18/11/2020

Singly Linked program

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
struct node
{
int info;
struct node *link;
};
typedef struct node *NODE;
NODE getnode();
{
NODE x;
x=(NODE)malloc(sizeof(struct node));
if(x==NULL)
{
printf("mem full\n");
exit(0);
}
return x;
}
void freenode(NODE x)
{
free(x);
}
NODE insert_front(NODE first, int item)
{
NODE temp;
temp=getnode();
temp->info=item;
temp->link=NULL;
if(first==NULL)
{
```

```
printf("list is empty cannot delete\n");
return first;
}
temp = first;
temp = temp->link;
printf("item deleted at front-end is = %d\n", first->info);
free(first);
return temp;
}
NODE insert_rear(NODE first, int item)
{
NODE temp, cur;
temp = getnode();
temp->info = item;
temp->link = NULL;
if(first == NULL)
return temp;
cur = first;
while(cur->link != NULL)
cur->link = temp;
return first;
}
NODE delete_rear(NODE first)
{
NODE cur, prev;
if(first == NULL)
{
printf("list is empty cannot delete\n");
return first;
}
if(first->link == NULL)
{
printf("item deleted is %d\n", first->info);
free(first);
return NULL;
}
```

```
prev = cur;
cur = cur->link;
}
printf("Item deleted at rear-end is %d", cur->info);
free(cur);
prev->link = NULL;
return first;
}
void display(NODE first)
{
NODE first = NULL;
for(;;)
{
printf("\n1: Insert-front\n2: Delete-front\n3: Insert-rear
\n4: Delete-rear\n5: Display-list\n6: Exit\n");
printf("ENTER THE CHOICE\n");
scanf("%d", &choice);
switch(choice)
{
case 1: printf("Enter the item at front-end\n");
scanf("%d", &item);
first = insert-front(first, item);
break;
case 2: first = delete-front(first);
break;
case 3: printf("Enter the item at rear-end\n");
scanf("%d", &item);
first = insert-rear(first, item);
break;
case 4: first = delete-rear(first);
break;
case 5: display(first);
break;
default: exit(0);
break;
}
}
```