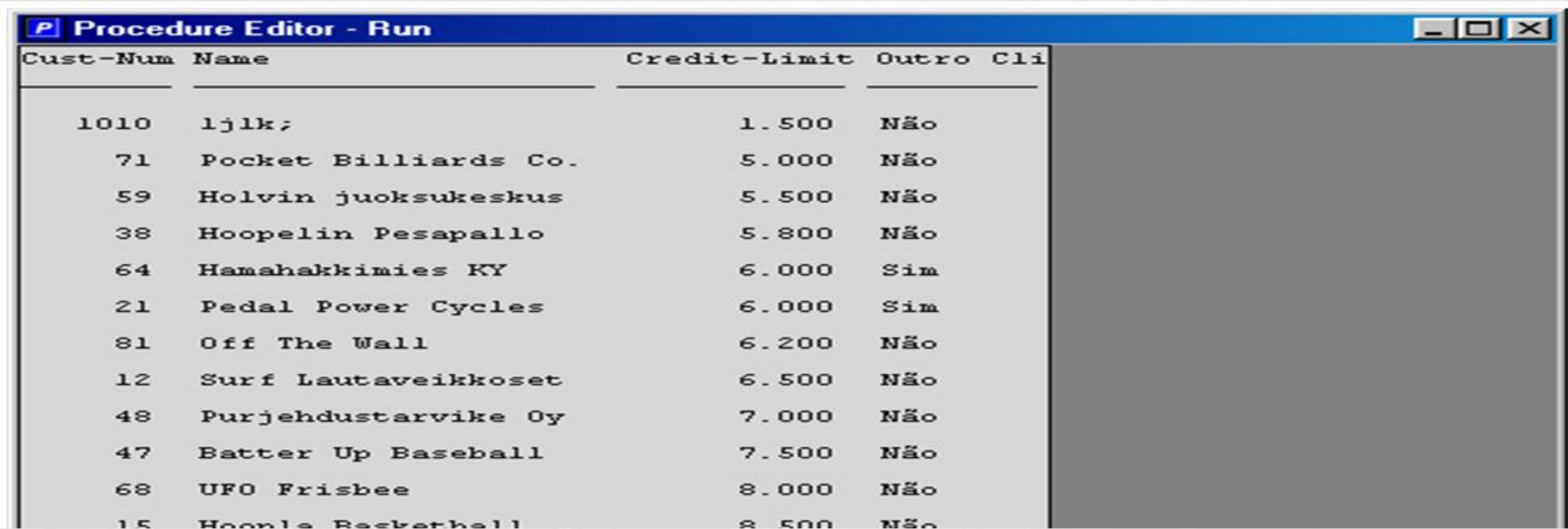
9. Modifique o programa **l-del.p**, chamando-o de **l-del2.p**, de modo que verifique se o cliente pode ser eliminado. Se existir algum pedido para o cliente mostrar uma mensagem e não permitir a eliminação do cliente.

Dica: Use a declaração NEXT.

10. Modifique o programa **l-del2.p**, chamando-o de **l-del3.p**, de modo que se o cliente teve algum pedido pergunte se os pedidos também devem ser eliminados. Se confirmar eliminar o cliente e seus pedidos. Explique o que aconteceu quando confirmar a eliminação de todos os pedidos de um cliente.

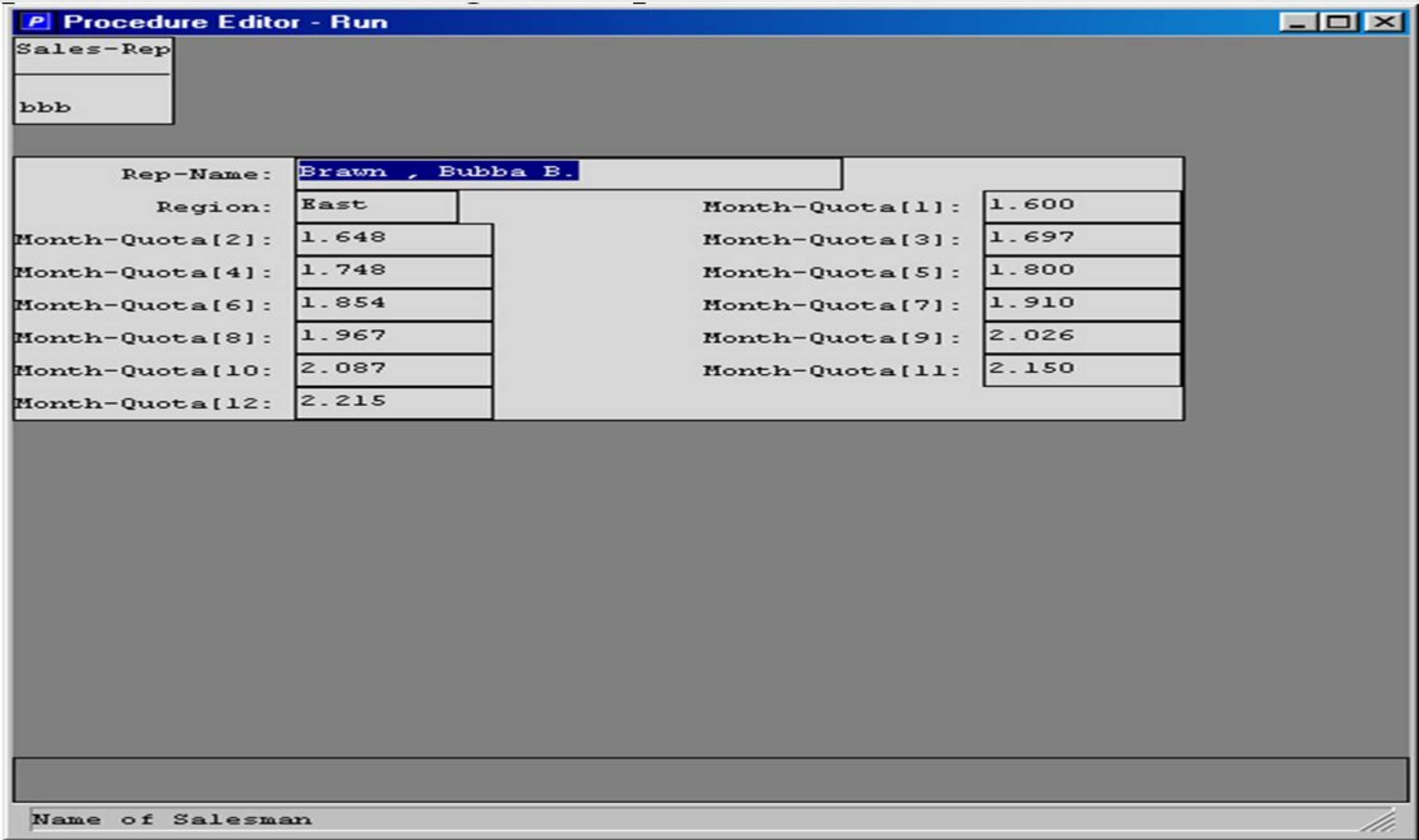
11 – Escreva um programa **l-buffer.p** que mostre o código, nome e crédito limite de todos os clientes, classificado por crédito limite. O programa deverá também mostrar uma coluna “Outro Cli” que indica que existe outro cliente com o mesmo limite de crédito. A tela deve assemelhar-se com a que segue:

Dica: Utilize o comando BUFFER para ler outro cliente. **Obs:** Garanta que o outro cliente que ele encontrar não seja o mesmo que ele está lendo.



