



KSIT
K.S. INSTITUTE OF TECHNOLOGY

Kammavari Sangham (R)-1952

K.S.GROUP OF INSTITUTIONS

K.S. INSTITUTE OF TECHNOLOGY

ACCREDITED BY NAAC

(Approved by AICTE & Affiliated to VTU)

KANAKPURA ROAD, BENGALURU - 560 109

A P J ABDUL KALAM

"Tell me and I'll forget;
Show me and I may
remember; Involve me
and I'll understand"

GRAHAM BELL

MARIE CURIE

SRINIVASA
RAMANUJAN

J R D TATA

THOMAS ALVA EDISON

DENNIS RITCHIE

PRACTICAL RECORD

NAME : Monika.k.c

SEM/ BRANCH : III A CSE

SUBJECT & CODE : Data structure lab 18C5L38

USN :

1	K	S	1	8	C	S	0	5	2
---	---	---	---	---	---	---	---	---	---

K.S. INSTITUTE OF TECHNOLOGY

VISION

“To impart quality technical education with ethical values, employable skills and research to achieve excellence”

MISSION

- * To attract and retain highly qualified, experienced & committed faculty.
- * To create relevant infrastructure
- * Network with industry & premier institutions to encourage emergence of new ideas by providing research & development facilities to strive for academic excellence
- * To inculcate the professional & ethical values among young students with employable skills & knowledge acquired to transform the society



K S I T
K.S. INSTITUTE OF TECHNOLOGY

Kammavari Sangham (R)-1952

K.S. GROUP OF INSTITUTIONS

K.S. INSTITUTE OF TECHNOLOGY

ACCREDITED BY NAAC

(Approved by AICTE & Affiliated to VTU)

KANAKPURA ROAD, BENGALURU - 560 109

Laboratory Certificate

*This is to certify that Mr./Ms. ...Monika.k...c.....
has satisfactorily completed the course of experiments in
Data.....structures.....laboratory, Code..18CSL38...
prescribed by Visvesvaraya Technological University, Belgaum for
the.....3.....Semester B.E.....CSE.....Branch
in this College during the academic year 20.19... - 20.20...*

Name of the Candidate :...Monika.k...c.....

USN :...18CSL38052.....Subject (with code)...Data.....structures
laboratory (18CSL38)

Internal assessment marks awarded :

29	09	38
30	10	40

Signature of Staff Incharge

Date:

25/11/2019

Signature of Head of

Department

Head of the Department
Dept. of Computer Science & Engg
K.S. Institute of Technology
Bengaluru - 560 109

INSTRUCTIONS FOR MAINTAINING THE PRACTICAL RECORD

1. Record should be written neatly in ink on the right side page only, left side pages being reserved for diagrams and graphs in pencil.
2. Record should contain
 - Number and name of the Experiment
 - The date
 - Principle
 - Procedure
 - Observation calculation (to be entered on the left hand side pages in neat tabular forms wherever applicable)
 - Results

Output:-

IMPLEMENTATION OF OPERATIONS ON ARRAY

MENU

1:CREATE

2:DISPLAY

3:INSERT

4.DELETE

Enter Your choice :: 1

ARRAY CREATION

Enter the number of elements in array :: 5

Enter the elements

12 13 14 15 16

MENU

1:CREATE

2:DISPLAY

3:INSERT

4:DELETE

Enter your choice :: 2

ARRAY ELEMENTS ARE ::

12 13 14 15 16

MENU

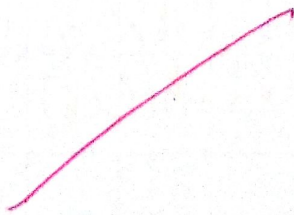
1:CREATE

2: DISPLAY

3: INSERT

4:DELETE

Enter your choice :: 3



- Design, Develop and Implement a menu driven Program in C for the following Array Operations
- Creating an Array of N Integer Elements
 - Display of Array Elements with Suitable Headings
 - Inserting an Element (ELEM) at a given valid position (POS)
 - Deleting an Element at a given valid position (POS)
 - Exit.

Support the program with functions for each of the above Operations.

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 10
int a[MAX], u;
void create(), display(), insert, delete();
void main()
{
    int ch;
    printf("\n IMPLEMENTATION OF OPERATIONS ON ARRAY\n");
    for(;;)
    {
        printf("\n MENU\n");
        printf("\n 1: CREATE\n 2: DISPLAY\n 3: INSERT\n 4: DELETE\n");
        printf("\n Enter your choice:");
        scanf("%d", &ch);
        switch(ch)
```

ELEMENT INSERTION

Enter the position less or equal to $n+1$ to insert
the element :: 4

Enter the element to insert :: 100

MENU

1: CREATE

2: DISPLAY

3: INSERT

4: DELETE

Enter your choice :: 2 elements in array :: 5

ARRAY ELEMENTS ARE ::

13 13 14 100 15 16

MENU

1: CREATE

2: DISPLAY

3: INSERT

4: DELETE

Enter your choice :: 3

ELEMENT INSERTION

Enter the position less or equal to $n+1$ to
insert the element :: 9


```
{  
    case 1: create();  
        break;  
    case 2: display();  
        break;  
    case 3: insert();  
        break;  
    case 4: delete();  
        break;  
    case 5: exit(0);  
}
```

```
}  
  
void create()  
{
```

```
    int i;  
    printf("\n ARRAY CREATION\n");  
    printf("\nEnter the number of elements in an array:");  
    scanf("%d", &n);  
    printf("\nEnter the elements\n");  
    for(i=0; i<n; i++)  
        scanf("%d", &a[i]);  
}
```

```
void display()  
{
```

```
    int i;  
    printf("\n ARRAY ELEMENTS ARE:\n");
```

Inuvalid position

MENU

- 1: CREATE
- 2: DISPLAY
- 3: INSERT
- 4: DELETE

Enter your choice::4

ELEMENT DELETION

Enter the position less than or equal to n to delete element::4

Deleted item is::100

MENU

- 1: CREATE
- 2: DISPLAY
- 3: INSERT
- 4: DELETE

Enter your choice::2

ARRAY elements are::

12 13 14 15 16

```
for(i=0; i<n; i++)
    printf("%d\t", a[i]);
}

void insert()
{
    int i, ele, pos;
    printf("\n ELEMENT INSERTION\n");
    if(n == MAX)
        printf("\n ARRAY IS FULL, INSERTION IS NOT
                POSSIBLE\n");
    else
    {
        printf("\n Enter the position less or equal to n+1 to
                insert the element::");
        scanf("%d", &pos);
        if(pos <= n+1)
        {
            printf("\n Enter the element to insert::");
            scanf("%d", &ele);
            for(i=n-1; i >= pos-1; i--)
                a[i+1] = a[i];
            a[pos-1] = ele;
            ++n;
        }
        else
            printf("\n Invalid position\n");
    }
}
```

void delete()

```
int item, i, pos;
printf("\n ELEMENT DELETION\n");
if (n == 0)
    printf("\n ARRAY IS EMPTY DELETION NOT POSSIBLE\n");
else
{
    printf("\n Enter the position less than or equal to n to
           delete element ::");
    scanf("%d", &pos);
    if (pos <= n)
    {
        item = a[pos-1];
        printf("\n DELETED ITEM IS :: %d\n", item);
        --n;
        for (i = pos-1; i < n; i++)
            a[i] = a[i+1];
    }
    else
        printf("\n Invalid position\n");
}
```

observation	10
Record	10
Using	10
total	30
Sign	ms 19/11/19

OUTPUT

1) Read main string, pattern string and replace string

Enter a string::

KSITRNSIT

Enter a pattern string::

SIT

Enter a replace string::

AAAAA

The resultant string is:

KAAAAATRNSIT

2) Read main string, pattern string and replace string

Enter a string::

KSITRNSIT

Enter a pattern string::

IT

Enter a replace string::

BBBBB

The resultant string

No pattern in the string

Design, Develop and Implement a Program in C for the following Operations On strings

a) Read a main String (STR), a Pattern String (PAT) and a Replace String (REP)

b) Perform Pattern Matching Operation: Find and Replace suitable messages in case 'PAT' does not exist in STR

Support the program with functions for each of the above Operations. Don't use Built-in functions.

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

```
void read(char str[], char pat[], char rep[])
```

```
{
```

```
    printf("\n Enter a string::\n");
```

```
    gets(str);
```

```
    printf("\n Enter a pattern string::\n");
```

```
    gets(pat);
```

```
    printf("\n Enter a replace string::\n");
```

```
    gets(rep);
```

```
}
```

```
void patreplace(char str[], char pat[], char rep[],  
                char ans[])
```

```
{
```

```
    int i, m, c, j, count, k;
```

```
    i = m = c = j = count = 0;
```

```
    while (str[c] != '\0')
```

```
    {
```

```
        if (str[m] == pat[i])
```

```
        {
```

DATE 22-08-19
EXP. NO. 2

EXPT. TITLE :

PROGRAM - 02

PAGE NO. 6

```
i++;  
m++;  
if (pat[i] == '\0')  
{  
    count++;  
    for (k=0; rep[k] != '\0'; k++; j++)  
        ans[j] = rep[k];  
    i=0;  
    c=m;  
}  
}  
else  
{  
    ans[j] = str[c];  
    j++;  
    c++;  
    m=c;  
    i=0;  
}  
}  
ans[j] = '\0';  
if (count == 0)  
    printf("No pattern in the string\n");  
else  
    printf("In The resultant string is\n%s", ans);  
}
```

```
void main()
```

```
{
```

OUTPUT:-

Op 1:-

Read main string, pattern string and replace string
Enter a string::

KSITRNIST

Enter a pattern string::

IIT

Enter a replace string::

BBBBB

No pattern in the string

Op 2:-

Read main string, pattern string and replace string
Enter a string::


```
char str[100], pat[100], rep[100], ans[100];  
printf("Read main string, pattern string and replace  
string\n");  
read(str, pat, rep);  
patreplace(str, pat, rep, ans);  
}
```

Observation -	9
Record -	10
viva -	9
total -	28
Sign -	28 +9 total 9

Output

Stack Operations

1. Push
2. Pop
3. Character Stack to check palindrome
4. Overflow - Underflow
5. Display
6. Exit

Enter Your Choice: 1

Enter element to push: 10

1. Push
2. Pop
3. Character Stack to check palindrome
4. Overflow - Underflow
5. Display
6. Exit

Enter Your Choice: 1

Enter the element to push: 20

1. Push
2. Pop
3. Character Stack to check palindrome

Design, Develop and Implement a menu driven Program in C for the following Operations On STACK Of Integers (Array Implementation Of Stack with maximum size MAX)

- a) Push an Element On to stack
- b) Pop an Element from stack
- c) Demonstrate how stack can be used to check palindrome
- d) Demonstrate Overflow and Underflow situations On stack
- e) Display the status of Stack.
- f) Exit.

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

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

