

Deletion

classmate

Date _____
Page _____

```
#include <stdlib.h>
#include <stdio.h>
```

```
void create();
void display();
void delete_begin();
void delete_end();
void delete_pos();
```

struct node

{

```
int info;
struct node *next;
```

};

```
struct node *start = NULL
int main()
```

```
{ int choice;
while(1){}
```

```
printf("1. Create\n");
printf("2. Display\n");
printf("3. Delete from beginning\n");
printf("4. Delete from the end\n");
printf("5. Delete from specified position\n");
printf("6. Exit\n");
printf("Enter your choice: ");
scanf("%d", &choice);
switch(choice)
```

}

case 1:

create 0;
break;

case 2: display 0;
break;

case 3: delete - begin 0;
break;

case 4:

delete - end 0;
break;

case 5: delete - pos 0;
break;

case 6 :

exit(0);
break;

default

printf ("n wrong choice:n");
break;

}

}

return 0;
} void create 0
{

```
struct node * temp, * pti;
temp = (struct node *) malloc (size of (struct node));
if (temp == NULL)
```

{

```
printf ("\\n Out of memory space:\\n");
exit (0);
```

3

```
printf ("\\n Enter the date value for the node: |t");
scanf ("%d", &temp->info);
temp->next = NULL;
if (start == NULL)
```

{

```
start = temp;
```

3

```
else
```

{

```
temp->next = start;
```

```
start = temp;
```

3

{

```
void display ()
```

{

```
struct node * pti;
if (start == NULL)
```

{

```
printf ("\\n list is empty:\\n");
return;
```

{

else

{

```
ptr = start;  
printf ("The list elements are : \n");  
while (ptr != NULL)
```

{

```
printf ("%d \t" ptr->info);  
ptr = ptr->next
```

{

{

{

void delete - begin ()

{

struct node * ptr

if (ptr == NULL)

{

```
printf ("List is Empty: \n");  
return;
```

{

else

{

ptr = start

while (ptr->next != NULL)

{

temp = ptr
ptr = ptr → next

{

temp → next = NULL

printf ("In The deleted element is: %d | t", ptr->info)
free (ptr);

{

{

void delete - pos ()

{

int i, pos;
struct node * temp, * ptr;
if (start == NULL)

{

printf ("In The list is Empty: \n");
return;

{

else

{

printf ("In Enter the position of the node to be
deleted: | t ");

scanf ("%d", &pos);

if (pos == 0)

{

ptr = start

start = start → next;

printf ("In The deleted element is: %d | t", ptr->info);
free (ptr);

3
else

{

ptr = start

for (i = 0; i < pos; i++) { temp = ptr; ptr = ptr->next;

if (ptr == NULL)

} printf ("\n Position not found: \n");
return;

3

3

temp->next = ptr->next;

printf ("\n The deleted element is: %d",

ptr = ptr->next;

if (ptr == NULL)

}

printf ("\n Position not found: \n");
return;

3

3

temp->next = ptr->next;

printf ("\n The deleted element is: %d",

free(ptr);

3

3

3