

#include < std.lib.h >  
#include < stdio.h >

void create();  
void display();  
void insert\_begin();  
void insert\_end();  
void insert\_pos();  
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("\n\n\n 1. Create \n");

printf("\n 2. Display \n");

printf("\n 3. Insert at the beginning \n");

printf("\n 4. Insert at the end \n");

printf("\n 5. Insert at specified position \n");

printf("\n 6. Exit \n");

printf(" Enter your choice : |t ");

scanf("%d", &choice);

switch(choice)

{

case 1:

create();

break;

case 2:

display();

break;

case 3:

insert\_begin();

break;

case 4:

insert\_end();

break;

case 5:

insert\_pos();

break;

case 6:

exit(0);

~~break;~~

default

printf("n Wrong choice:n");

{

{

return 0;

{

void create()

{

struct node \*temp, \*p1;

temp = (struct node \*) malloc(sizeof(struct node));

if (temp == NULL)

{

printf("n Out of memory space:n");

exit(0);

```
3   printf ("\n Enter the data 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 *ptr;
      if (start == NULL)
```

```
{
```

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

```
E
```

```
print else
```

```
L
```

```
ptr = start
```

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

```
{  
    printf("%d | t", ptx->info);  
    ptx = ptx->next;  
}
```

{

{

void insert\_begin()

{

struct node \* temp;

temp = (struct node \*) malloc (size of (struct node))  
if (temp == NULL)

{

printf ("| n Out of memory space: | n");  
return;

{

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

{

start = temp

3  
elseE  
temp->next = start  
start = temp

{

void insert\_end()

{

```
struct node *temp, *ptr;
temp = (struct node *) malloc (sizeof (struct node));
if (temp == NULL)
```

{

```
start = temp;
```

{

```
else
```

{

```
ptr = start;
```

```
while (ptr->next != NULL)
```

{

```
ptr = ptr->next;
```

{

```
ptr->next = temp
```

{

{

void insert\_pos()

{

```
struct node *ptr, *temp;
```

```
int c_pos;
```

```
temp = (struct node *) malloc (sizeof (struct node));
if (temp == NULL)
```

{

```
printf ("|n Enter the position for the new node
to be inserted: |t");
```

~~scanf~~

```
scanf ("%d", &pos);
printf ("\nEnter the data value of %d to the node:");
scanf ("%d", &temp->info);
```

```
temp->next = NULL;
if (pos == 0)
```

{

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

3

else

{

```
for (i=0, ptr = start; i<pos-1; i++) {ptr = ptr->next;
if (ptr == NULL)
```

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

3

3

```
temp->next = ptr->next;
ptr->next = temp;
```

3

3