```
#include <string.h>
```

```
#define max 5
```

```
void push(), pop(), display(), overflow-underflow(),  
palindrome();
```

```
int stack[max], top = -1, top1 = -1;
```

```
void main()
```

```
{
```

```
int ch, item;
```

```
printf("In stack Operations\n");
```

```
for(;;)
```

```
{
```

```
printf("In 1. Push\n 2. Pop\n 3. Character Stack to check  
palindrome\n 4. Overflow - Underflow\n 5. Display  
In 6: Exit\n");
```

```
printf("In Enter, your, Choice:");
```

4. Overflow-Underflow
5. Display
6. Exit

Enter your choice: 5

Elements on stack are::

- 50
- 40
- 30
- 20
- 10

1. Push
2. Pop
3. Character stack to check palindrome
4. Overflow-Underflow
5. Display
6. Exit

Enter your choice: 2

Pop element: 50

1. Push
2. Pop
3. Character stack to check palindrome
4. Overflow-Underflow
5. Display
6. Exit

```
scanf("%d", &ch);  
switch(ch)  
{  
    case 1: push();  
           break;  
    case 2: pop();  
           break;  
    case 3: palindrome();  
           break;  
    case 4: overflow-underflow();  
           break;  
    case 5: display();  
           break;  
    case 6: exit(0);  
}
```

```
void push()  
{
```

```
    int ele;  
    if(top == max - 1)  
        printf("\n stack overflow\n");  
    else
```

```
        {  
            printf("\n Enter element to push:");  
            scanf("%d", &ele);  
            stack[++top] = ele;  
        }
```

Enter your choice: 3

Enter the string to check palindrome:: Bhauya

String Bhauya is not palindrome

1. Push
2. Pop
3. Character Stack to check palindrome
4. Overflow - Underflow
5. Display
6. Exit

Enter your choice: 3

Enter the string to check palindrome :: dad

String dad is palindrome

String dad

1. Push
2. Pop
3. Character Stack to check palindrome
4. Overflow - Underflow
5. Display
6. Exit

Enter your choice: 6

```
}  
void pop()  
{  
    int num;  
    if(top == -1)  
        printf("\n Underflow\n");  
    else  
    {  
        num = stack[top--];  
        printf("\n popped element :: %d\n", num);  
    }  
}
```

```
void overflow_underflow()  
{  
    if(top == max - 1)  
        printf("\n Stack full\n");  
    if(top == -1)  
        printf("\n Stack underflow\n");  
}
```

```
void display()  
{  
    int i;  
    if(top == -1)  
        printf("\n Stack is Empty\n");  
    else  
    {  
        printf("\n Elements on stack are ::\n");  
        for(i = top; i >= 0; i--)
```

```

    printf("%d\n", stack[i]);
  }
}

```

```

void palindrome()
{

```

```

  int i=0, j=0;

```

```

  char s[10], st[10], rev[10];

```

```

  printf("\n Enter the string to check palindrome:");

```

```

  scanf("%s", s);

```

```

  while(s[i]!='\0');

```

```

  {
    st[++top1] = s[i];
    i++;
  }

```

```

  while(top1 != -1)
  {

```

```

    rev[j] = st[top1--];

```

```

    j++;
  }

```

```

  rev[j] = '\0';

```

```

  if(strcmp(s, rev) == 0)

```

```

    printf("\n String %s is palindrome\n", s);

```

```

  else

```

```

    printf("\n String %s is not palindromeln", s);
  }
}

```

observation	9
Reverd	10
Uua	9
total	me total 9 KSM

Design, Develop and Implement a Program in C for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the Operators: +, -, *, /, % (Remainder), ^ (Power) and alphanumeric Operands

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

```
typedef enum { lparen, rparen, plus, minus, mul, div, mod, pow, eos, op } pred;
```

```
int isp[] = { 0, 19, 12, 12, 13, 13, 13, 14, 0 };
```

```
int icp[] = { 20, 19, 12, 12, 13, 13, 13, 15, 0 };
```

```
char exp[50], symbol;
```

```
pred stack[50];
```

```
int top = 0, n = 0;
```

```
pred readtoken()
```

```
{  
symbol = exp[n++];
```

```
switch(symbol)
```

```
{  
case '(': return lparen;
```

```
case ')': return rparen;
```

```
case '+': return plus;
```

```
case '-': return minus;
```

```
case '*': return mul;
```

```
case '/': return div;
```

```
case '%': return mod;
```

```
case '^': return pow;
```

```
case '0': return eos;
```

DATE 19-09-19

EXPT. TITLE :

PROGRAM-04

EXP. NO. 04

PAGE NO.

14

```
} default: return op;
}
}
void displaytoken(pred token)
{
switch(token)
{
case plus: printf("+");
            break;
case minus: printf("-");
            break;
case mul: printf("*");
            break;
case div: printf("/");
            break;
case mod: printf("%");
            break;
case pow: printf("^");
            break;
}
}
```

```
void push(pred item)
```

```
stack[++top] = item;
```

```
pred pop()
```

```
return stack[top--];
```

```
}  
void infixtopost()  
{  
    pred token, token1;  
    stack[0] = eos;  
    token = readtoken();  
    while(token != eos)  
    {  
        if(token == op)  
        {  
            printf("%c", symbol);  
        }  
        else if(token == rparen)  
        {  
            while(stack[top] != lparen)  
            {  
                token1 = pop();  
                displaytoken(token1);  
            }  
            pop();  
        }  
        else  
        {  
            while(isp[stack[top]] >= icp[token])  
            {  
                token1 = pop();  
                displaytoken(token1);  
            }  
        }  
    }  
}
```

OUTPUT 1

PROGRAM TO CONVERT INFIX EXP TO POSTFIX EXP

Enter infix expression: $a^{\wedge}(b+c\%d)+e$
Postfix expression: $abcd\ \%+^{\wedge}e+$

Output 2

PROGRAM TO CONVERT INFIX EXP TO POSTFIX EXP

Enter infix expression: $a+(b*c-(d/e)+f)$
Postfix expression: $abc^*de/-f++$

```

    push(token);
}
token = readtoken(symbol, n);
}
while((token = pop()) != eos)
{
    displaytoken(token);
}
printf("\n");
}
void main()
{
    printf("PROGRAM TO CONVERT INFIX TO POSTFIX Explain");
    printf("Enter infix expression :");
    gets(exp);
    printf("Postfix expression :");
    infixtopost();
}

```

Observation	-	10
Record	-	10
used	-	10
total	-	30
sign	-	ms 20/19 20/19

Output 1

EVALUATION OF POSTFIX EXPRESSION

Enter the postfix expression
 ab^*cd/t

Enter a value of a :: 2

Enter a value of b :: 3

Enter a value of c :: 8

Enter a value of d :: 4

The result of the evaluation is: 8

Output 2

EVALUATION OF POSTFIX EXPRESSION

Enter the postfix expression
 $93\%24^1+$

The result of the evaluation is: 16

Design, Develop and Implement a Program in C for the following Stack Applications

- Evaluation of Suffix expression with single digit Operands and Operators: +, -, *, /, %, ^
- Solving Tower of Hanoi problem with n disks.

a) Evaluation of Suffix expression

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <math.h>
#include <stdlib.h>
#define STACK_SIZE 50
int stack[STACK_SIZE];
int top = -1;
void push(int a)
{
    stack[++top] = a;
    return;
}
int pop()
{
    return (stack[top--]);
}
int eval(char s[])
{
    int i;
```

```
int temp, op1, op2, result;  
char symbol[1];  
for(i=0; i < strlen(s); i++)  
{  
    if(isalnum(s[i]))  
    {  
        if(isdigit(s[i]))  
        {  
            symbol[0] = s[i];  
            temp = atoi(symbol);  
            push(temp);  
        }  
        else  
        {  
            printf("\n\nEnter a value of '%c':", s[i]);  
            scanf("%d", &temp);  
            push(temp);  
        }  
    }  
    else  
    {  
        op2 = pop();  
        op1 = pop();  
        switch(s[i])  
        {  
            case '^': result = pow(op1, op2);  
                    break;
```



```
case '*' : result = op1 * op2;  
          break;  
case '/' : result = op1 / op2;  
          break;  
case '%' : if (op2 == 0)  
            {  
              printf("Divide by zero error\n\n");  
              exit(0);  
            }  
          else  
            result = op1 % op2;  
            break;  
case '+' : if (op2 == 0)  
            {  
              printf("Divide by zero error\n\n");  
              exit(0);  
            }  
          else  
            result = op1 + op2;  
            break;  
case '-' : result = op1 - op2;  
          break;  
default:  
  printf("Invalid Expression\n\n");  
  exit(0);  
}
```

DATE ... 19-09-19

EXP. NO. ... 05

EXPT. TITLE :

PROGRAM -05

PAGE NO. 20

```
    push(result);  
}  
  
result = pop();  
if (top != -1)  
{  
    printf("Invalid Expression\n");  
    exit(0);  
}  
return result;  
  
void main()  
{  
    char s[50];  
    printf("||| EVALUATION OF POSTFIX EXPRESSION \n");  
    printf("\n\n Enter the postfix expression\n");  
    gets(s);  
    printf("\n\n The result of the evaluation is : %d\n",  
           eval(s));  
    return ;  
}
```

Output 1

Enter the number of disks: 3

Sequence of disk

Move disk 1 from A to C

Move disk 2 from A to B

Move disk 1 from C to B

Move disk 3 from A to C

Move disk 1 from B to A

Move disk 2 from B to C

Move disk 1 from A to C

Output 2

Enter the number of disks

Sequence of disk

Move disk 1 from A to B

Move disk 2 from A to C

Move disk 1 from B to C

b) Solving Tower of Hanoi problem with n disks.

```
#include <stdio.h>
int tower (int n, char s, char t, char d)
{
    if (n == 0)
        return 0;
    tower (n-1, s, d, t);
    printf ("Move disk %d from %c to %c\n", n, s, d);
    tower (n-1, t, s, d);
    return 0;
}

void main()
{
    char A, B, C;
    int n;
    printf ("Enter the number of elements\n");
    scanf ("%d", &n);
    printf ("Sequence of disk\n");
    tower (n, 'A', 'B', 'C');
    printf ("\n");
}
```

Observation -	9
Record -	10
Uiva -	9
total -	28
Sign -	ms 2/11/19

Output:-

IMPLEMENTATION OF CIRCULAR QUEUE

MENU

1. INSERT
2. DELETE
3. DISPLAY
4. EXIT

Enter Your Choice :: 1

Enter the item to be inserted :: a

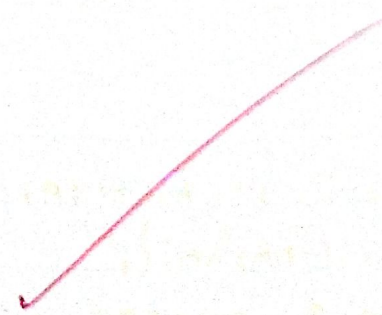
IMPLEMENTATION OF CIRCULAR QUEUE

MENU

1. INSERT
2. DELETE
3. DISPLAY
4. EXIT

Enter Your Choice :: 1

Enter the item to be inserted :: b



Design, Develop and Implement a menu driven Program in C for the following Operations On Circular QUEUE of Characters

- a) Insert an Element into circular QUEUE
- b) Delete an Element from circular QUEUE
- c) Demonstrate Overflow and Underflow situations On Circular QUEUE
- d) Display the status of Circular QUEUE
- e) Exit.

