```
Original Linked List: 1 -> 2 -> 3 -> 4 -> 5 -> NULL
After deleting the first element: Linked List: 2 -> 3 -> 4 -> 5 -> NULL
After deleting element '3': Linked List: 2 -> 4 -> 5 -> NULL
After deleting the last element: Linked List: 2 -> 4 -> NULL
```

SURYA Gold After deleting the Just element:

After deleting the Just element:

Ainked dist: 2>3>4>5>AULL

After deleting element '3': dinked list: 2>4>5>AULL

After deleting the latest last element: dinked dist: 2>4>5>AULL deleta"

Place & While (phr) data - val) Pheph - phr; Pheph - phr; Prephr - next - phr) next; Telen Alak after head of the fact of the data
elac & while (Ar-data and) physical
Cho & Chile (Ar-data evel) Shipt = phr; Phr - phr - phr; After diliting the After d
Cho & While (ptr-)date evel) Phyph - ptr; ptr - ptr; ptr - ptr; ptr - p
while (Ar-data and) September 1 Pteph - phr; Althor deliting the After deliting the
Phephrophy; Phephrophy; Phephrophy; Prophromat; Prephromat; Prephromat; Prephromat; Prephromat; Shut note * Ar, * prephr; Int val; Printf M entry du value gita which du stade has to be delitté; Prephrophy; Chile (prephrodute) = val) Sprinte (prephrodute) = val)
Proper - personat; Proper - personat; Proper - personat; Free (per); Teture male * delate after Interest and * chart) Should note * per * proper; me val; Print [*M enter the value after which the rode has be delate; Frepty = per; Chile (proper adult) = val) Spropty = per; Chile (proper adult) = val) Spropty = per; Proper
After deliting ele The proper - new - propert; Free (ptr); Tetorn short: 3 3 3 3 5 5 5 5 6 5 6 6 6 7 7 7 8 7 8 7 8 8 8 8 8 8
Prepty > next = ptr>next; free (ptr); 7 cturn Mart; 3 3 3 3 3 3 3 3 3 3 3 3 3
Prepty = nent = ptr-scent; free (ptr); Yether Next; 3 3; Great and * ptr, * prepty; Int val; Print [1] M entry du value giter which the roade has to be dibble; gty = start; Prepty = ptr; Ohile [prepty-state] = val) 5 Prepty = ptr; ptr = ptr = ptr; ptr = ptr; ptr = ptr = ptr = ptr; ptr = ptr = ptr = ptr; ptr = pt
Teturn Most; 3 3) Struct node * Allet after [Ahnot node * Chart) Struct node * Ar, * prepty; Int val; Print [M] enter the value often which the rode has to be abbted; Scant ("Id", a val); Pty = start; Prepty = At ; While [prepty-adate] = val) Sprepty = At; Aty > potent point; 3
Struct node * delete after Interest node * Struct Struct node * Ar, * Oreptr; Int val; Printf [M] enter the value after which the node has to be abble of the performance of the perf
Struct and * delate of the Short and * Stort) Shout node * ptr, * preptr; Int val; Printf [*\M enter du value often which du node has to be delaté; Scant (* 1d", a val); ptr = start; Preptr = ptr; While [preptr->data] = val) E Preptr = ptr; ptr > ptr > ptr; ptr > ptr > ptr; preptr > ptr; prep
Struct node * Ar, * prepty; The val; Print [] M enter the value often which the node has to be dibted; Scant ("Yd", aval); Ptepty = pty; While [prepty - pty; Prepty = pty; Prepty
Struct node * ptr, * preptr; Int val; Printf ["M enter the value often which the node has to be debted; Scant ("Id", a val); ptr = start; Preptr = ptr; Chile [preptr-adata] = val) Expreptr = ptr; Preptr = ptr
Struct node * ptr, * preptr; Int val; Printf ["M enter the value often which the node has to be debted; Scant ("Id", a val); ptr = start; Preptr = ptr; Chile [preptr-adata] = val) Expreptr = ptr; Preptr = ptr
Struct node * ptr, * preptr; Int val; Printf ["M enter the value often which the node has to be debted; Scant ("Id", a val); ptr = start; Preptr = ptr; Chile [preptr-adata] = val) Expreptr = ptr; Preptr = ptr
Print ["M enter the value often which the node has to be debted? Scant ["1d", sval]; ptr = start; Preptr = ptr; While [preptr->data] = val) Expreptr = ptr; Preptr = ptr;
Print (M) enter the value often which the mode has to be debted? Scant ("Id", sval); Pty = start; Prepty = pty; While (prepty-sdata) = val) Sprepty = pty; Pty > ptot > prepty; 3
Ptr = stant; Preptr = ptr; While (preptr->data =val) Spreptr = ptr; Ptr > ptor > punt; 3
Ptr = start; Preptr = ptr; While (preptr->data =val) Spreptr = ptr; Ptr > ptor > purt; 3
Preparation of the policy of t
Chile (Prepty-)data ! = val) § Prepty = Pty; Pty > pty > purt; 3
$\frac{3}{3}$ $\frac{1}{3}$ $\frac{1}{3}$ $\frac{1}{3}$ $\frac{1}{3}$
3
3
3
Die ot 8 > rest = otx > rest:
free (ptr);
return starti
3;