Cust-Num	Name	Credit-Limit	Outro Cli
1010	ljljk;	1.500	Não
71	Pocket Billiards Co.	5.000	Não
59	Holvin juoksukeskus	5.500	Não
38	Hoopelin Pesapallo	5.800	Não
64	Hamahakkimies KY	6.000	Sim
21	Pedal Power Cycles	6.000	Sim
81	Off The Wall	6.200	Não
12	Surf Lautaveikkoset	6.500	Não
48	Purjehdustarvike Oy	7.000	Não
47	Batter Up Baseball	7.500	Não
68	UFO Frisbee	8.000	Não
15	Hoonla Basketball	8.500	Não

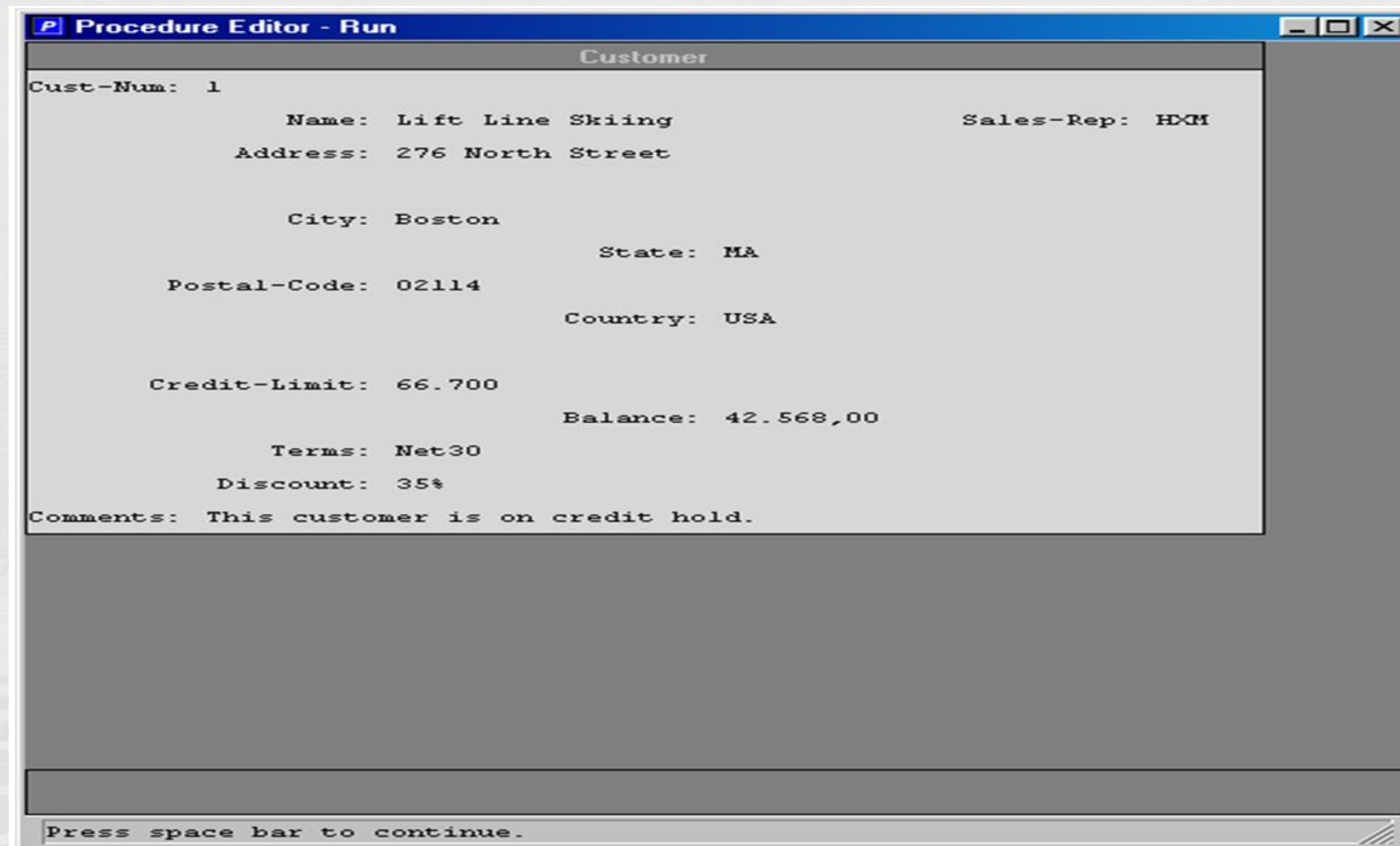
1. Escreva um programa **l-updrep.p**. para adicionar ou alterar registros da tabela de representante. Este programa deverá utilizar três frames:
- a) Para solicitar o código do representante
 - b) Para modificar as informações do representante
 - c) Para perguntar por outro representante
- Use o comando REPEAT para permitir a alteração de diversos representantes.
- Obs:** O programa não deve permitir alterar o código do representante. A tela deve assemelhar-se a que segue:



Sales-Rep	
bbb	
Rep-Name:	Brawn, Bubba B.
Region:	East
Month-Quota[1]:	1.600
Month-Quota[2]:	1.648
Month-Quota[3]:	1.697
Month-Quota[4]:	1.748
Month-Quota[5]:	1.800
Month-Quota[6]:	1.854
Month-Quota[7]:	1.910
Month-Quota[8]:	1.967
Month-Quota[9]:	2.026
Month-Quota[10]:	2.087
Month-Quota[11]:	2.150
Month-Quota[12]:	2.215

Name of Salesman

2. Escreva um programa **I-disord.p** que solicite o número de um pedido e então mostre todas as linhas deste pedido. Quando as linhas do pedido aparecerem a frame na qual o usuário informou o número do pedido deverá ser apagada e vice-versa.
3. Escreva um programa **I-dcust.p** que mostre todos os clientes da tabela de cliente. Somente um cliente deve aparecer por vez na tela. Utilize o comando FORM. A tela deve assemelhar-se a que segue:

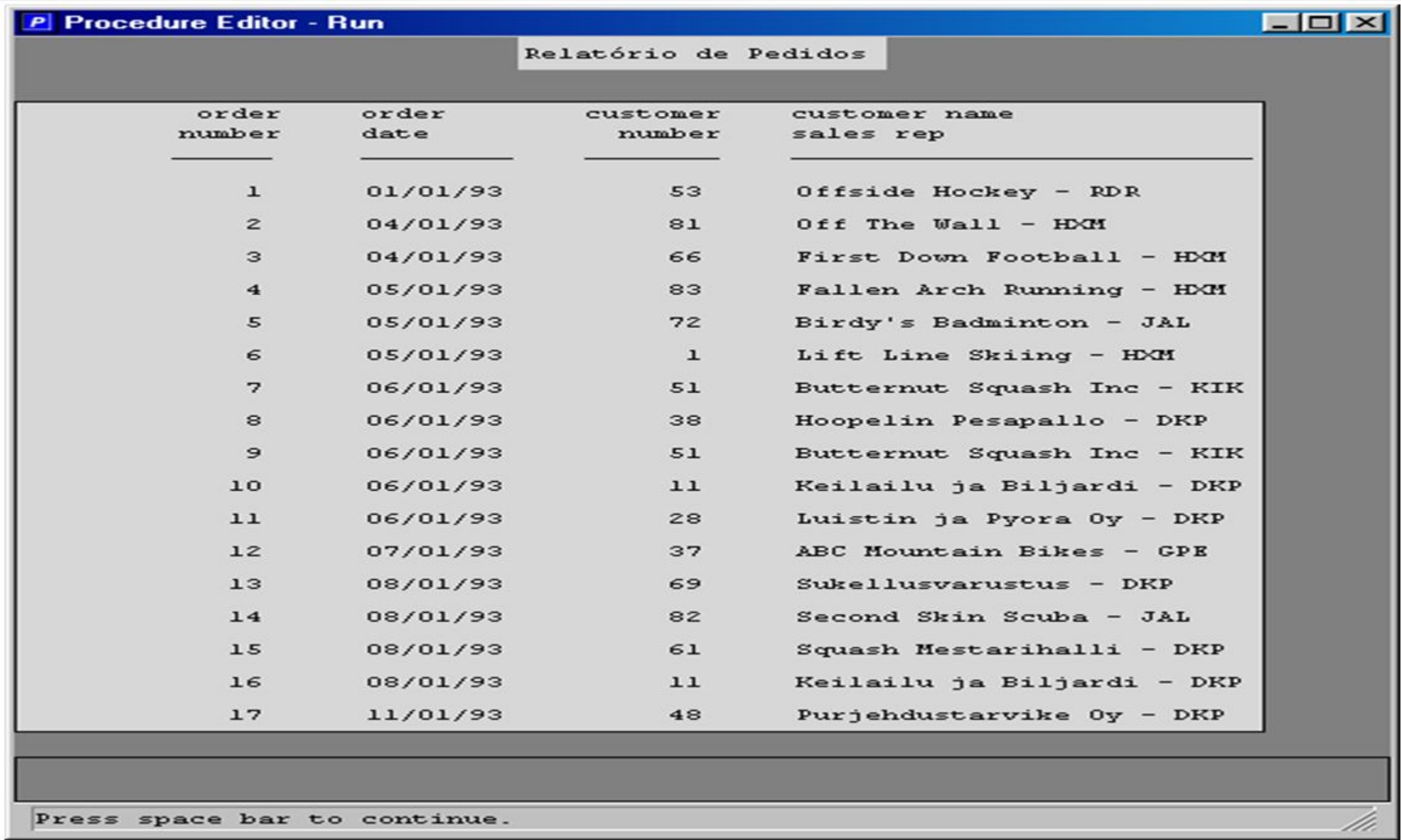


The screenshot shows a window titled "Procedure Editor - Run" with a "Customer" form. The form contains the following fields and values:

Customer	
Cust-Num: 1	
Name: Lift Line Skiing	Sales-Rep: HXM
Address: 276 North Street	
City: Boston	
State: MA	
Postal-Code: 02114	Country: USA
Credit-Limit: 66.700	Balance: 42.568,00
Terms: Net30	
Discount: 35%	
Comments: This customer is on credit hold.	

