29/01/2023

1. Single linked digit - sort, reverse, concatination.

Concatenation

```
void concatenation (struct node *a, struct node *b)
{
  if (a != NULL && b != NULL)
  {
    if (a->next == NULL)
      a->next = b.
    else
    {
      printf ("Either a or b is NULL \n");
    }
```

```
struct node *concat (struct node *start1, struct node
*start2)
{
    struct node *ptr;
    if (start1 = start2 == NULL) { start1 = start2;
    return start1;
    }
    if (start2 = NULL)
        return start1;
    ptr = start1;
```

```
ptr = start1;
while (ptr -> link != NULL)
        ptr = ptr -> link;
        ptr -> link = start2;
        return start1;
    }
```

Reversing.

```
void Reverse (struct node *head)
{
    struct node *ptr = NULL, *pre = NULL;
    while (head != NULL)
    {
        ptr = head -> next;
        head -> next = prev;
        prev = head;
        head = ptr;
    }
head = prev;
}
display (head);
}
```

Sorting

```
void sort (struct node *head)
{
    struct node *ptr1, *ptr2;
    int temp;
    ptr1 = head;
    while (ptr -> next != NULL)
    {
        ptr2 = ptr -> next;
        while (ptr1 != NULL) {
        if (ptr1 -> data > ptr2 -> data)
        temp = ptr -> data;
        ptr1 -> data = ptr2 -> data;
        ptr2 -> data = temp;
        }.
        ptr2 = ptr2 -> next;
    }.
ptr1 = ptr -> nut
}.
display (head);
}.
```

Output: 1: Insert 1 2. Insert 2 3. display 1 4. display 2.
5: Concat 6. Sort 7. reverse -1. exit.

Enter your choice: 1

1 2 3

Enter your choice: 3.

Elements of linked list: 1 -> 2 -> 3 ->

Enter your choice: 2.

4 5 6

Enter your choice: 4

Elements of the linked list: 4 -> 5 -> 6 ->

Enter your choice: 5

Element of linked list: 1 -> 2 -> 3 -> 4 -> 5 -> 6 ->

Enter your choice: 6

Elements of linked list: 1 -> 2 -> 3 -> 4 -> 5 -> 6 ->

Enter your choice: 7.

Elements of linked list: 6 -> 5 -> 4 -> 3 -> 2 -> 1 ->

Enter your choice: -1

Q. Stack implementation using single linked list.

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

void push();
void pop();
void display();

struct node
{
  int val;
  struct node *~~pt~~next;
};

struct node *head;

void main()
{
 int choice = 0;

  --11--
  --11--
  --1--
}
```

```
void push()
{
 int val;
 Struct node *ptr = (Struct node*) malloc (sizeof (Struct
node));
 if (ptr == NULL)
{
 printf ("not able to push the element ");
}
else
{
 printf (" Enter the value ");
 scanf ("%d", &val);
 if (head == NULL)
 {
 ptr -> val = val;
 ptr -> next = NULL;
 head = ptr;
 }.
 else
 {
 ptr -> val = val;
 ptr -> next = head;
 head = ptr;
}.
```

```
printf ("Item pushed");
}
}

void pop()
{
  int item;
  struct node *ptr;
  if (head == NULL)
  {
   printf ("Underflow");
  }
  else
  item = head -> val;
  ptr = head;
  head = head -> next;
  free (ptr);
  printf ("Item popped");
 }
}
```

```
void display()
{
 int i;
 struct node *ptr;
 ptr=head;
 if(ptr==NULL)
 {
  printf("Stack is empty
 }.
 else
 {
  printf(" Printing Stack elements \n");
  while(ptr!=NULL)
  {
   printf("%d \n", ptr->val);
   ptr=ptr->next;
  }.
 }.
}
```

Output:

```
*** Menu ***
1. push 2. pop 3. show 4. Exit

Enter your choice : 1
Enter the value : 1

Enter your choice : 1
Enter the value : 2

Enter your choice : 1
Enter the value : 3.

Enter your choice : 3.
printing stack elements
3
2
1

Enter your choice : 2.
Item popped

Enter your choice : 3
stack elements
2
1
```

Enter your choice : 4

2b. Queue Implementation using Single linked list:

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

Struct node
{
  int data;
  Struct node * next;
};

Struct node * front;
Struct node * rear;

void insert () {
  Struct node *ptr;
  int item;
  ptr = (Struct node *) malloc (sizeof (Struct node));
  if (ptr == NULL)
  {
    printf ("In overflow In");
    return;
  }
edge {
  printf ("In Enter value: ");
```

```
        scanf ("%d", &item);
        ptr -> data = item;
        if (front == NULL)
        {
        front = ptr;
        rear = ptr;
        front -> next = NULL;
        rear -> next = NULL;
        }
        else
        {
        rear -> next = ptr;
        rear = ptr;
        rear -> next = NULL;
        }
    }
}

void delete()
{
    struct node *ptr;
    if (front == NULL)
    {
        printf ("Underflow");
        return;
    }
```

```
else
{
  ptr = front;
  front = front -> next;
  free(ptr);
}
}

void display()
{
  Struct node *ptr;
  ptr = front;
  if (front == NULL)
  {
   printf (" Empty Queue\n");
  }
  else
  {
   printf ("\n printing Values ...\n");
   while (ptr != NULL)
   {
   printf ("\n %d ", ptr -> data);
   ptr = ptr -> next;
   }
  }
}
```

Output:

* * * Main Menu * * *

1. Insert an element
2. Delete an element
3. Display the queue
4. exit.

Enter your choice: 1
Enter value 1

Enter your choice: 1
Enter value: 2

Enter your choice: 1
Enter value: 3.

Enter your choice: 3.
printing values...
1
2
3.

Sw
5/2/24

Enter your choice: 2
Enter your choice: 3
printing values...
2
3

Enter your choice: 4