Support the program with appropriate functions for each of the above Operations.

```
#include <stdio.h>
#include <stdlib.h>
#include <stdio_ext.h>
#define max 5
int front = 0, rear = -1, count = 0;
char circularq[max];
void insert(), delete(), display();
void main()
{
    int choice;
    for(;;)
        printf("\n IMPLEMENTATION OF CIRCULAR QUEUE \n");
        printf("\n MENU \n");
        printf("\n 1. INSERT \n 2. DELETE \n 3. DISPLAY \n 4. EXIT \n \n");
```

IMPLEMENTATION OF CIRCULAR QUEUE

MENU

1. INSERT
2. DELETE
3. DISPLAY
4. EXIT

Enter Your Choice:: 1

Enter the item to be inserted:: c

IMPLEMENTATION OF CIRCULAR QUEUE

MENU

1. INSERT
2. DELETE
3. DISPLAY
4. EXIT

Enter Your Choice:: 3

Contents of the Circular queue are:

a b c

IMPLEMENTATION OF CIRCULAR QUEUE

MENU

```
printf("\nEnter Your choice:");  
scanf("%d", &choice);  
switch(choice)  
{  
    case 1: insert();  
           break;  
    case 2: delete();  
           break;  
    case 3: display();  
           break;  
    default: exit(0);  
}
```

```
}  
}  
void insert()
```

```
{  
    char item;  
    if(count == max)  
        printf("\n CIRCULAR QUEUE IS FULL\n");  
    else
```

```
{  
    printf("\nEnter the item to be inserted:");  
    purge(stdin);  
    scanf("%c", &item);  
    rear = (rear + 1) % max;  
    circularq[rear] = item;  
    count ++;  
}
```


1. INSERT
2. DELETE
3. DISPLAY
4. EXIT

Enter Your Choice: 1

Enter the item to be inserted::d

IMPLEMENTATION OF CIRCULAR QUEUE:

MENU

1. INSERT
2. DELETE
3. DISPLAY
4. EXIT

Enter Your Choice:: 1

Enter the item to be inserted::f

IMPLEMENTATION OF CIRCULAR QUEUE:

MENU

1. INSERT
2. DELETE
3. DISPLAY
4. EXIT

}

void delete()

{

char ch;

if(count == 0)

printf("\n CIRCULAR QUEUE IS EMPTY\n");

else

{

ch = circularq[front];

printf("\n The deleted element is:");

printf("%c", ch);

front = (front + 1) % max;

count--;

}

}

void display()

{

int i, j;

char item;

if(count == 0)

printf(" CIRCULAR QUEUES IS EMPTY\n");

else

{

printf("\n contents of the circular queue are: \n");

j = front;

for(i = 1; i <= count; i++)

{

item = circularq[j];

Enter Your Choice::3

Contents of the Circular queue are:

a b c d

IMPLEMENTATION OF CIRCULAR QUEUE

MENU

1. INSERT
2. DELETE
3. DISPLAY
4. EXIT

Enter Your Choice::1

CIRCULAR QUEUE IS FULL

IMPLEMENTATION OF CIRCULAR QUEUE

MENU

1. INSERT
2. DELETE
3. DISPLAY
4. EXIT

Enter Your Choice::2

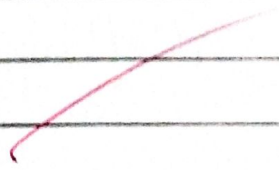
The deleted element is::a

IMPLEMENTATION OF CIRCULAR QUEUE

MENU

1. INSERT
2. DELETE
3. DISPLAY
4. EXIT

```
printf("%c", item);  
j = (j + 1) % max;  
printf("%t");  
}  
}  
}
```



Enter Your Choice :: 2

The deleted element is :: b

IMPLEMENTATION OF CIRCULAR QUEUE

MENU

1. INSERT
2. DELETE
3. DISPLAY
4. EXIT

Enter Your choice :: 1

Enter the item to be inserted :: x

IMPLEMENTATION OF CIRCULAR QUEUE

MENU

1. INSERT
2. DELETE
3. DISPLAY
4. EXIT

Enter Your Choice :: 1

Enter the item to be inserted :: y

IMPLEMENTATION OF CIRCULAR QUEUE

MENU

1. INSERT
2. DELETE
3. DISPLAY

4. EXIT

Enter Your Choice::1

CIRCULAR QUEUE IS FULL

IMPLEMENTATION OF CIRCULAR QUEUE

MENU

1. INSERT

2. DELETE

3. DISPLAY

4. EXIT

Enter Your Choice::3

Contents of the circular queue are:

c d f x y

IMPLEMENTATION OF CIRCULAR QUEUE

MENU


1. INSERT

2. DELETE

3. DISPLAY

4. EXIT

Enter Your Choice::4



DATE 19-09-19
EXP. NO. 06

EXPT. TITLE :

PROGRAM-06

PAGE NO. 27

Observation	-	9
Reward	-	10
Winnings	-	9
Total	-	28
Sign	-	ms 4/11/19

Output:-

SINGLY LINKED LIST OPERATIONS

MENU

- 1: List creation using Insert front
- 2: Display and count
- 3: List as Stack
- 4: Insert at end
- 5: Delete at end

Enter choice::1

Enter how many student information you want to enter:3

Enter information of student 1

Enter student data such as USN, NAME, BRANCH, SEM, PHNO

1ks13cs015

Chaitra

CSE

1

9288891919

Enter information of student 2

Enter student data such as USN, NAME, BRANCH, SEM, PHNO

1KS13CS002

Sharath

CSE

2

8991791829

Design, Develop and Implement a menu driven Program in C for the following operations on SLL of Student Data with the fields: USN, Name, Branch, Sem, PhNo

- create a SLL of N students Data by Using front insertion
- Display the status of SLL and count the no. of nodes in it
- Perform Insertion / Deletion at End of SLL
- Perform Insertion / Deletion at Front of SLL (Demonstration of stack)

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

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

```
struct node
```

```
{
```

```
char usn[12], name[20], branch[10], phno[15];
```

```
int sem;
```

```
struct node *link;
```

```
};
```

```
typedef struct node *NODE;
```

```
NODE start = NULL;
```

```
NODE start2 = NULL;
```

```
NODE create_node()
```

```
{
```

```
NODE ptr;
```

```
ptr = (NODE) malloc (sizeof (struct node));
```

```
if (ptr == NULL)
```

```
{
```

```
printf ("Insufficient Memory\n");
```

```
exit(0);
```

Enter information of student 3

Enter student data such as USN, NAME, BRANCH, SEM, PHNO

1K513C5004

Bharath

CSE

4

7822781426

SINGLY LINKED LIST OPERATIONS

MENU

1: List creation using Insert front

2: Display and Count

3: List as Stack

4: Insert at end

5: Delete at end

Enter choice::2

SINGLY LINKED LIST IS:

|1K513C5004|Bharath|CSE|4|7822781426| → |1K513C5002|Sharath|
CSE|2|8991791829| → |1K513C5015|Chaithra|CSE|1|9288891919| →

NUMBER OF NODE IN LIST::3

SINGLY LINKED LIST OPERATIONS

MENU

1: List creation using Insert front

2: Display and Count

3: List as Stack

4: Insert at End.

DATE 03-10-19

EXPT. TITLE :

PROGRAM - 07

EXP. NO. 07

PAGE NO. 29

```
}  
printf("\n Enter student data such as USN, NAME, BRANCH, PHNO,  
SEM\n\n");  
scanf("%s%s%s%s%d", ptr->usn, ptr->name, ptr->branch, ptr->  
phno, &ptr->sem);  
ptr->link = NULL;  
return ptr;  
}
```

```
NODE insert_front(NODE start)
```

```
{  
    NODE ptr;  
    ptr = create_node();  
    if (start == NULL)  
        start = ptr;  
    else  
    {  
        ptr->link = start;  
        start = ptr;  
    }  
    return start;  
}
```

```
NODE insert_end(NODE start)
```

```
{  
    NODE ptr, temp;  
    ptr = create_node();  
    if (start == NULL)  
        start = ptr;  
    else
```

```
temp = start;
```

5: Delete at end

Enter choice::4

Enter student data such as USN, NAME, BRANCH, PHNO, SEM

1KS13C5005

Chinnu

CSE

9838712347

4

-

SINGLY LINKED LIST OPERATIONS

MENU

1: List creation using Insert front

2: Display and count

3: List as stack

4: Insert at end

5: Delete at end

Enter choice::2

SINGLY LINKED LIST IS::

1KS13C5004 | Bharath | CSE | 7822781426 | 4 | → 1KS13C5002 | Sharath
CSE | 8991791829 | 2 | → 1KS18C5001 | Chaithrol | CSE | 9288891919 | →
Chinnu | CSE | 9838712347 | 4 | →

NUMBER OF NODE IN LIST::4

SINGLY LINKED LIST OPERATIONS

```
while (temp->link != NULL)
{
    temp = temp->link;
}
temp->link = pbr;
}
return start;
}
NODE delete_front (NODE start)
{
    NODE temp;
    if (start == NULL)
        printf ("\n LIST EMPTY \n");
    else
    {
        temp = start;
        start = start->link;
        printf ("\n Deleted node is \n");
        printf ("%s | %s | %s | %s | %d \n \n", temp->usr, temp->name,
            temp->branch, temp->phno, temp->sem);
        free(temp);
    }
    return start;
}
NODE delete_end (NODE start)
{
    NODE p, temp;
    if (start == NULL)
    {
        printf ("\n LIST EMPTY \n");
    }
    return;
```