YA Gold			
Page	-	DATE OF THE PARTY	
		[deletion]	
	(20)	Struct node * dulek beg (struct node * Start)	
		& duck - beg (struct node * Start)	
		Stand and A . >	The same
		Struct node * pt x 3; Pt x = Start;	E TOTAL
		Stal = 01	
		Start = Star = nent; free (ptx);	
		Yelen all In	
		3;	
			THE STREET
		Street and # 110	
		Street node * delete-end (struct node * Start)	
			The same of
		Struct mode * Ptr, + ptupty;	
		Pfr= Start;	DE THE
_		while (pts->next!=nbll)	
		Prepty=Pty;	
		Ptr= ptr >nent;	- C.
		3	
to be murks:")		Pretpto->nent=NUL;	
		free (Pts);	
nede);		Veturn Start;	
		? ;	
	6	Fruit node * delete node (struct node * Start)	
		5	
		Sterner most tota, *prepty;	
		int Val:	
		print (!d", sval);	
		Ptr = start;	
		if Irts > data = = val)	
		2	
		Stood = delete - beg (stood); return = stood;	
		3	

	SURYAGOID Date Page New-node > data= num; Ptr = start; While (Ptr->data)=val) E Ptr=Ptr->newt; 3.	(00)	Struct noc Struct Struct Pty = 1 Start : Free (P
,	play epty-ment = new-nale;		7 durn
-	New_node => new t = Pty;		3;
	Peturn Short:		10
	3,		Street node
			Struct
	Struct node + insort often (struct node + steet)		Pfr= S
			While
	Struct node + new node, + ptr, + prefit;		3
	Provides		Pre
	Printf (" \a enter the date:");		Ptr
	Scant (21) d ", snum);		3
	Printf ("In order the Value after which data has to be mores"). Scant ("I'd", s val);		Pretpt
	Start (rd , I val);		tree (Pt
	New noch = (struct node +) malluc (size of (struct mode));		Tetwa S
	Pty-starti		3;
	Prepty=pty;		(1)
	While (Atre ptr-sdate != val)		Street node
	E /		Stephen
	Preptraptri		int Val
	ptr-ptr-next;		prints (
	2		Ptr = sd
	Ptepty-) next = new_node;		if late.
	new node -> next - ptri		g.
	return starti		Star
	3;		3
			NO. OF THE PARTY O