Press space bar to continue.

4. Modifique o programa **l-dcust.p**, chamando-o de **l-ucust.p**, de modo que deixe você alterar as informações dos clientes. Garanta que o cursor siga a seguinte seqüência: endereço, nome, cidade, estado, país, código postal, balanço, limite de crédito, desconto, termos e comentário.
5. Escreva um procedimento **l-col1.p** que leia todos os pedidos da tabela de pedidos. Para cada um leia o seu cliente, mostre o número do pedido, data do pedido, número do cliente, nome do cliente e representante. O nome do cliente e o representante devem ser concatenados. A tela deve assemelhar-se a que segue:



order number	order date	customer number	customer name	sales rep
1	01/01/93	53	Offside Hockey - RDR	
2	04/01/93	81	Off The Wall - HCM	
3	04/01/93	66	First Down Football - HCM	
4	05/01/93	83	Fallen Arch Running - HCM	
5	05/01/93	72	Birdy's Badminton - JAL	
6	05/01/93	1	Lift Line Skiing - HCM	
7	06/01/93	51	Butternut Squash Inc - KIK	
8	06/01/93	38	Hoopelin Pesapallo - DKP	
9	06/01/93	51	Butternut Squash Inc - KIK	
10	06/01/93	11	Keilailu ja Biljardi - DKP	
11	06/01/93	28	Luistin ja Pyora Oy - DKP	
12	07/01/93	37	ABC Mountain Bikes - GPE	
13	08/01/93	69	Sukellusvarustus - DKP	
14	08/01/93	82	Second Skin Scuba - JAL	
15	08/01/93	61	Squash Mestarihalli - DKP	
16	08/01/93	11	Keilailu ja Biljardi - DKP	
17	11/01/93	48	Purjehdustarvike Oy - DKP	

Press space bar to continue.

6. Modifique o procedimento **l-col1.p**, chamando-o de **l-col2.p** de modo que solicite uma data de pedido. Então o programa deverá mostrar com cor de message todos os pedidos cuja data seja mais antiga que a informada. O valor inicial da data deverá ser a data atual (Função TODAY). A tela deve assemelhar-se a que segue:

Procedure Editor - Run

Data do Pedido: 06/03/93

order number	order date	customer number	customer name sales rep
129	05/03/93	22	Kesport Katiska - DKP
130	05/03/93	6	Fanatical Athletes - SLS
131	05/03/93	33	Tennis Agentti - DKP
132	08/03/93	83	Fallen Arch Running - HXM
133	09/03/93	56	Super Golf Center - DKP
134	09/03/93	16	Thundering Surf Inc. - JAL
135	09/03/93	6	Fanatical Athletes - SLS
136	09/03/93	46	Olympique marseilles - DOS
137	10/03/93	42	Stall-Turn Flying - SLS
138	10/03/93	24	Jazz Futis Kauppa - DKP
139	10/03/93	42	Stall-Turn Flying - SLS
140	10/03/93	40	Heikin Kuntosali - DKP
141	10/03/93	7	Aerobics valine KY - DKP
142	11/03/93	75	Hard Knocks Skating - KIK
143	11/03/93	52	Hangon Potkulauta KY - DKP
144	11/03/93	24	Jazz Futis Kauppa - DKP

Press space bar to continue.

7. Escreva o programa **l-mltfrm.p** que mostre as informações dos clientes, dos seus pedidos, do seu representante e a quantidade de pedidos. Note que a cada novo cliente as outras duas frames são atualizadas. A tela deve assemelhar-se a que segue:

Procedure Editor - Run

Name	ord. num
Lift Line Skiing	5

Order-num	Ordered
6	05/01/93
36	19/01/93
79	10/02/93
177	30/03/93
185	05/04/93

Sales-Rep: HDM

Region: Sverige

Press space bar to continue.

1. Escreva um programa **l-include.p** que possua uma include genérica para mostrar dados de uma determinada tabela, onde sejam passados como parâmetros o nome da tabela e uma lista com 2 atributos.
2. Modifique o programa **l-include.p**, chamando-o de **l-inc2.p**, de modo que utilizando a mesma include mostre os clientes e os estados.

1. Escreva um programa **l-break1.p** que mostre o nome de todos os representantes, do clientes associados a cada representante mostre o nome, o balanço, o total, o máximo (maior) e a média do balanço dos clientes de cada representante. A tela deve assemelhar-se a que segue:

P Procedure Editor - Run		
Rep-Name	Name	Balance
Pitt , Dirk K.	Hangon Potkulauta KY	\$7.402
Pitt , Dirk K.	Super Golf Center	\$11.805
Pitt , Dirk K.	Holvin juoksukeskus	\$519
Pitt , Dirk K.	Squash Mestarihalli	\$27.999
Pitt , Dirk K.	Luopioisten Biljardi	\$9.517
Pitt , Dirk K.	Hamahakkimies KY	\$2.459
Pitt , Dirk K.	Sukellusvarustus	\$9.771
Pitt , Dirk K.	Kempeleen Intersport	\$11.516
		\$366.137 TOTAL
		\$61.767 MAX
		\$13.076 AVG
Donna Swindall	Paris St Germain	\$19.428
Donna Swindall	La Boule Lyonnaise	\$8.662
Donna Swindall	La voile en folie	\$2.592
Donna Swindall	Olympique marseilles	\$33.284
Donna Swindall	Le Hollandais volant	\$0
		\$63.966 TOTAL
Press space bar to continue.		

2. Modifique o programa **l-break1.p**, chamando-o de **l-break2.p** de modo que somente mostre o representante se for o primeiro e o último cliente do representante.

Dica: use as funções FIRST-OF e LAST-OF. A tela deve assemelhar-se a que segue:

P Procedure Editor - Run			
Rep-Name	Name	Balance	
Robert Roller	Game Set Match	3.373,00	
	Offside Hockey	7.851,00	
	Lagt Kort Ligger	40.439,00	
		51.663,00	TOTAL
		40.439,00	MAX
		17.221,00	AVG
Smith , Spike Louise	Go Fishing Ltd	689,00	
	Fanatical Athletes	37.697,00	
	Just Joggers Limited	16.621,00	
	High Tide Sailing	5.889,00	
	Bulls Eye Sports	23.235,00	
	Stall-Turn Flying	19.862,00	
	Spokes Cycles	868,00	
Smith , Spike Louise	U-Jump Parachuting	14.225,00	
		119.086,00	TOTAL
		37.697,00	MAX
Press space bar to continue.			