MENU

- 1: List creation using Insert front
- 2: Display and count
- 3: List as Stack
- 4: Insert at end
- 5: Delete at end

Enter choice::2

SINGLY LINKED LIST IS:

|1ks13c5004| Bharath | cse | 7822781426 | 4 | → |1ks13c5002| Sharath |
cse | 8991791829 | 2 | → |1ks13c5015| Chaithra | cse | 9288891919 | 1 | →

NUMBER OF NODE IN LIST::3

SINGLY LINKED LIST OPERATIONS

MENU

- 1: List creation using Insert front
- 2: Display and count
- 3: List as Stack
- 4: Insert at end
- 5: Delete at end

Enter choice::5

Deleted node is

|1ks13c5001| Chaithra | cse | 9288891919 | 1 |

SINGLY LINKED LIST OPERATIONS

MENU

```
}
temp = start
if (temp->link == NULL)
{
    start = NULL;
}
else
{
    temp = start;
    while (temp->link != NULL)
    {
        p = temp;
        temp = temp->link;
    }
    p->link = NULL;
}
printf("\n Deleted node is \n");
printf("\n |%s| %s| %s| %s| %d", temp->usr, temp->name,
        temp->branch, temp->phoneno, temp->sem);
free(temp);
return start;
}

void display (NODE start)
{
    NODE temp;
    if (start == NULL)
        printf("\n LIST EMPTY");
    else
    {
        int count = 0;
```

- 1: List creation using Insert front
- 2: Display and count
- 3: List as Stack
- 4: Insert at end
- 5: Delete at end

Enter choice :: 3

OPERATIONS ON STACK

LIST AS STACK

1. PUSH
2. POP
3. DISPLAY
4. EXIT

Enter choice ::

1

Enter student data such as USN, NAME, BRANCH, PHNO, SEM

1KS1315006

VINUTHA

CSE

3

8927679912

LIST AS STACK

1. PUSH
2. POP
3. DISPLAY
4. EXIT


```
temp = start;
printf("\n SINGLY LINKED LIST IS:\n");
while(temp != NULL)
{
    printf("|%s|%s|%s|-%s|-%d|->", temp->usr, temp->name,
        temp->branch, temp->phno, temp->sem);
    temp = temp->link;
    count++;
}
printf("\n NUMBER OF NODE IN LIST::%d\n\n", count);
}
```

```
void list_stack(NODE start1)
```

```
{
    int choice;
    printf("\n OPERATIONS ON STACK\n");
    for(;;)
    {
        printf("\n\n LIST AS STACK\n\n");
        printf("\n 1: PUSH\n 2: POP\n 3: DISPLAY\n 4: EXIT\n");
        printf("\n Enter choice::\n");
        scanf("%d", &choice);
        switch(choice)
        {
            case 1: start1 = insert_front(start1);
                    break;
            case 2: start1 = delete_front(start1);
                    break;
            case 3: display(start1); break;
        }
    }
}
```

Enter choice::

1

Enter student data such as USN, NAME, BRANCH, PHNO, SEX

1KS1315010

BHARATH

CSE

8977561328

7

LIST AS STACK

1. PUSH

2. POP

3. DISPLAY

4. EXIT

Enter choice::

3

SINGLY LINKED LIST IS:

1KS1315010|BHARATH|CSE|8977561328|7| → |1KS1315008|
DVAUTH|CSE|8992777826|5| → |1KS1315006|VINUTHA|CSE|
8927679912|2| →

NUMBER OF NODE IN LIST :: 3

LIST AS STACK

1. PUSH

2. POP

3. DISPLAY

4. EXIT

```
default: exit(0);  
}
```

```
void main()  
{  
int ch, n, i;  
for(i=1; i<=n; i++)  
{  
printf("\n SINGLY LINKED LIST OPERATIONS\n");  
printf("\n MENU\n");  
printf("\n 1: list creation using Insert front\n 2: Display and count\n 3: Insert at end\n 4: Delete at end\n 5: list as Stack\n");  
printf("Enter choice::");  
scanf("%d", &ch);  
switch(ch)  
{  
case 1: printf("\n\n Enter how many student information you want enter::");  
scanf("%d", &n);  
for(i=1; i<=n; i++)  
{  
printf("\n\n Enter information of student %d\n\n", i);  
start = insert_front(start);  
}  
break;  
case 2: display(start);  
break;  
case 3: start = insert_end(start);
```

Enter choice::

2

Deleted element is |1k513c5010|Bharath|cse|8977561328|7|

LIST AS STACK

1: PUSH

2: POP

3: DISPLAY

4: EXIT

Enter choice:: 4

DATE 03-10-19.....

EXPT. TITLE :

PROGRAM - 07

EXP. NO. 07.....

PAGE NO.

34

```
        break;
    case 4: start = delete_end(start);
        break;
    case 5: start1 = list_stack(start1);
        break;
    default: exit(0);
}
```

Observation	9
Record	10
uiva	10
total	29
sign	$\frac{29}{411119}$

OUTPUT

MENU

- 1: CREATE DLL of N EMPLOYEE USING INSERT END
- 2: DISPLAY and COUNT
- 3: DISPLAY INSERT END
- 4: DELETE END
- 5: INSERT FRONT
- 6: DELETE FRONT
- 7: DLL AS DEQUEUE
- 8: EXIT

Enter the choice::3

Enter the employee data such as SSN, NAME, DEPARTMENT, DESIGNATION, PHONENO AND SALARY.

300

SUVESH

SALES

DIRECTOR

8569321456

30000

MENU

- 1: CREATE DLL of N EMPLOYEE USING INSERT END
- 2: DISPLAY and COUNT
- 3: INSERT END
- 4: DELETE END
- 5: INSERT FRONT
- 6: DELETE FRONT

Design, Develop and Implement a menu driven Program in C for the following operations on Doubly Linked List (DLL) of Employee Data with the fields: SSN, Name, Dept, Designation, Sal, PhNo

- Create a DLL of N Employees Data by using end insertion
- Display the status of DLL and count the no. of nodes in it
- Perform Insertion and Deletion at End of DLL
- Perform Insertion and Deletion at front of DLL
- Demonstrate how this DLL can be used as Double Ended Queue
- Exit.

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

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

```
struct node
```

```
{
```

```
    struct node *llink;
```

```
    int ssn;
```

```
    char name[20], dept[20], desi[20], phno[12];
```

```
    float sal;
```

```
    struct node *rlink;
```

```
};
```

```
typedef struct node *NODE;
```

```
NODE start = NULL;
```

```
NODE create_node()
```

```
{
```

```
    NODE ptr;
```

```
    ptr = (NODE) malloc (sizeof (struct node));
```

```
    if (ptr == NULL)
```

7: DLL AS DEQUEUE

8: EXIT

Enter the choice::5

Enter employee data such as SSN, NAME, DEPARTMENT,
DESIGNATION, PHONENO AND SALARY

50

Diya

PRODUCTION

MANAGER

8526545234

25000

MENU

1: CREATE DLL of N EMPLOYEE USING INSERT END

2: DISPLAY and COUNT

3: INSERT END

4: DELETE END

5: INSERT FRONT

6: DELETE FRONT

7: DLL AS DEQUEUE

8: EXIT

Enter the choice::2

The nodes of the doubly linked list are:

←>|50|Diya|PRODUCTION|MANAGER|8526545234|25000.0000
←>|100|Ananya|SALES|MANAGER|9658698567|25000.000000|


```
{
printf(" \n\n INSUFFICIENT MEMORY \n\n");
exit(0);
}
```

```
printf(" Enter employee data such as SSN, NAME, DEPARTMENT  
DESIGNATION, PHONENO AND SALARY \n");
```

```
scanf("%d%s%s%s%s%", &pb->ssn, pb->name, pb->dept,  
pb->desi, pb->phno, &pb->sal);
```

```
pb->llink = NULL;
```

```
pb->rlink = NULL;
```

```
return pb;
```

```
NODE insert_front(NODE start)
```

```
NODE pb;
```

```
pb = create_node();
```

```
if(start == NULL)
```

```
start = pb;
```

```
else
```

```
{
```

```
pb->rlink = start;
```

```
start->llink = pb;
```

```
start = pb;
```

```
}
```

```
return start;
```

```
}
```

```
NODE insert_end(NODE start)
```

```
NODE pb, temp;
```

<->|200|Zaina|PRODUCTION|DIRECTOR|7412589678|25000.000000
<->|300|Suresh|SALES|DIRECTOR|8569321456|30000.000000

Number of node in doubly link list are::4

MENU

- 1: CREATE DLL of N EMPLOYEE USING INSERT END
- 2: DISPLAY and COUNT
- 3: INSERT END
- 4: DELETE END
- 5: INSERT FRONT
- 6: INSERT FRONT
- 7: DLL AS DEQUEUE
- 8: EXIT

Enter the choice:: 4

DELETED END NODE IS::

|300|Suresh|SALES|DIRECTOR|8569321456|30000.000000)

MENU

- 1: CREATE DLL of N EMPLOYEE USING INSERT END
- 2: DISPLAY and COUNT
- 3: INSERT END
- 4: DELETE END
- 5: INSERT FRONT
- 6: DELETE FRONT
- 7: DLL AS DEQUEUE
- 8: EXIT

```
ptr = create_node();  
if (start == NULL)  
    start = ptr;  
else  
{  
    temp = start;  
    while (temp->rlink != NULL)  
    {  
        temp = temp->rlink;  
    }  
    temp->rlink = ptr;  
    ptr->llink = temp;  
}  
return start;
```

```
NODE delete_front (NODE start)
```

```
{  
    NODE temp;  
    if (start == NULL)  
        printf("\n\n Doubly Linked List is empty:\n\n");  
    else  
    {  
temp = start;  
        start = start->rlink;  
        if (start != NULL)  
        {  
            start->llink = NULL;  
        }  
        printf("\n\n DELETED FRONT NODE IS:.\n\n");  
    }  
}
```

Enter the choice: 6

DELETED FRONT NODE IS:

150|Divya|PRODUCTION|MANAGER|8526545234|25000.000000

MENU

1: CREATE DLL of N EMPLOYEE USING INSERT END

2: DISPLAY and COUNT

3: INSERT END

4: DELETE END

5: INSERT FRONT

6: DELETE FRONT

7: DLL AS DEQUEUE

8: EXIT

Enter the choice: 2

The nodes of the doubly linked list are:

