

NODE delete-front (NODE first) printf ("List is empty. Cannot delete")
return first; temp = temp -> link;

printf ("Item deleted at front end is if d \n",

first -> info); (NODE second, int item) temp = getnodel); temp > info = item; temp > link = NVLL; if (Second = = NULL) seturn temp;

temp -> link = swend; second = temp; neturn second; NODE IR (NODE second, int item) { NODE temp, cur; temp = jutnode(1); temp > link = null; seturn = Null; return temp; sum = slownd; while (um -> link! = Null) seturn stemp; return accord; NODE sum, temp; return scond; NODE sum, temp; while (first! = NULL) temp = first; first = first -> link; temp -> link = sum; sum = temp; return sum; NODE saccording (NODE first) { NODE saccording (NODE first) }	
NODE IR (NODE slowed; int item) \[\text{NODE temp, cur;} \\ \text{temp = getnode(1);} \\ \text{temp = sinft = item;} \\ \text{temp > sink = NULL;} \\ \text{if (acond = NULL)} \\ \text{veturm temp;} \\ \text{cur = suond;} \\ \text{while (um > link! = NULL)} \\ \text{veturm > link = temp;} \\ \text{veturm alcond;} \\ \text{NODE memmal (NODE fixst)} \\ \text{NODE and, temp;} \\ \text{nun = NULL;} \\ \text{maile (first! = NULL)} \\ \text{temp > link = cur;} \\ \text{veturm and = temp;} \\ \text{veturm sun;} \\ veturm	1 0 1 - 100 and i
NODE IR (NODE slavnel, int item) { NODE temp, curr; tump = gatnode(); temp = infor = item; temp > link = NULL; if (surend = NULL)	tup -> hm - source
NODE IR (NODE second, int item) { No DE temp; cur; temp = getnode(); temp > info = item; temp > info = item; temp > hink = NULL; if (second == NULL) xeturn temp; xem = second; while (um > link! = NULL) xem = cur > link; xum = cur > link; xum = link; xum = null; while (first! = NULL) { temp = first; first = first > link; temp -> link = cur; cur = temp; } yeturn cur; } yeturn cur;	second = temp
{ No DE temp; CM, temp = getnode(); temp = getnode(); temp = hok = NULL; if (second == NULL)	Mttum shang
{ No DE temp; CM, temp = getnode(); temp = getnode(); temp = hock = NULL; if (second == NULL)	a : t items
{ No DE temp; CM, temp = getnode(); temp = getnode(); temp = hock = NULL; if (second == NULL)	NODE IR (NODE second, Int 1700)
temp = getnode(1; temp = hings = 1 tem; temp -> hink = NULL; if (second == NULL) return temp; sum = shownd; while (sum > hink != NULL) seturn shownd; NODE memmal (NODE first) NODE sum, temp; sum = NULL; while (first! = NULL) temp = first; first = first -> link; temp -> link = sum; sum = temp; rum = temp; rum = temp;	(AAM
return temp; return temp; return temp; return temp; return = second; return > link ! = NULL) return > link = temp; return alcoud; NODE remal (NODE first) NODE am, temp; return = NULL; while (first! = NULL) temp = first > link; temp -> link = rem; return -> link = rem; return -> link = rem;	temp = getnode();
return temp; return temp; return temp; rem = shound; while (um > link! = NULL) rem = remp; return shound; NODE remal (NODE first) NODE am, temp; rem = NULL; while (first! = NULL) temp = first > link; temp -> link = rem; rem = temp; return rem; return rem;	temp sinfo = 1 tem;
return temp; return temp; return temp; rem = shound; while (um > link! = NULL) rem = remp; return shound; NODE remal (NODE first) NODE am, temp; rem = NULL; while (first! = NULL) temp = first > link; temp -> link = rem; rem = temp; return rem; return rem;	temp -> link = NULL;
return temp; Ann = slashed; While (um > link! = NULL) Aum = am > link; Aum = link = temp; Selam record; NODE newrol (NODE fixst) [NODE aum, femp; Aum = NULL; while (first! = NULL) { temp = first; first = first > link; temp -> link = cum; Aum = temp; } yeturn cum;	if (second = = NULL)
AM = sland; While (um > link! = NULL) Am = am > link; Am = hink = temp; Yelum record; NODE am, temp; Am = NULL; While (first! = NULL) { temp = first; first = first > link; temp -> link = am; Am = temp; Yelum am;	return temp;
while (um > link ! = NULL) rum = rum > link; rum -> link = temp; velum record; NODE remal (NODE first) NODE rum; femp; rum = NULL; while (first! = NULL) temp = first; temp -> link = rum; rum = temp; velum rum;	and - alaskad
Aum = mm > honk; Aum = link = temp; Seturn alcond; NODE memoral (NODE first) NODE cum, temp; Aum = NULL; while (first! = NULL) temp = first -> link; temp -> link = cum; Aum = temp; seturn cum;	1 do 10 (com > link = NULL)
NODE memeral (NODE fixst) NODE and, temp; NODE and, temp; Mule (first! = NULL) Temp = first ; first = first -> link; temp -> link = com; com = temp; Yelum and;	CAM = rum -> link;
NODE memeral (NODE first) [NODE awn, femp; awn = NULL; while (first! = NULL) { temp = first; first = first > link; temp -> link = awn; awn = temp; } return aun;	Pink - temp
NODE memeral (NODE first) NODE care, femp; aum = NULL; while (first! = NULL) temp = first; first = first > link; temp -> link = care; care = temp; return care;	~h reared
NODE am, temp; am = NULL; while (first! = NULL) temp = first; first = first -> link; temp -> link = cur; am = temp; Yeturn cur;	1 (NWW) NXXXVIII
NODE am, temp; am = NULL; while (first! = NULL) temp = first; first = first -> link; temp -> link = cur; am = temp; Yeturn cur;	June Lung (Syst)
while (first! = NULL) temp = first; first = first > link; temp -> link = run; run = temp; return run;	NODE MINIMAL (NODE 1851)
while (first! = NULL) { temp = first; first = first > link; temp -> link = cm; cm = temp; } *etum cm;	A. A
while (first! = NULL) temp = first; first = first > link; temp -> link = cm; cm = temp; return cm;	NODE am, Tempi
temp = finst; finst = finst > link; temp -> link = cun; cun = temp; yetum cun;	MM = NULL;
first = first -> link; temp -> link = cun; cun = temp; } yetum cun;	while (first! = NULL)
first = first -> link; temp -> link = cun; cun = temp; } yetum cun;	
return cun;	temp= finst;
return cun;	first = first -> link;
return cun;	temp - link = cm;
return cun;	cun = temp;
	}
	setum cum:
NODE ascending (NODE first)	
	NOTE morranding / NOTE limit)
	C (1001)

```
rint temp;
if (first == NULL)
che
```