3. Escreva um programa l-rpt1.p que produza o seguinte relatório: para cada pedido, mostre o número do pedido, a data, o nome do cliente, o número do item, a quantidade e a descrição do item de todas as linhas do pedido. Classifique o relatório por número do cliente, número do pedido e número da linha. Envie o relatório para o arquivo l-rpt1.rpt com o tamanho de página igual a 20. O cabeçalho de cada página deve conter a esquerda, a data (today) e, centralizado, a constante "Relatório Detalhado de Pedidos". O relatório deve assemelhar-se ao que segue:

23/01/02		Relatório Detalhado de Pedidos				
Order-num	Ordered	Name			Item-num	Qty Item-Name
-----	-----	-----			-----	-----
6	05/01/93	Lift	Line	Skiing	00009	67 Swim Goggles
6	05/01/93	Lift	Line	Skiing	00009	12 Swim Goggles
6	05/01/93	Lift	Line	Skiing	00011	34 Lacrosse Stick
6	05/01/93	Lift	Line	Skiing	00019	20 Ski Wax - Red
6	05/01/93	Lift	Line	Skiing	00043	13 Frisbee
36	19/01/93	Lift	Line	Skiing	00003	62 Sweat Band
36	19/01/93	Lift	Line	Skiing	00041	80 Hockey puck
79	10/02/93	Lift	Line	Skiing	00026	5 Ski mask
79	10/02/93	Lift	Line	Skiing	00031	60 Sneakers
79	10/02/93	Lift	Line	Skiing	00054	2 Shin pads

4. Modifique o programa **l-rpt1.p**, chamando-o de **l-rpt2.p** para também mostrar o número da página no rodapé de cada página do relatório.
5. Uma vez por semana seu gerente necessita que você produza dois relatórios. Um listando o número, nome, cidade e país de todos os clientes, o outro listando o número, nome, representante, limite de crédito e o balanço de todos os clientes. Pelo fato de haverem muitos clientes, o seu gerente quer que você leia apenas uma vez a tabela de clientes para gerar os dois relatórios. Escreva um programa **l-rpt3.p** que produza estes relatórios:

Exemplo do arquivo **l-info3.rpt**

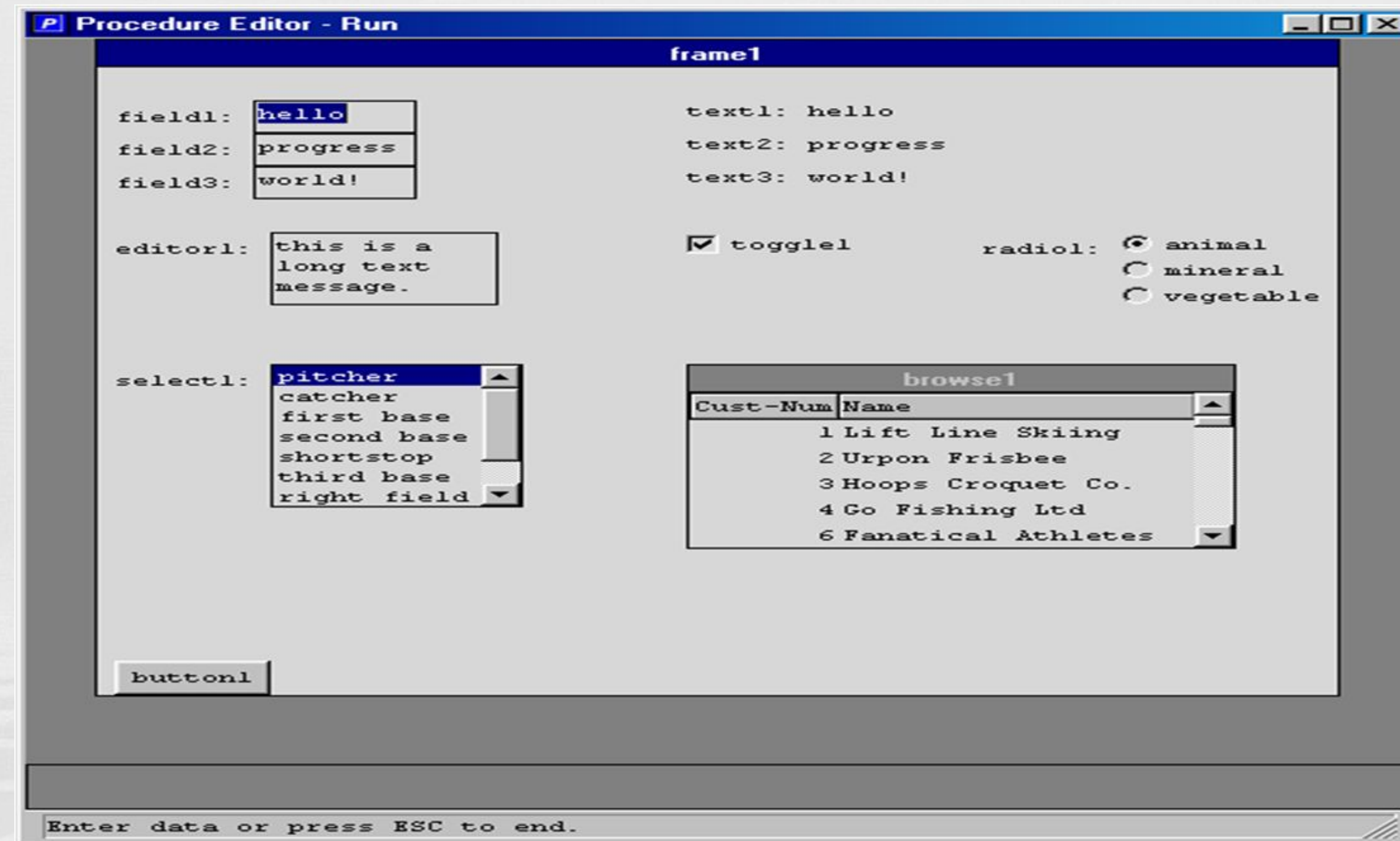
<u>Cust-Num</u>	Name	City	Country
1	Lift Line Skiing	Boston	USA
2	<u>Urpon Frisbee</u>	<u>Valkeala</u>	Finland
3	Hoops Croquet Co.	Hingham	USA
4	Go Fishing Ltd	Harrow	United Kingdom
6	Fanatical Athletes	Aberdeen	United Kingdom
7	Aerobics valine KY	<u>Tikkurila</u>	Finland
8	Game Set Match	Sundsvall	<u>Sverige</u>
9	<u>Pihtiputaan Pyora</u>	<u>Pihtipudas</u>	Finland
10	Just Joggers Limited	<u>Ramsbottom</u>	United Kingdom

Exemplo do arquivo **l-bal3.rpt**

<u>Cust-Num</u>	Name	Sales-Rep	Credit-Limit	Balance
1	Lift Line Skiing	HXM	66.700	42.568,00
2	<u>Urpon Frisbee</u>	DKP	27.600	17.166,00
3	Hoops Croquet Co.	HXM	75.000	66.421,00
4	Go Fishing Ltd	SLS	15.000	689,00
6	Fanatical Athletes	SLS	38.900	37.697,00
7	Aerobics valine KY	DKP	13.500	10.439,00
8	Game Set Match	RDR	15.000	3.373,00
9	<u>Pihtiputaan Pyora</u>	DKP	29.900	25.792,00
10	Just Joggers Limited	SLS	22.000	16.621,00

6. Modifique o programa **l-rpt3.p**, chamando-o de **l-rpt4.p**, de modo que as informações sejam classificadas por representante, e para cada novo representante uma nova página deve ser utilizada.

1. Agora que você já conhece os Widgets. Crie um programa **l-app1.p** conforme abaixo, contendo três FILL-IN, três TEXT, um EDITOR, um TOGGLE-BOX, um RADIO-SET, um SELECTION-LIST, um BROWSE de clientes com o código e nome do cliente e um botão. Ao clicar no botão o programa deve ser encerrado.