←>|100|Anu|SALES|MANAGER|9658698569|25000.000000

←>|200|Zaina|PRODUCTION|DIRECTOR|7412589698|25000.000

Number of nodes in doubly link list are: 2

MENU

1: CREATE DLL of N EMPLOYEE USING INSERT END

2: DISPLAY and COUNT

3: INSERT END

4: DELETE END

```
printf("\n\n%d|%s|%s|%s|%s|\n\n", temp->ssn, temp->name, temp->dept, temp->dist, temp->phno, temp->sal);  
free(temp);  
}
```

```
return start;
```

```
}  
NODE delete_end(NODE start)
```

```
{  
    NODE temp, p;  
    if (start == NULL)
```

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

```
    }  
    temp = start;  
    if (temp->rlink == NULL)
```

```
        start = NULL;
```

```
    }  
    else  
    {  
        while (temp->rlink != NULL)
```

```
            {  
                temp = temp->rlink;
```

```
            }  
            p = temp->llink;  
            p->rlink = NULL;
```

```
        }  
        printf("DELETED END NODE IS:\n\n");
```

5: INSERT FRONT

6: DELETE FRONT

7: DLL AS DEQUEUE

8: EXIT

Enter the choice:: 7

DEQUEUE INSERTION AND DELETION AT BOTH ENDS

INSERTION AT FRONT OF DEQUEUE

Enter employee data such as SSN, NAME, DEPARTMENT,
DESIGNATION, PHONENO AND SALARY

25

kawitha

SALES

EMPLOYEE

9639639678

20000

The nodes of the doubly linked list are:

←→ | 25 | kawitha | SALES | EMPLOYEE | 9639639678 | 20000.000000 |

←→ | 100 | Ananya | SALES | MANAGER | 9658698567 | 25000.000000 |

←→ | 200 | Zaina | PRODUCTION | DIRECTOR | 7412589678 | 25000.000000 |

Number of nodes in doubly link list are:: 3

INSERTION AT END OF DEQUEUE

Enter employee data such. as SSN, NAME, DEPARTMENT,
DESIGNATION, PHONENO AND SALARY

```
printf("%d %s %s %s %s %s\n", temp->ssn, temp->name,  
temp->dept, temp->desi, temp->phno, temp->sal);  
free(temp);  
return start;  
}  
void display(NODE start)  
{  
    NODE temp;  
    int count=0;  
    if(start==NULL)  
    {  
        printf("\n\nDoubly linked list is empty:\n\n");  
        return;  
    }  
    temp = start;  
    printf("\n\nThe node of the doubly linked list are:\n\n");  
    while(temp!=NULL)  
    {  
        printf("<> %d %s %s %s %s %s\n", temp->ssn, temp->name,  
temp->dept, temp->desi, temp->phno, temp->sal);  
        temp = temp->rlink;  
        count++;  
    }  
    printf("\n\nNumber of node in doubly link list are: %d",  
count);  
}  
void main()  
{  
    int ch, n, i;
```

250

Sandya
PRODUCTION
EMPLOYEE
78945602
20000

MENU

- 1: CREATE DLL of N EMPLOYEE USING INSERT END
- 2: DISPLAY and COUNT
- 3: INSERT END
- 4: DELETE END
- 5: INSERT FRONT
- 6: DELETE
- 7: DLL AS DEQUEUE
- 8: EXIT

Enter the choice::7

DELETION AT FRONT OF DEQUEUE

DELETED FRONT NODE IS::

|25| Kawitha | SALES | EMPLOYEE | 9639639678 | 20000.000000 |

The nodes of the doubly linked list are:

←-> |100| Ananya | SALES | MANAGER | 9658698567 | 25000.000000 |

←-> |200| Zaina | PRODUCTION | EMPLOYEE | 7412589678 | 25000.000000 |


```
for(;;)
```

```
{  
printf("\n\nMENU\n\n");
```

```
printf("\n1: CREATE DLL of N EMPLOYEE USING INSERT END\n  
2: DISPLAY and COUNT\n3: INSERT END\n4: DELETE  
END\n5: INSERT FRONT\n6: DELETE FRONT\n7: DLL AS  
EQUEUE\n8: EXIT\n\n");
```

```
printf("\n Enter your choice:");
```

```
scanf("%d", &ch);
```

```
switch(ch){
```

```
{  
case 1: printf("\n\n Enter how many employee information  
you want to enter:");
```

```
scanf("%d", &n);
```

```
for(i=1; i<=n; i++)
```

```
{  
printf("\n\n Enter information of employee %d\n", i);  
start = insert_end(start);
```

```
}
```

```
break;
```

```
case 2: display(start);
```

```
break;
```

```
case 3: start = insert_end(start);
```

```
break;
```

```
case 4: start = delete_end(start);
```

```
break;
```

```
case 5: start = insert_front(start);
```

```
break;
```

```
case 6: start = delete_front(start); break;
```

←→ |250| Sandya | PRODUCTION | EMPLOYEE

Number of nodes in doubly link list are:: 3

DELETION AT END OF DEQUEUE

DELETED END NODE IS::

|250| Sandya | PRODUCTION | EMPLOYEE | 7894560934 | 20000.000000

The nodes of the doubly linked list are:

←→ |100| Ananya | SALES | MANAGER | 9658698569 | 25000.000000

←→ |200| Zaina | PRODUCTION | DIRECTOR | 74125896 | 25000.000000

Number of nodes in doubly link list are:: 2

MENU

- 1: CREATE DLL of N EMPLOYEE USING INSERT END
- 2: DISPLAY and COUNT
- 3: INSERT END
- 4: DELETE END
- 5: INSERT FRONT
- 6: DELETE FRONT
- 7: DLL AS DEQUEUE
- 8: EXIT

Enter the choice:: 8

```
case 7: printf("\n\nDEQUEUE INSERTION AND DELETION AT  
BOTH ENDS\n\n");  
printf("\n\nINSERTION AT FRONT OF DEQUEUE\n\n");  
start = insert_front(start);  
display(start);  
printf("\n\nINSERTION AT END OF DEQUEUE\n\n");  
start = insert_end(start);  
printf("\n\nDELETION AT FRONT OF DEQUEUE\n\n");  
start = delete_front(start);  
display(start);  
printf("\n\nDELETION AT END OF DEQUEUE\n\n");  
start = delete_end(start);  
display(start);  
break;
```

```
default: exit(0);  
}
```

Observation	-	9	
Record	-	10	
time	-	10	
total	-	29	
sign	-	m2	4/10/19

OUTPUT

1. Represent and Evaluate
2. Add Two poly
3. Exit

Enter Your Choice: 1

Enter the polynomial

Enter the coefficient and stop polynomial reading enter -9

Enter the 1 term

coeff = 6

pow x = 2

pow y = 2

pow z = 1

Enter the 2 term

coeff = -4

pow x = 0

pow y = 1

pow z = 5

Enter the 3 term

coeff = 3

pow x = 3

pow y = 1

pow z = 5

Enter the 4 term

Design, Develop and Implement a Program in C for the following operations on Singly circular linked list (SCLL) with header nodes

a) Represent and Evaluate a Polynomial $P(x, y, z) = 6x^3y^2z - 4yz^5 + 3x^2yz + 2xy^5z - 2xyz^3$

b) Find the sum of 2 polynomials $POLY1(x, y, z)$ and $POLY2(x, y, z)$ and store the result in $POLYSUM(x, y, z)$
Support the program with appropriate functions for each of the above operations.

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

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

```
#include <math.h>
```

```
struct node
```

```
{
```

```
int cf, px, py, pz, flag;
```

```
struct node *link;
```

```
};
```

```
typedef struct node *NODE;
```

```
NODE create_node()
```

```
{
```

```
NODE ptr;
```

```
ptr = (NODE) malloc(sizeof(struct node));
```

```
if (ptr == NULL)
```

```
{
```

```
printf("Insufficient memory\n");
```

```
exit(0);
```

```
}
```

```
return ptr;
```

```
}
```

coeff = 2

pow x = 1

pow y = 5

pow z = 1

Enter the 5 term

coeff = -2

pow x = 1

pow y = 1

pow z = 3

Enter the 6 term

coeff = -999

Polynomial is

$$+6x^1 2y^1 2z^1 - 4x^1 0y^1 1z^1 5 + 3x^1 3y^1 1z^1 + 2x^1 1y^1 5z^1 + 2x^1 1y^1$$

Enter the values of variables x, y and z

2 2 3

The total sum is: -1314

1. Represent and Evaluate
2. Add Two poly
3. Exit

Enter Your Choice: 2

Enter the first polynomial

Enter the coefficient and stop polynomial reading enter = 999

Enter the 1 term

coeff = 4

```
NODE insert_end (int x, int y, int z, NODE head)
```

```
{  
    NODE temp, ptr;  
    ptr = create_node();  
    ptr -> x = x;  
    ptr -> y = y;  
    ptr -> z = z;  
    ptr -> flag = 0;  
    if (head -> link == head)
```

```
        head -> link = ptr;  
        ptr -> link = head;
```

```
    }  
    else
```

```
{  
    temp = head -> link;  
    while (temp -> link != head)  
        temp = temp -> link;  
    temp -> link = ptr;  
    ptr -> link = ptr;  
}
```

```
return head;
```

```
void display (NODE head)
```

```
{  
    NODE temp;
```

```
if (head -> link == head)
```

pow x = 2

pow y = 3

pow z = 3

Enter the 2 term

coeff = -6

pow x = 2

pow y = 2

pow z = 2

Enter the 3 term

coeff = 4

pow x = 1

pow y = 1

pow z = 1

Enter the 4 term

coeff = 10

pow x = 0

pow y = 0

pow z = 0

Enter the 5 term

coeff = -999

Enter the second polynomial

Enter the coefficient and stop polynomial reading enter -99

Enter the 1 term

coeff = 5

pow x = 0

pow y = 3

pow z = 5

Enter the 2 term


```
printf("\n\n Polynomial empty\n");  
else  
{  
temp = head->link;  
while(temp != head)  
{  
if(temp->cf < 0)  
printf("%dx^%dy^%dz^%d", temp->cf, temp->px,  
temp->py, temp->pz);  
else  
printf(" + %dx^%dy^%dz^%d", temp->cf, temp->px,  
temp->py, temp->pz);  
temp = temp->link;  
}  
}  
}
```

```
NODE add_poly(NODE h1, NODE h2, NODE h3)
```

```
{  
NODE p1, p2;  
int cof;  
p1 = h1->link;  
while(p1 != h1)  
{  
p2 = h2->link;  
while(p2 != h2)  
{  
if((p1->px == p2->px) && p1->py == p2->py && p1->pz &&  
p2->pz)  
break;
```