A STATE OF THE STA		SURYAGold	
		DataPage	- HUMBL
	While Ptr-> next = NUL	L	
	Ptr=Ptr>nent;		Struct no
	Ptr=next=new_node;		5
1	New-node -> next = NULL	, , ,	Startet
_	3		int no
-	Print ("In enter the data:	");	Printf
	Scant (" I'd", soum);		Scant (
			New-100
- return Ata	x);		New p
- 3,	(0)	and hand	new_n Pty=E
-		The second second	if late
- Starret node	* display Shrut node * Sta	ert)	£
2.			5
- Strut r	lock of pby;	Manual Harman	3
Ptr = Sto			- dre
while (pt	x != Null)	beautiff I	٤
-	· /\\\ \		While
Print	f ('At 1d", ptradata);	Manager 1	
3	Ptr->nent;		Pho
3;			Telun
			3
Whent onto	* insert-beg[struct node * 6		3;
3	ment begistact node * 6	Hart)	5-1-01
Street node	* 000 mt :		Street node
Int num;			Street noo
Print ("In ent	a de data; 1);		ins num, y
Scant ("/d",1	num);		print(1)
New -node = (St	rent node*) malloc (sign of show	(d (m. 1)):	Scant (
Non-Josh - Ma	ati=num;	Marin,	print (")
New_node -> 1	unit = start;		Stant (")
start = new =1			New-nate
Petur start)			
\$;			ALTERNATION OF THE PARTY OF THE

		C01
SURYA Gold Data Page		
f a node at Jesse	SURYAGOLA	
Jess .	Case 2: Start = insert beg (start);	The Later of the L
	DIGK.	
	Case 3: Start = insert end(start);	
	DYGA:	
	Case 4: Start = insert before (start);	
	B.T. L. I	
	Care 5: Start = insert_after (start)	
	3 break;	100
	36hile (Choice 1 = 6);	THE STREET
	return 0;	
e * Ji	3	
(*); (*).		100
ole*);	Struct node + Cruste 11 (Struct node + stort)	
ode*);	8	
rd #):	Struct node * new_node, * ptr;	
	print f (" \n eater -1 to cod ");	- FEET 188 3
<i>(</i>);	print (" In enter the data: ");	
	Scant (" td", Anom);	
	hhile (num 1 = -1)	100000
list \t2. display \t3. involy	£	The same
t-ofter);	New-node = (struct poste +) malloc (size of (struct poste));	CA STANDARD
	New-nost > det = num;	
	if (start == NULL)	AND THE RESERVE
	New-node > nend = NULL;	
	Start = new_node;	THE CAN
	3	THE WAY THE
	else.	
(Seated)	£ 01 01 1.	Carrie Const
	Pty= Start;	
TAKE I		THE REAL PROPERTY.

STREET, STREET	SURYAGold	
	Singly Linked List Date Page	
Cruste	a linked list and insertion of a node at Jesse	C 0. (1)
Position	J	Case 2: Start=
	The second secon	Case 3: Start=
# include <		by wh;
# Include <		Case 4: Start =
Struct- no	2de	break;
<u> </u>	·	Care 5: Start=
	data;	break,
Stru	et node * nert;	3
2;	- L V Cl	3 While (Chaice) = 6
11 0	node * Start = Null; node * Create 11 (Strut notte *);	return a;
Strus 1		ξ.
Struct 1		Strut node + Ca
	node * insert-end (Street nole*);	E
	rode * insert-before (struct node*);	Struct pode * n
	nool * insert-after (should nool *):	int num;
	- The state of the	print f (" In enter
5trust 1	nade * Sort_list (street nade *);	printf (" \n ent
int main		Scant ("Id", An
int Choice		hihik (num 1=-1)
Print+(')	n * * meru * * \ M. Create a lint \t2 . display \t3. ites	New-node = (st
do	next-end) to insert before to insert ofter);	New-next >
<u> </u>		if (stint ==
Drint	f ("In enter the Choice:");	
	(" 1d, 1 cha (e);	New-node
	(Choice)	Start = 1
٤		else
Case	1: Start = (rust _11 (start);	E E
	Print (" In linked list Guard")	Pty=
	break;	