	else
	9 1 1 - 011124
	& while (mer! = NULL)
	musile (com! = NULL)
	MI = NULL)
100	if [prev > info < sum > info)
	temp = prev -> rispo; prev -> rispo = curl -> rispo; curr -> rispo = temp;
	mer -> info = cum -> x vgo)
	m -> info = Temp,
	em = cm -> link;
	prev = prev > link;
	A A A A A A A A A A A A A A A A A A A
	} \
	- return first;
	}
	NODE concatenate (NODE furst, NODE second)
	> NODE com;
	if (finst = = NULL) return second; if (second == NULL) return finst;
	Metroma second
	it / 2000000 == 21111 1)
	Me francisco Cinat
	1 www fired;
	while (m -> link!= NULL)
	While (mm -> link != NVLL)
	com = com -> link;
1	
	Jam > link = second;
	XXI XXIV XXIV
1	

neturn first; display (NODE first List is empty. Comnot display"); List contents are: "); p=first; temp! = NULL; temp= the choice: ") case 1: printf ("Futer item at front-end:

seant ("-1.d", & item);
hirst went front (first item);
fixst = insent = front (first, item); print f ("/- d insented at front end", first > int
mak:
case 2: first = delete - front (first);
break;
case 3: first = veneroe (first);
case 3: first = reverse (first); montf ("List is reversed");
bulate;
Case 4: printf ("Press, for Ascending-sort and 2
for Descending -sout"); scanf("/d", & option);
scanflyd 11, & sption);
if (aption = =1)
first = ascending (first); mintf ("List is sented in ascending ording);
printf ("List is sented in according ording).
$\inf \left(\operatorname{sphon} = 2 \right)$
first = descending (first);
printf (" List is sorted in descending order")
A STATE OF THE STA
break;
Cases: mint (" Create a second list \n");
mint f ("Enter the number of elements
mint f ("Enter the number of elements in the second list.")
scant (') d', a num);
fox (int i= 1; i <= num; i++)
mint ("Press 1 to Iwent front and
and 2 to hourt - year: ");
scant ("/.d", & choice 2);
it (choice 2 = =1)

```
printf ("Enter the item at front end: ");
  scanf ("j.d", hitemi);
second = IR(second, itemi);
```