$$\text{coeff} = 4$$

$$\text{pow } x = 2$$

$$\text{pow } y = 2$$

$$\text{pow } z = 2$$

Enter the 3 term

$$\text{coeff} = 8$$

$$\text{pow } x = 1$$

$$\text{pow } y = 1$$

$$\text{pow } z = 1$$

Enter the 4 term

$$\text{coeff} = -999$$

The first polynomial is

$$+4x^2y^2z^2 - 6x^2y^2z^2 + 4x^1y^1z^1 + 10x^10y^10z^10$$

Second polynomial is

$$+5x^10y^13z^15 + 4x^12y^12z^12 + 8x^11y^11z^11$$

The sum of two polynomial is

$$+4x^12y^13z^13 - 2y^12z^12 + 12x^11y^11z^11 + 10x^10y^10z^10 + 5x^10y^13z^15$$

1. Represent and Evaluate
2. Add Two poly
3. Exit

Enter Your choice: 3

```
p2 = p2 -> link;  
}  
if (p2 != h2)  
{  
    cof = p1 -> cf + p2 -> cf;  
    p2 -> flag = 1;  
    if (cof != 0)  
        h3 = insert_end(cof, p1 -> px, p1 -> py, p1 -> pz, h3);  
}  
else  
    h3 = insert_end(p1 -> cf, p1 -> px, p1 -> py, p1 -> pz, h3);  
p1 = p1 -> link;  
}  
p2 = h2 -> link;  
while (p2 != h2)  
{  
    if (p2 -> flag == 0)  
        h3 = insert_end(p2 -> cf, p2 -> px, p2 -> pz, h3);  
    p2 = p2 -> link;  
}  
return h3;
```

```
NODE read_poly(NODE head)
```

```
{  
    int i, cf, x, y, z;  
    printf("\n Enter the coefficient and exponents to stop  
    polynomial reading enter -999\n");  
    for (i = 1; i <= 10; i++)  
        ;  
}
```

```
printf("\nEnter the %d term\n", i);  
printf("coeff = ");  
scanf("%d", &cf);  
if(cf == -999)  
    break;  
printf("pow x = ");  
scanf("%d", &x);  
printf("pow y = ");  
scanf("%d", &y);  
printf("pow z = ");  
scanf("%d", &z);  
head = insert_end(cf, x, y, z, head);  
}
```

```
return head;
```

```
void polysum()
```

```
Node h1, h2, h3;
```

```
h1 = create_node();
```

```
h2 = create_node();
```

```
h3 = create_node();
```

```
h1->link = h1;
```

```
h2->link = h2;
```

```
h3->link = h3;
```

```
printf("\nEnter the first polynomial\n");
```

```
h1 = read_poly(h1);
```

```
printf("\nEnter the second polynomial\n");
```

```
h2 = read_poly(h2);
```

```
h3 = add_poly(h1, h2, h3);
```

DATE 17-10-19
EXP. NO. 09

EXPT. TITLE :

PROGRAM-09

PAGE 111 (19)

```
printf("\n The first polynomial is\n");  
display(h1);  
printf("\n The second polynomial is\n");  
display(h2);  
printf("\n\n The sum of two polynomial is\n");  
display(h3);  
}
```

```
void eval()
```

```
{  
Node h, temp;
```

```
int x, y, z, sum = 0;
```

```
h = create_node();
```

```
h->link = h;
```

```
printf("\n Enter the polynomial\n");
```

```
h = read_poly(h);
```

```
printf("\n\n Polynominal is\n");
```

```
display(h);
```

```
printf("\n Enter the values of variables x, y and z\n");
```

```
scanf("%d%d%d", &x, &y, &z);
```

```
temp = h->link;
```

```
while(temp != h)
```

```
{
```

```
sum += temp->coef * pow(x, temp->px) * pow(y, temp->py) *  
pow(z, temp->pz);
```

```
temp = temp->link;
```

```
}
```

```
printf("\n The total sum is: %d\n", sum);
```

```
}
```

```
void main()
```

```

int ch;
for(;;)
{
printf("\n 1. Represent and Evaluate\n 2. Add Two poly\n 3.
Exit");
printf("\nEnter Your choice");
scanf("%d", &ch);
switch(ch)
{
case 1: eval();
break;
case 2: polysum();
break;
default: exit(0);
}
}

```

Observation	- 9
Reward	- 10
Live	- 10
total	- 29
sign	- 29/10/19

OUTPUT

OPERATIONS ON BST

1: CREATE BST

2: TRAVERSE BST

3: SEARCH

4: EXIT

Enter your choice: 1

Enter data: 6

OPERATIONS ON BST

1: CREATE BST

2: TRAVERSE BST

3: SEARCH

4: EXIT

Enter your choice: 1

Enter data: 9

OPERATIONS ON BST

1. CREATE BST

2: TRAVERSE BST

3: SEARCH

4: EXIT

Enter your choice: 1

Enter data: 5

OPERATIONS ON BST

1. CREATE BST

DATE 31-11-19
EXP. NO. 10

EXPT. TITLE :

PROGRAM-10

PAGE NO. 11

Design, Develop and Implement a menu driven Program in C for the following operations on Binary Search Tree (BST) of Integers.

- Create a BST of N Integers: 6, 9, 5, 8, 8, 15, 14, 7, 8, 5, 1
- Traverse the BST in Inorder, Preorder, Post Order
- Search the BST for a given element (KEY) and report the appropriate message
- Exit

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

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

```
struct node
```

```
{
```

```
    struct node *lchild;
```

```
    int data;
```

```
    struct node *rchild;
```

```
};
```

```
typedef struct node *NODE;
```

```
NODE root = NULL;
```

```
NODE create_node()
```

```
{
```

```
    NODE ptr;
```

```
    ptr = (NODE) malloc(sizeof(struct node));
```

```
    if (ptr == NULL)
```

```
{
```

```
        printf("Insufficient memory\n");
```

```
        exit(0);
```

```
}
```


2: TRAVERSE BST

3: SEARCH

4: EXIT

Enter your choice::1

Enter data::8

OPERATIONS ON BST

1: CREATE BST

2: TRAVERSE BST

3: SEARCH

4: EXIT

Enter your choice::1

Enter data::15

OPERATIONS ON BST

1: CREATE BST

2: TRAVERSE BST

3: SEARCH

4: EXIT

Enter your choice::1

Enter data::24

OPERATIONS ON BST

1: CREATE BST

2: TRAVERSE BST

3: SEARCH

4: EXIT

```
else  
{  
    printf("Enter data:");  
    scanf("%d", &ptr->data);  
    ptr->lchild = ptr->rchild = NULL;  
}  
return ptr;  
}
```

```
NODE create_bst (NODE root)
```

```
{  
    NODE ptr, temp, p;  
    ptr = create_node();  
    if (root == NULL)  
    {  
        root = ptr;  
        return root;  
    }  
    p = NULL;  
    temp = root;  
    while (temp != NULL)  
    {  
        p = temp;  
        if (ptr->data == temp->data)  
        {  
            printf("Duplicate items are not allowed");  
            free(ptr);  
            return root;  
        }  
        if (ptr->data < temp->data)
```

Enter your choice::1
Enter data::14

OPERATIONS ON BST

- 1: CREATE BST
- 2: TRAVERSE BST
- 3: SEARCH
- 4: EXIT

Enter your choice::1
Enter data::7

OPERATIONS ON BST

- 1: CREATE BST
- 2: TRAVERSE BST
- 3: SEARCH
- 4: EXIT

Enter your choice::1
Enter data::8

Duplicate items are not allowed

OPERATIONS ON BST

- 1: CREATE BST
- 2: TRAVERSE BST
- 3: SEARCH
- 4: EXIT

Enter your choice::1
Enter data::5

Duplicate items are not allowed

```
{
temp = temp->lchild;
}
else
temp = temp->rchild;
}
if (ph->data < p->data)
p->lchild = ph;
else
p->rchild = ph;
return root;
}

void inorder (NODE t)
{
if (t != NULL)
{
inorder (t->lchild);
printf ("%d\t", t->data);
inorder (t->rchild);
}
}

void preorder (NODE t)
{
if (t != NULL)
{
printf ("%d\t", t->data);
preorder (t->lchild);
preorder (t->rchild);
}
}
}
```

OPERATIONS ON BST

- 1: CREATE BST
- 2: TRAVERSE BST
- 3: SEARCH
- 4: EXIT

Enter your choice:: 1

Enter data:: 2

Duplicate items are not allowed

OPERATIONS ON BST

- 1: CREATE BST
- 2: TRAVERSE BST
- 3: SEARCH
- 4: EXIT

Enter your choice:: 2

Preorder traversal: 6 5 2 9 8 7 15 14 24

Inorder traversal: 2 5 6 7 8 9 14 15 24

Postorder traversal: 2 5 7 8 14 24 15 9 6

OPERATIONS ON BST

- 1: CREATE BST
- 2: TRAVERSE BST
- 3: SEARCH
- 4: EXIT

Enter your choice:: 3

Enter key to search in bst:: 24

key found.

DATE 31-11-19
EXP. NO. 10

EXPT. TITLE :

PROGRAM-10

PAGE NO.