The screenshot shows a window titled "Procedure Editor - Run" with a sub-header "frame1". The interface contains the following elements:

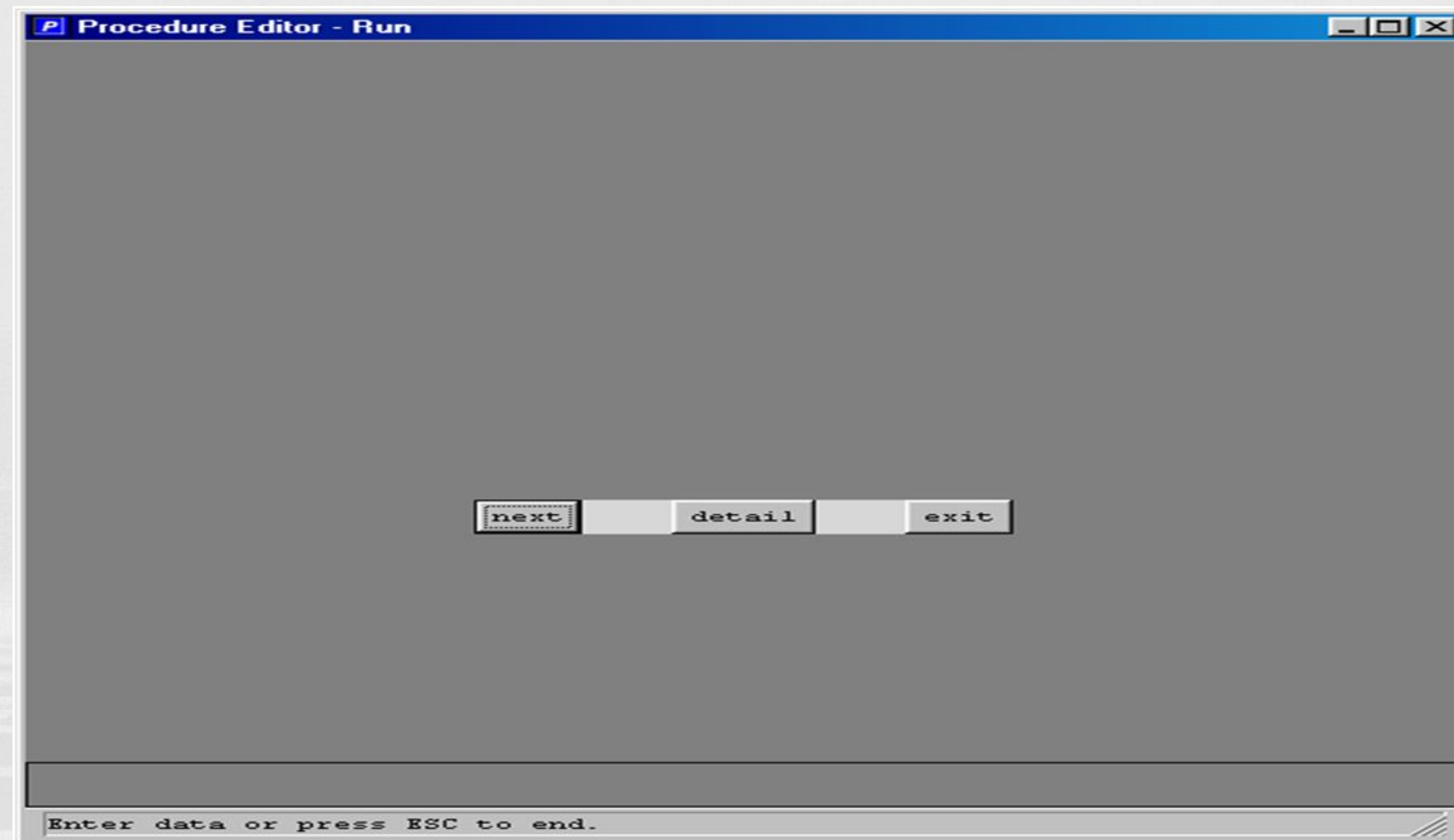
- field1:** A text box containing "hello".
- field2:** A text box containing "progress".
- field3:** A text box containing "world!".
- editor1:** A text area containing "this is a long text message."
- select1:** A selection list with options: "pitcher", "catcher", "first base", "second base", "shortstop", "third base", and "right field". "pitcher" is selected.
- text1:** A text box containing "hello".
- text2:** A text box containing "progress".
- text3:** A text box containing "world!".
- toggel1:** A checked checkbox.
- radiol:** A radio button set with options: "animal" (selected), "mineral", and "vegetable".
- browse1:** A browse window with a table of client data.
- button1:** A button at the bottom left.

The table in **browse1** has the following data:

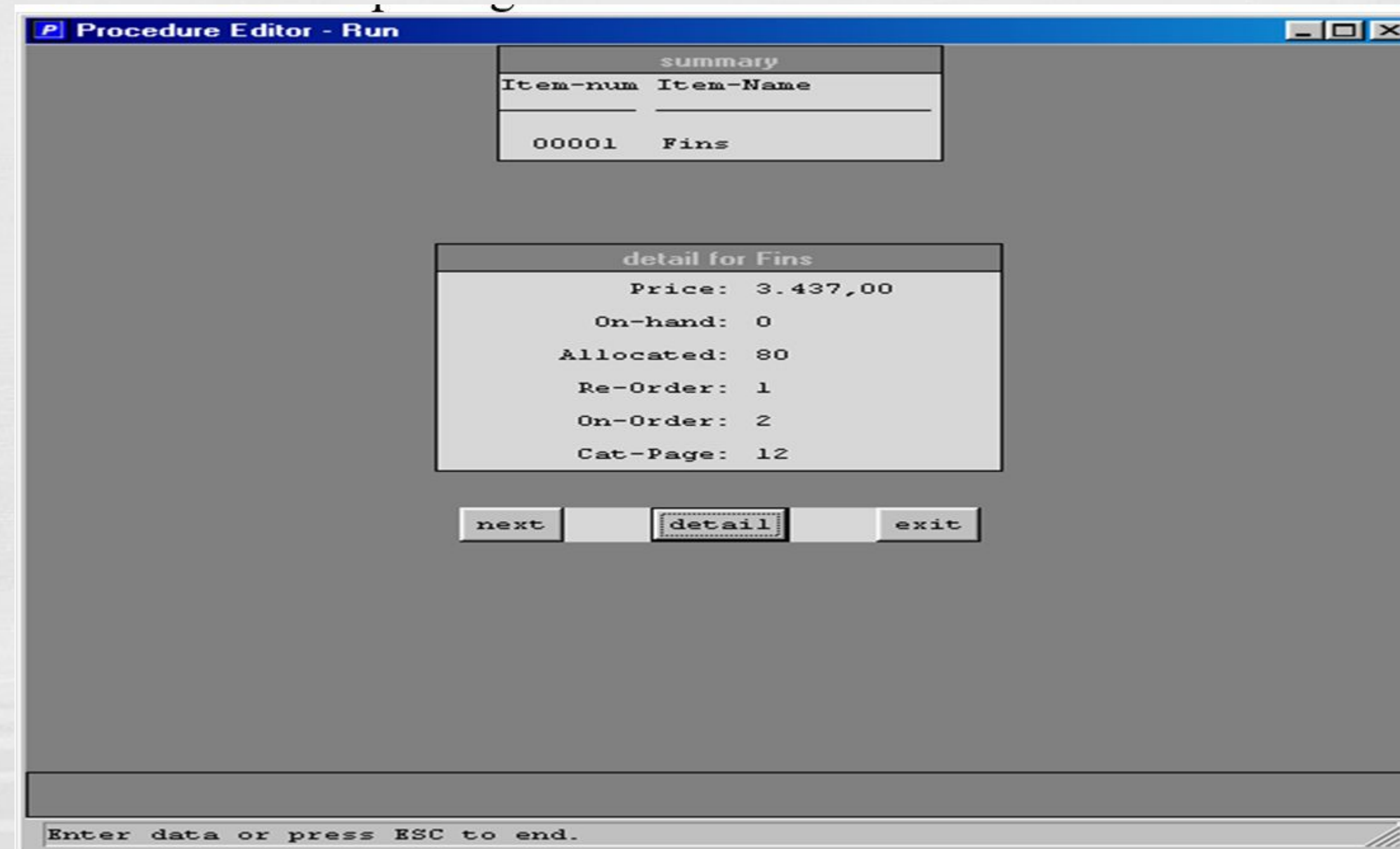
Cust-Num	Name
1	Lift Line Skiing
2	Urpon Frisbee
3	Hoops Croquet Co.
4	Go Fishing Ltd
6	Fanatical Athletes

At the bottom of the window, a status bar reads: "Enter data or press ESC to end."

