8/09/20

6000419005-7
JUNAID. GIRKAR
~~Unpaid~~

DATA STRUCTURES TT1

Q1 Implement following functions of a stack using array

ANS
```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#define MAX 5

int st[MAX], top = -1;
```

(i)
```
// INSERTION
void push(int st[], int val)
{
    if(top == MAX-1)
    {
        printf("\n STACK OVERFLOW");
    }
    else
    {
        top++;
        st[top] = val;
    }
}
```

```
// DELETION
int pop(int st[])
{
    int val;
    if(top == -1)
    {
```

```
            printf ("\n STACK OVERFLOW");
            return -1;
        }
        else
        {
            val = st[top];
            top--;
            return val;
        }
}

// DISPLAYING
void display(int st[])
{
    int i;
    if (top == -1)
    {
      printf("\n STACK IS EMPTY");
    }
    else
    {
        for (i= top ; i>=0 ; i--)
        {
            printf("\n %d", st[i]);
        }
        printf("\n");
    }
}
```

B00041900S7
Junaid.

Q2 Write a function to insert an element in a linked list before a specific value present in the linked list

```
void insert_before (struct node **head, int value, int before)
{
    struct node * new_node = NULL;
    struct node * tmp = *head;
    new_node = (struct node *) malloc (sizeof (struct node));
    if (new_node == NULL)
    {
        printf ("Failed! Out of memory);
        return ;
    }
    new_node -> val = value;
    if ((*head) -> val == before)
    {
        new_node -> next = *head;
        *head = new_node;
        return;
    }
    while (tmp && tmp -> next)
    {
        if (tmp -> next -> val == before)
        {
            new_node -> next = tmp -> next;
            tmp -> next = new_node;
            return;
        }
        tmp = tmp -> next;
    }
}
```

B00041900S7
Junaid

Q3

0	1	2	3	4	5	6	7	8	9	10	11
10	12	35	40	45	50	80	82	85	90	100	110

Arr = [10, 12, 35, 40, 45, 50, 80, 82, 85, 90, 100, 110]

Using '0' indexing.

No of elements = n = 12 ;

x = 100

fib M = 13 (Nearest fibonacci no >= 12)

fib M_1 = 8

fib M_2 = 5

offset = -1

step 1:

i = min (offset + fib M_2 , n - 1)

= min (-1 + 5, 11)

i = 4

a[i] = 45 ; 45 < 100

∴ fib M = 8

fib M_1 = 5

fib M_2 = 3

offset = 4

step 2:

i = min (7, 11)

= 7

a[i] = 82 ; 82 < 100

∴ fib M = 5

fib M_1 = 3

fib M_2 = 2

offset = 7

Step 3 $i = \min(9, 11)$

$= 9$

$a[i] = 90$; $90 < 100$

$\therefore$ fib $m = 3$

fib $M_1 = 2$

fib $M_2 = 1$

offset $= 9$

step 4: $i = \min(10, 11)$

$= 10$

$a[i] = 100$

$\therefore x = 100$ found at index $i = 10$