64

```
void postorder(NODE t)
```

```
{  
if(t == NULL)
```

```
{  
postorder(t->lchild);
```

```
postorder(t->rchild);
```

```
printf("%d\t", t->data);  
}
```

```
}
```

```
void traverse(NODE root)
```

```
{
```

```
if(root == NULL)
```

```
printf("Empty tree\n");
```

```
else
```

```
{
```

```
printf("\n Preorder traversal: \t");
```

```
preorder(root);
```

```
printf("\n Inorder traversal: \t");
```

```
inorder(root);
```

```
printf("\n Postorder traversal: \t");
```

```
postorder(root);  
}
```

```
}
```

```
void search_bst(NODE root)
```

```
{
```

```
int key, flag = 0;
```

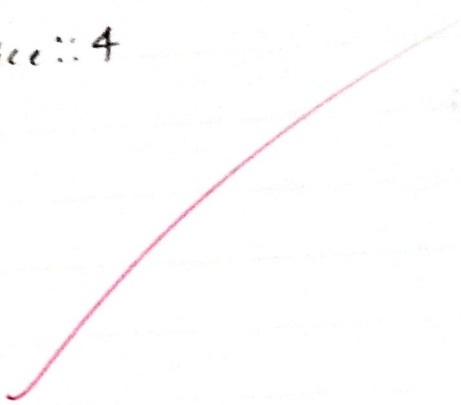
```
NODE temp;
```

```
printf("Enter key to search in bst:");
```

OPERATIONS ON BST

- 1: CREATE BST
- 2: TRAVERSE BST
- 3: SEARCH
- 4: EXIT

Enter your choice:: 4



```
scanf("%d", &key);  
if (root == NULL)  
{  
    printf("BST IS EMPTY\n");  
    return;  
}  
else  
{  
    temp = root;  
    while (temp != NULL)  
    {  
        if (key == temp->data)  
        {  
            printf("key found\n\n");  
            flag = 1;  
            break;  
        }  
        else if (key < temp->data)  
            temp = temp->lchild;  
        else  
            temp = temp->rchild;  
    }  
}  
  
if (flag == 0)  
    printf("key not found\n");  
  
void main()  
{  
    int ch;
```



```
for(i;)\n{\nprintf("\\n OPERATIONS ON BST \\n");\nprintf("\\n 1: CREATE BST \\n 2: TRAVERSE BST \\n 3: SEARCH \\n\n4: EXIT \\n");\nprintf("Enter your choice: ");\nscanf("%d", &ch);\nswitch(ch)\n{\n    case 1: root = create_bst(root);\n            break;\n    case 2: traverse(root);\n            break;\n    case 3: search_bst(root);\n            break;\n    default: exit(0);\n}\n}\n}
```

Observation	-	9
Revised	-	10
mark	-	10
total	-	29
sign	-	ms 4/11/19

OUTPUT

1) Enter number of vertices:: 4

Enter the graph as adjacency matrix

0 1 1 0

0 0 1 1

0 0 0 1

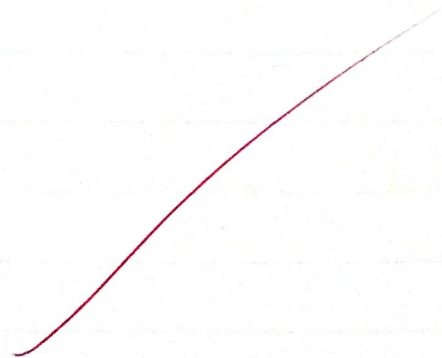
0 0 0 0

Enter source vertex:: 0

BFS OF GIVEN GRAPH

The nodes visited from 0 are::

0 1 2 3



Design, Develop and Implement a Program in C for the following operations on Graph(G) of Cities.

- create a Graph of N cities using Adjacency matrix
- Print all the nodes reachable from a given starting node in a digraph using DFS/BFS method.

11 a] BFS PROGRAM

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

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

```
void bfs (int a[10][10], int n, int u)
```

```
{
```

```
int front, rear, q[10], visited[10], v, i;
```

```
for(i=0; i<n; i++)
```

```
visited[i]=0;
```

```
front=0;
```

```
rear=-1;
```

```
printf("\n The nodes visited from %d are: \n", u);
```

```
q[++rear]=u;
```

```
visited[u]=1;
```

```
printf("%d \t", u);
```

```
while (front <= rear)
```

```
{
```

```
u=q[front++];
```

```
for(v=0; v<n; v++)
```

```
{
```

```
if (a[u][v] == 1 && visited[v] == 0)
```

```
{
```

```
printf("%d \t", v);
```

2) Enter number of vertices :: 5

Enter the graph as adjacency matrix

0 1 1 0 0

0 0 1 1 0

0 0 0 1 0

0 0 0 0 1

0 0 0 0 0

Enter source vertex :: 1

BFS OF GIVEN GRAPH

The nodes visited from 1 are ::

1 2 3 4

```
visited[v]=1;  
q[++rear]=v;
```

```
}
```

```
}
```

```
}
```

```
}
```

```
void main()
```

```
{
```

```
int a[10][10], n, source, i, j;
```

```
printf("\n Enter number of vertices :");
```

```
scanf("%d", &n);
```

```
printf("\n Enter the graph as adjacency matrix\n");
```

```
for(i=0; i<n; i++)
```

```
{
```

```
for(j=0; j<n; j++)
```

```
scanf("%d", &a[i][j]);
```

```
}
```

```
printf("Enter source vertex:");
```

```
scanf("%d", &source);
```

```
printf("\n BFS OF GIVEN GRAPH\n");
```

```
bfs(a, n, source);
```

```
}
```

OUTPUT

1)

Enter number of vertices :: 4

Enter graph as adjacency matrix form

0 1 1 0

0 0 1 1

0 0 0 1

0 0 0 0

Enter source vertex :: 0

DFS of given graph

Node visited from source node 0 are ::

0 1 2 3

11 b) DFS PROGRAM

```
#include <stdio.h>
int a[10][10], visited[10], n;
void dfs(int v)
{
    int v;
    visited[v] = 1;
    printf("%d |t", v);
    for (v = 0; v < n; v++)
    {
        if (a[v][v] == 1 && visited[v] == 0)
            dfs(v);
    }
}

void main()
{
    int source, i, j;
    printf("Enter number of vertices:");
    scanf("%d", &n);
    printf("Enter graph as adjacency matrix formula\n\n");
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
            scanf("%d", &a[i][j]);
    }
    for (i = 0; i < n; i++)
        visited[i] = 0;
}
```

9)

Enter number of vertices::5

Enter graph as adjacency matrix form

0 1 1 0 0

0 0 1 1 1

0 0 0 1 0

0 0 0 0 1

0 0 0 0 0

Enter source vertex::0

DFS of given graph

Node visited from source node 0 are::

0 1 2 3 4


```
printf("\n DFS of given graph\n");  
printf("Node visited from source node %d are :\n", source);  
dfs(source);  
}
```

Observation	- 9
Reward	- 10
wine	- 19
total	- 29
sign	- 29 25/11/19

OUTPUT:

Enter no of records to read from file : 10

Record: 0 is mapped to address : 12

Record: 1 is mapped to address : 45

Record: 2 is mapped to address : 4

Record: 3 is mapped to address : 32

Collision occurred for record 4 resolved using linear probing

Record: 4 is at address : 33

Record: 5 is mapped to address : 21

Record: 6 is mapped to address : 1

Collision occurred for record 7 resolved using linear probing

Record: 7 is at address : 2

Collision occurred for record 8 resolved using linear probing

Record: 8 is at address : 46

Collision occurred for record 9 resolved using linear probing

Record: 9 is at address : 13

The HASH Table content is:

Address	Empno	Name
---------	-------	------

####

1	1801	karun
---	------	-------

2	2101	Diya
---	------	------

####

4	1504	Rahul
---	------	-------

####

####

####

Given a file of N employee records with a set K of keys (4-digit) which uniquely determine the records in file F . Assume that file F is maintained in memory by a Hash Table (HT) of m memory locations with L as the set of memory addresses (2-digit) of locations in HT. Let the keys in K and addresses in L are Integers. Design and develop a Program in C that uses Hash function $H: K \rightarrow L$ as $H(k) = k \text{ mod } m$ (remainder method), and implement hashing technique to map a given key k to the address space L . Resolve the collision (if any) using linear probing.

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

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

```
#include <string.h>
```

```
struct record
```

```
{
```

```
int empno, flag;
```

```
char name[10];
```

```
} emp[100];
```

```
int hash(int m)
```

```
{
```

```
int r;
```

```
r = m % 100;
```

```
return r;
```

```
}
```

```
void main()
```

```
{
```

```
int m, k, emp, loc, i, n, j;
```

#####

#####

#####

#####

12 1012 Riya

13 3913 Harsha

#####

#####

#####

#####

#####

#####

#####

21 1721 Anu

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

32 1232 Rowi

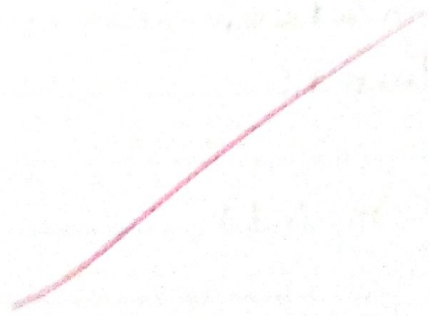
33 1433 Neha

#####

#####

#####

#####



```
char name[10];  
FILE *in;  
printf("\n Enter no. of records to read from file :");  
scanf("%d", &n);  
in = fopen("input.txt", "r");  
if (n <= 10)  
{  
    for (k = 0; k < 100; k++)  
        emp[k].flag = 0;  
    for (i = 0; i < n; i++)  
{  
        scanf(in, "%d %s", &eno, name);  
        loc = hash(eno);  
        if (emp[loc].flag == 0)  
        {  
            printf("\n Record : %d is mapped to address : %d\n",  
                i, loc);  
            emp[loc].empno = eno;  
            emp[loc].flag = 1;  
            strcpy(emp[loc].name, name);  
        }  
        else  
        {  
            printf("\n\n Collision occurred for record %d resolved  
                using linear probing\n", i);  
            for (j = loc + 1; j < 100; j++)  
            {  
                if (emp[j].flag == 0)
```

#####

#####

#####

#####

#####

#####

#####

45 1645 Anil

46 2545 Manya

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

```
printf("\n Record: %d is at address: %d\n",  
        i, j);  
strcpy(emp[j].name, name);  
emp[j].empno = eno;  
emp[j].flag = 1;  
break;  
}
```

```
}  
if(j >= 100)
```

```
{  
    printf("HASH TABLE IS FULL\n");  
    printf("\n-----\n");  
}
```

```
}  
fclose(in);
```

```
printf("\n The Hash Table content is:");  
for(i=0; i<100; i++)
```

```
{  
    if(emp[i].flag == 1)  
        printf("\n %d\t %d\t %s", emp[i].empno, emp[i].name);
```

```
else
```

```
printf("\n #####");  
}
```

```
}  
else
```

```
{  
    printf("\n file is containing only 10 records\n\n");  
}
```


DATE .07-11-19
EXP. NO. 12

EXPT. TITLE :

PROGRAM - 12

PAGE NO.