2. Escreva um programa **l-but1.p** que crie 3 botões que tenha os labels Detail, Next e Exit. Os botões devem ser colocados em uma única frame. Inclua o evento de choose que quando ele ocorrer o label do botão selecionado seja mostrado na linha de mensagens. A seleção do botão Exit deve encerrar o programa. A tela deve assemelhar-se a que segue:



3. Modifique o programa **l-but1.p**, chamando-o de **l-but2.p**. Este programa deve mostrar os campos número e nome do item quando o botão Next for selecionado. Quando o botão Detail for selecionado os campos preço, disponível, alocado, em pedidos e página do catálogo do item deve ser mostrados em outra frame. Inicialmente, mostre o primeiro registro da tabela (apenas o número e o nome do item). A tela deve assemelhar-se a que segue:

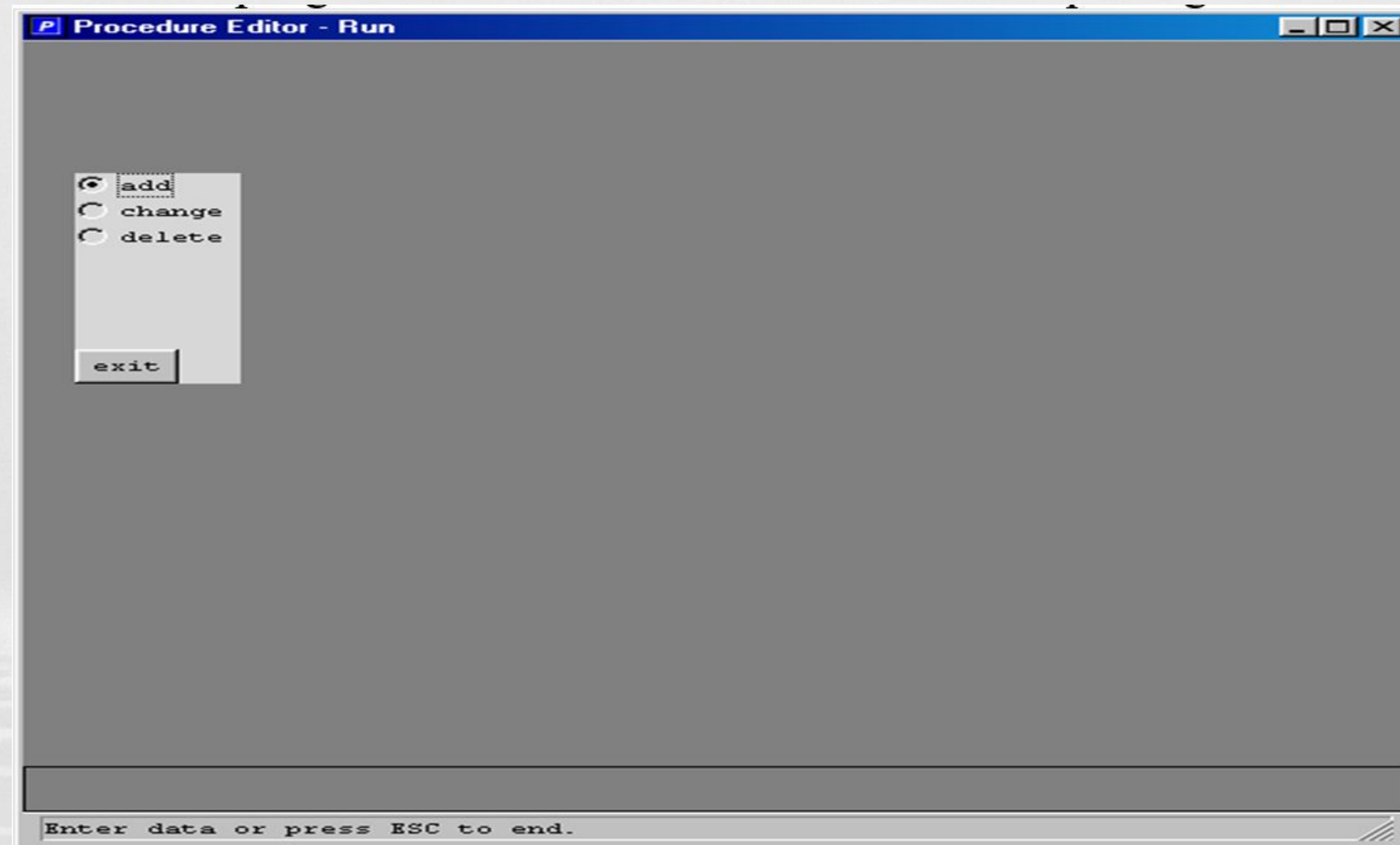


The screenshot shows a window titled "Procedure Editor - Run" with a grey background. At the top, there is a "summary" table with two columns: "Item-num" and "Item-Name". The first row contains the values "00001" and "Fins". Below this, there is a "detail for Fins" section containing the following information:

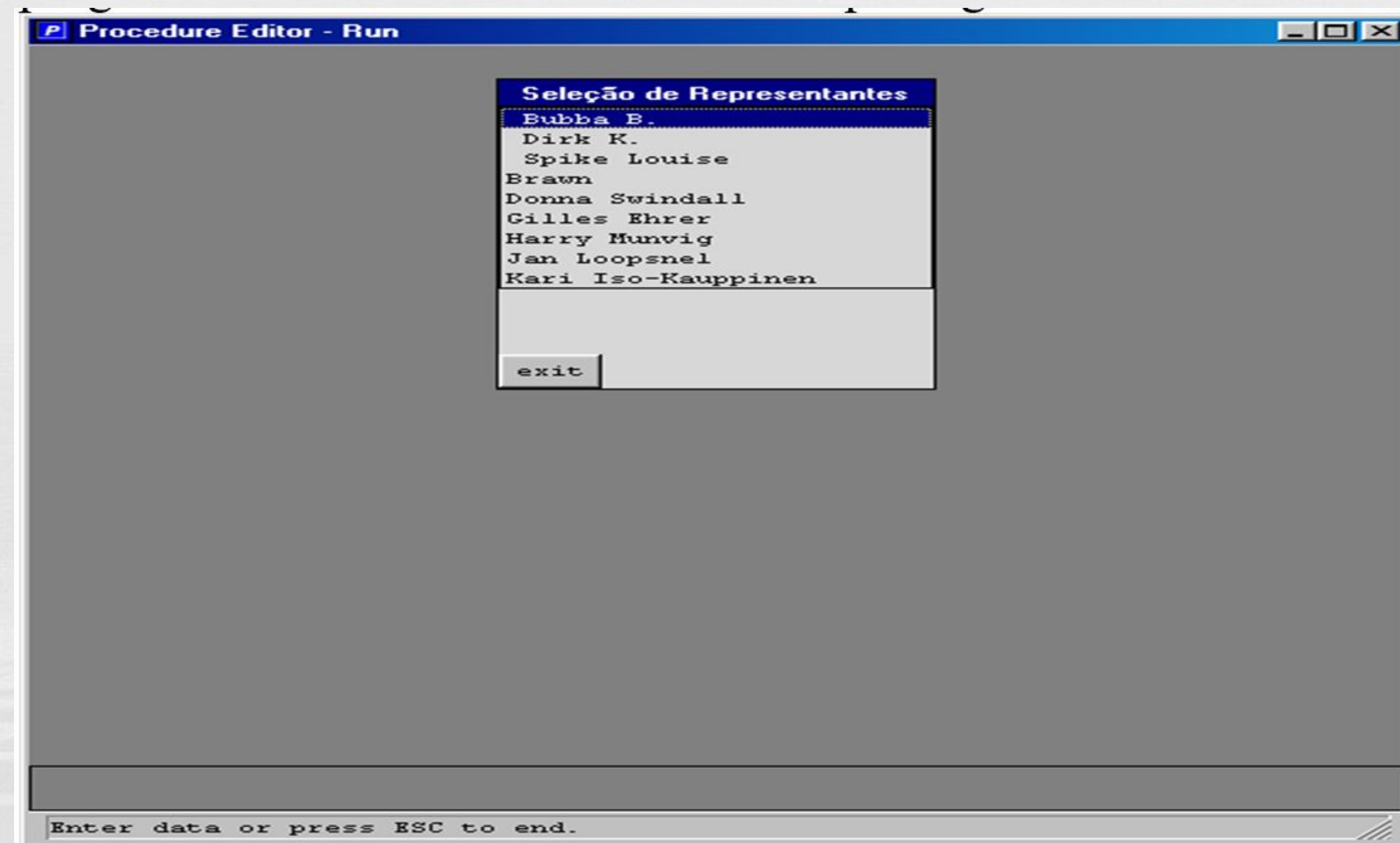
Price:	3.437,00
On-hand:	0
Allocated:	80
Re-Order:	1
On-Order:	2
Cat-Page:	12

At the bottom of the window, there are three buttons: "next", "detail", and "exit". The "detail" button is currently selected, indicated by a dashed border. At the very bottom of the window, there is a status bar that reads "Enter data or press ESC to end."

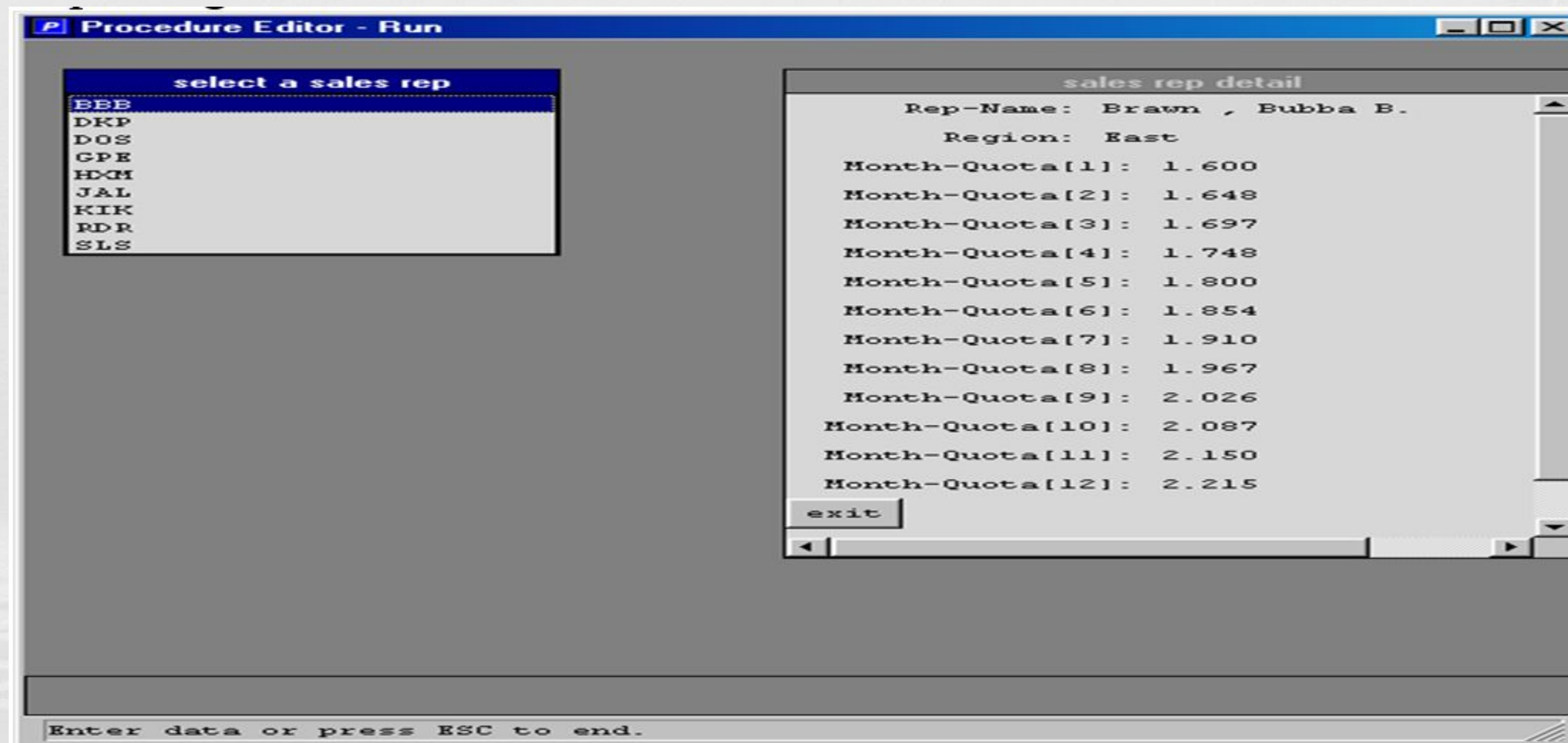
4. Escreva um programa **l-radio1.p** que mostre um RADIO-SET. Os labels do RADIO-SET devem ser Add, Change e Delete. Defina também um botão Exit. Quando o evento VALUE-CHANGED ocorrer no RADIO-SET o seu valor deve ser mostrado. Coloque esta saída num ALTER-BOX. O botão Exit deve encerrar o programa. A tela deve assemelhar-se a que segue:



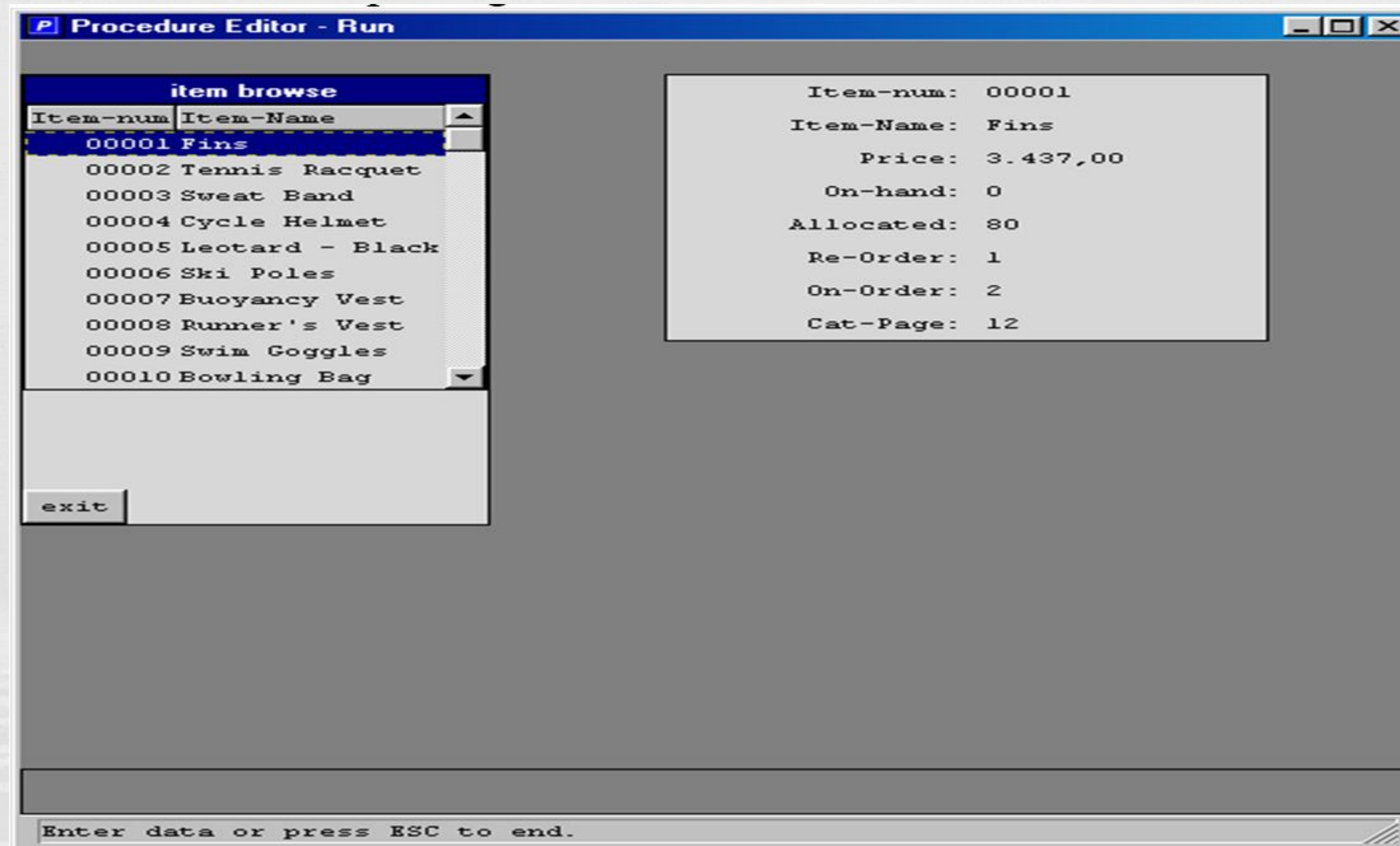
5. Crie um programa **l-select1.p**. Ele deve ter um SELECTION-LIST e um botão Exit. Estes widgets devem estar na mesma frame. Coloque os registros da tabela representante como elementos da SELECTION-LIST. A lista deve ser classificada pelo nome do representante. O botão Exit deve encerrar o programa. A tela deve assemelhar-se a que segue:



6. Modifique o programa **l-select1.p**, chamando-o de **l-select2.p**. Crie duas frames. A primeira frame contém o SELECTION-LIST, e a segunda contém os campos nome, região e cota mensal da tabela representante e um botão Exit. O SELECTION-LIST deve ter como elementos o código de todos os representantes. Toda vez que o evento VALUE-CHANGED ocorrer no SELECTION-LIST, os campos da segunda frame referente o representante selecionado devem ser mostrados. O botão Exit deve encerrar o programa. Inclua o evento TAB de modo que você possa se mover através das frames. A tela assemelhar-se a que segue:

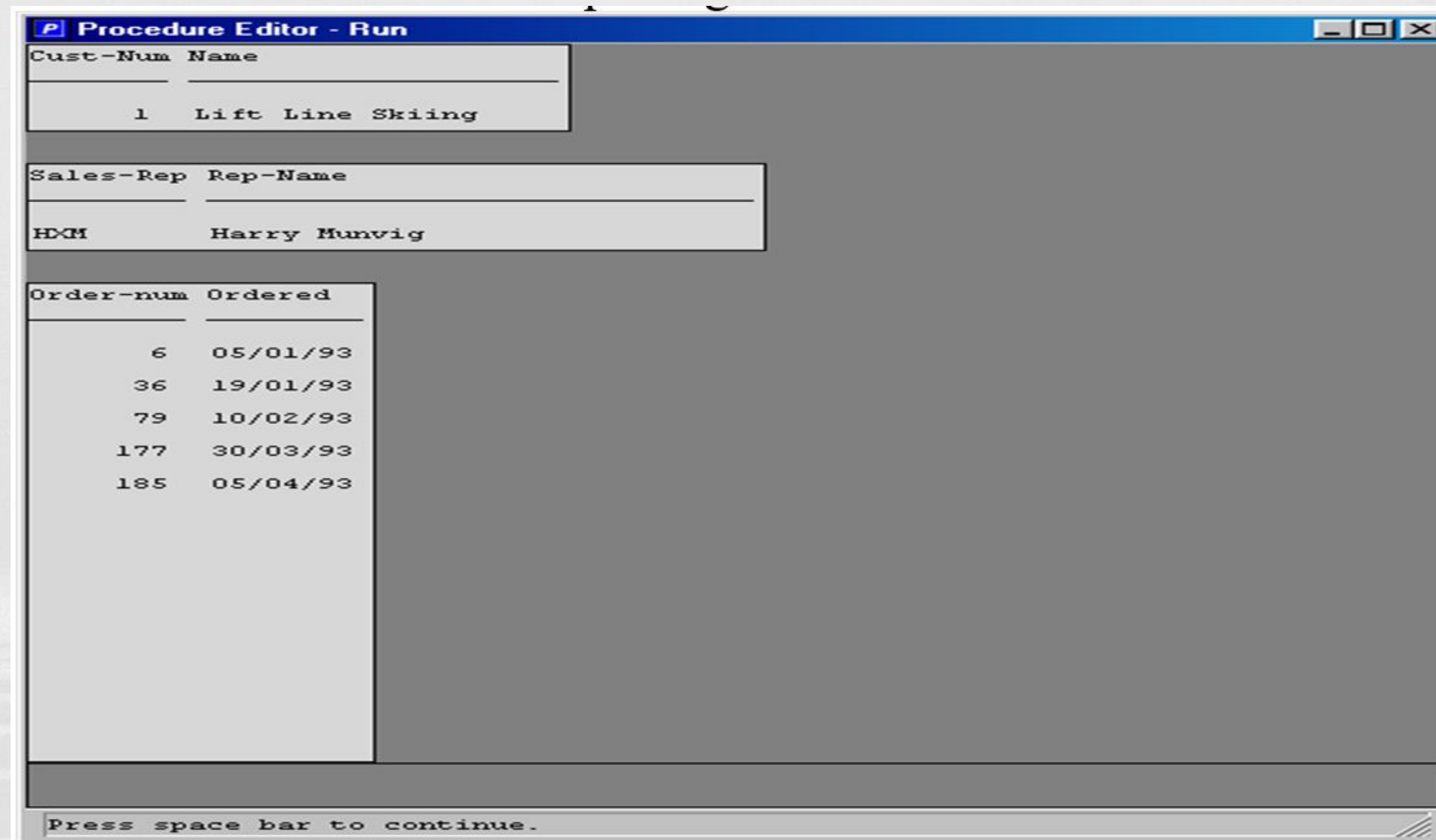


7. Crie o programa **l-browse1.p** usando o widget BROWSE para a tabela item. Ele deve conter os campos número e nome do item. Crie uma segunda frame que contenha todos os campos da tabela item, exceto a descrição de catálogo. Toda vez que o evento ITERATION-CHANGED ocorrer no BROWSE, deve-se mostrar o registro apropriado na segunda frame. O botão Exit deve encerrar o programa. A tela deve assemelhar-se a que segue:



1. Escreva um programa **l-tptb.p** que contenha uma tabela temporária que seja igual a tabela ORDER. Adicionar o atributo nome, este deverá ser igual ao atributo da tabela cliente. O programa deverá ler todos os clientes, todas os pedidos dos mesmos a atualizar a temp-table, alimentando os atributos: número, nome e representante do cliente, data e número do pedido. Uma listagem deve mostrar os atributos acima e todos os registros da temp-table.

1. Escreva um programa **I-proc.p** que leia todos os clientes, o representante do cliente e os pedidos do cliente. Crie duas procedures internas, uma para ler o representante o outra para ler os pedidos. O programa deve mostrar o código e nome do cliente, o código e nome do representante, o número e data do pedido. A tela deve assemelhar-se a que segue:



Cust-Num	Name
1	Lift Line Skiing

Sales-Rep	Rep-Name
HXM	Harry Munvig

Order-num	Ordered
6	05/01/93
36	19/01/93
79	10/02/93
177	30/03/93
185	05/04/93

Press space bar to continue.

2. Escreva um programa **l-comp.p** que compile o programa **l-include.p** e salve o .r no **c:\tmp**.

1. Crie três programas, **l-shar1.p**, **l-shar2.p** e **l-shar3.p**. O primeiro programa deve chamar o segundo e o segundo deve chamar o terceiro. O primeiro deverá conter uma variável do tipo SHARED que deverá ser atualizada com um valor qualquer, o segundo e o terceiro deverão ter esta variável compartilhada, e o terceiro deverá mostrar o valor desta variável.
2. Modifique os programas **l-shar1.p**, **l-shar2.p** e **l-shar3.p**, chamando-os de **l-param1.p**, **l-param2.p**, **l-param3.p** respectivamente. O programa deve ter a mesma funcionalidade, mas não poderá mais utilizar variável SHARED, deverá utilizar Parâmetros.

1. Escreva uma programa **l-1104.p** que crie novos registros na tabela cliente utilizando a importação de um arquivo conforme abaixo:

"160" "Jim Jones" "Baltimore" "20000"

"161" "Jon James" "New York" "22000"

"162" "Sam Snead" "Toronto" "16000"

- 2 – Escreva um programa **l-1105.p** que exporte todos os estados para o arquivo **state.exp**.

1. Escreva um programa **l-1101.p** que permita alterar todos os pedidos e todas as linhas de cada pedido. Garanta que a alteração de um pedido e todas as suas linhas seja uma única transação. Nota: Na alteração da linha do pedido não permitir alterar o número do pedido.
2. Modifique o programa **l-1101.p**, chamando-o de **l-1101a.p**, de modo que chame uma procedure interna para alterar as linhas de um pedido. Esta alteração muda a transação ?
3. Modifique o programa **l-1101a.p**, chamando-o de **l-1102.p**, de modo que a alteração do pedido seja uma transação e a alteração de cada linha do pedido seja outra transação.
4. Modifique o programa **l-1101.p**, chamando-o de **l-1103.p**, de modo que se você pressionar END-ERROR quando estiver alterando o pedido ou linhas do pedido, todas as alterações feitas no pedido e nas suas respectivas linhas já atualizadas seja desfeito e o programa deve ir para o próximo pedido.

Escreva um programa **J-128.p** que gere um JSON com o cliente (campos: CustNum e Name) com os seus respectivos pedidos (campos: OrderNum, OrderDate) e com o total de cada pedido. Nota: Para obter o total do pedido será necessário ler a tabela OrderLine.