64

Observation	-	9
Reward	-	10
uiua	-	10
total	-	29
sign	-	$\frac{28}{25/11/19}$

3) Design, Develop and Implement a menu driven Program in C for the following operations on STACK of Integers (Array Implementation of Stack with maximum size MAX)

- a) Push an Element on to Stack
- b) Pop an Element from Stack
- c) Demonstrate how stack can be used to check Palindrome
- d) Demonstrate Overflow and Underflow situations on stack
- e) Display the status of Stack
- f) Exit

Support the program with appropriate functions for each of the above operations.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define MAX 5
int stack[MAX], top = -1, top2 = -1;
int n;
void main()
{
    int ch;
    printf("\n Array Implementation of stack \n");
    printf("\n");
    printf("\n MENU \n");
    printf("\n 1: PUSH \n 2: POP \n 3: TO CHECK PALINDROME \n");
    printf("\n 4: Overflow - Underflow \n 5: DISPLAY \n 6: EXIT \n");
    printf("\n Enter your choice \n");
    scanf("%d", &ch);
    switch(ch)
    {
        case 1: push();
                break;
```

```

case 2: pop();
        break;
case 3: palindrome();
        break;
case 4: Overflow-Underflow();
        break;
case 5: Display()
        break;
case 6: Exit(0);
default: exit(0);
}
}
}

```

```

void push()
{

```

```

    int ele, pos;

```

```

    if ( n == MAX top == MAX - 1)
    {

```

```

        printf(" ARRAY IS FULL, INSERTION IS NOT POSSIBLE\n");
        printf(" STACK");
        exit(0);
    }

```

```

    else
    {

```

```

        printf("Enter the position less than or equal to n+1\n");

```

```

        scanf("%d", &pos);

```

```

if (pos <= n+1)

```

```

        printf("\n Enter the element to be inserted\n");

```

```

        scanf("%d", &ele);

```

```

        for (i = 0; i >= pos - 1; i--)
        {

```

```

            a[i+1] = a[i];
        }

```

```

        a[pos-1] = ele;

```

```

        ++n;
    }
}

```

```

        stack[++top] = ele;

```

```
void pop()
```

```
{  
    int i, item, pos, num;  
    if (n == 0) top = -1;  
    {  
        printf("STACK  
ARRAY IS EMPTY, DELETION IS NOT POSSIBLE\n");  
        exit(0);  
    }  
    else  
    {  
        printf("Enter the position less than or equal to n  
to delete an element\n");  
        scanf("%d", &pos);  
        if (pos <= n)  
        {  
            num  
            item = a[pos-1]; stack[top--];  
            printf("Deleted element is %d\n", item);  
            num  
            --n;  
            for (i = pos; i < n; i++)  
                a[i] = a[i+1];  
        }  
    }  
}
```

```
void Overflow_Underflow()
```

```
{  
    if (top == MAX)  
        printf("Stack Overflow");  
    if (top == -1)  
        printf("Stack Underflow");  
}
```

```
void display()
```

```
{  
    int i;
```



KSIT
K.S. INSTITUTE OF TECHNOLOGY

Kammavari Sangham (R)-1952

K.S. GROUP OF INSTITUTIONS

K.S. INSTITUTE OF TECHNOLOGY

(Approved by AICTE & Affiliated to VTU)

KANAKAPURA ROAD, BANGALORE - 560 062

USN No.:

1 K S I 8 C S O S 2

```
if (top == -1)
{
    printf("Stack is EMPTY");
    exit(0);
}
else
{
    printf("The elements of the stack are : \n");
    for (i = top; i >= 0; i--)
    {
        printf("%d\t", a[i]);
    }
}
}

void palindrome()
{
    int i = 0; j = 0;
    char int s[10], st[10], rev[10];
    printf("Enter the string");
    gets(s); scanf("%s", s);
    while (s[i] != '\0')
    {
        st[++top] = s[i];
        i++;
    }
    if (top != '\0')
    {
        rev[j] = st[top--];
        j++;
    }
}
```

```
rev[j] = '\0';  
if (strcmp(s, rev) == 0)  
    printf("String is palindrome\n");  
else  
    printf("String is not palindrome");  
}
```

Output

Array implementation of stack

MENU

1. PUSH
2. POP
3. TO CHECK PALINDROME
4. OVERFLOW AND UNDERFLOW
5. DISPLAY
6. EXIT

Enter your choice:

1

Enter the element to be inserted

1

MENU

1. PUSH
2. POP
3. TO CHECK PALINDROME
4. OVERFLOW AND UNDERFLOW
5. DISPLAY
6. EXIT

Enter your choice:

1

Enter the element to be inserted

2

O/P ✓
seed

25/11/2019

MENU

1. PUSH
2. POP
3. TO CHECK PALINDROME
4. OVERFLOW AND UNDERFLOW
5. DISPLAY
6. EXIT

Enter your choice :

5

The elements of the stack are:

2 1

MENU

1. PUSH
2. POP
3. TO CHECK PALINDROME
4. OVERFLOW AND UNDERFLOW
5. DISPLAY
6. EXIT

Enter your choice:

2

Deleted element is ~~2~~

MENU

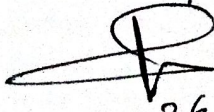
1. PUSH
2. POP
3. TO CHECK PALINDROME
4. OVERFLOW AND UNDERFLOW
5. DISPLAY
6. EXIT

K.S.INSTITUTE OF TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE & ENGG
SESSION : AUG 2019 TO NOV 2019
III A SEC DSC LAB INTERNALS MARKS DETAILS

SL. NO	USN	NAME OF THE STUDENT	OBSERVATION (10M)	RECORD (10M)	VIVA (10M)	TEST (10M)	TOTAL (40M)	SIGNATURE
1	1KS18CS001	ADARSH K	10	10	10	02	32	<i>[Signature]</i>
2	1KS18CS002	AMARAVATHI M	10	10	10	06	36	<i>[Signature]</i>
3	1KS18CS003	ANIKETH H	10	10	10	05	35	<i>[Signature]</i>
5	1KS18CS005	ANUSHRUTI SINGH	10	10	9	05	34	<i>[Signature]</i>
6	1KS18CS006	ARUNA P	10	10	10	08	38	<i>[Signature]</i>
7	1KS18CS007	ASHVINI J	10	10	10	05	35	<i>[Signature]</i>
8	1KS18CS008	AVINASH PRASAD	10	10	10	10	40	<i>[Signature]</i>
9	1KS18CS009	B DEVA DEEKSHITH	10	10	10	03	33	<i>[Signature]</i>
10	1KS18CS010	BHAGWAT GOUTAM	10	10	07	03	30	<i>[Signature]</i>
11	1KS18CS012	BHOOMIKA H	10	10	10	05	35	<i>[Signature]</i>
12	1KS18CS013	BHUVANA CHANDRIKA GANTI	10	10	10	04	34	<i>[Signature]</i>
13	1KS18CS014	BRIJESH S	10	10	09	03	32	<i>[Signature]</i>
14	1KS18CS015	CHAITHRA R	10	10	10	10	40	<i>[Signature]</i>
15	1KS18CS016	CHANDAN KUMAR	10	10	10	10	40	<i>[Signature]</i>
16	1KS18CS017	DANJU NIHARIKA	10	10	10	03	33	<i>[Signature]</i>
17	1KS18CS018	DHANANJAYA S	10	10	10	8	38	<i>[Signature]</i>
18	1KS18CS019	DHRUV JYOTI SHUKLA	10	10	09	8	37	<i>[Signature]</i>
19	1KS18CS020	FARIYA N	10	10	10	04	34	<i>[Signature]</i>
20	1KS18CS021	G.SARAYU CHOWDARY						
21	1KS18CS022	GAGANSURI M S	10	10	10	03	33	<i>[Signature]</i>
22	1KS18CS023	GANESH A	10	10	9	07	36	<i>[Signature]</i>
23	1KS18CS024	GOUTHAM M	10	10	10	10	40	<i>[Signature]</i>
24	1KS18CS025	GUNAL BINANI	10	10	07	07	34	<i>[Signature]</i>
25	1KS18CS026	HARSHITH C PRASAD	09	09	04	04	26	<i>[Signature]</i>
26	1KS18CS027	KANDIMALLA KRISHNA PAVITHRA	10	10	10	07	37	<i>[Signature]</i>
27	1KS18CS028	KARTHIK K	10	10	10	06	36	<i>[Signature]</i>
28	1KS18CS029	KAVITA CHAUDHARY	10	10	10	10	40	<i>[Signature]</i>
29	1KS18CS030	KENCHAM ARUN	10	10	09	02	31	<i>[Signature]</i>
30	1KS18CS031	KILARI ASHWIK	10	10	01	04	34	<i>[Signature]</i>
31	1KS18CS032	KILARI JASWANTH	10	10	10	07	37	<i>[Signature]</i>
32	1KS18CS033	KIRAN VEERANNA DAMBAL	10	10	08	08	36	<i>[Signature]</i>
33	1KS18CS034	KRITHIKA.K.N	10	10	10	03	33	<i>[Signature]</i>
34	1KS18CS035	KRUTHIKA.S.VASISHT	10	10	10	03	33	<i>[Signature]</i>
35	1KS18CS036	LEKKALA SHARANDEEP CHOWDARY	10	10	04	05	29	<i>[Signature]</i>

SL. NO	USN	NAME OF THE STUDENT	OBSERVATION (10M)	RECORD (10M)	VIVA (10M)	TEST (10M)	TOTAL (40M)	SIGNATURE
36	1KS18CS037	LATHA V	10	10	10	07	37	Latha V.
37	1KS18CS038	LAVANYA.C.R	10	10	10	10	40	Lavanya
38	1KS18CS039	LIKHITHA.N	10	10	10	05	35	Likhitha
39	1KS18CS040	LOKESH R	10	10	8	04	32	Lokesh
40	1KS18CS041	MADDULA JITENDRA	10	10	09	02	31	Maddula
41	1KS18CS042	MADHUSUDHAN.S.R	10	10	10	04	34	Madhusudhan
42	1KS18CS043	MAHARAJ S	10	10	05	03	28	Maharaj
43	1KS18CS044	MAHESH B V	10	10	04	04	28	Mahesh B.V.
44	1KS18CS045	MANIKONDA THARUN	10	10	09	06	35	Manikonda
45	1KS18CS046	MANVITH P	10	10	02	03	25	Manvith
46	1KS18CS047	MD SUJAN	10	10	10	10	40	MD Sujana
47	1KS18CS048	MEGHANA.J						
48	1KS18CS049	MEGHASHREE A	10	10	10	06	36	Meghashree A.
49	1KS18CS050	MIKIN K M	10	10	09	02	31	Mikin K.M.
50	1KS18CS051	MONICA S	10	10	10	09	39	Monica S.
51	1KS18CS052	MONIKA.K.C	10	10	09	09	38	Monika
52	1KS18CS053	MOPURI SREE!AKSHMI	10	10	09	5	34	Mopuri
53	1KS18CS054	N SAI JAHANAVI	10	10	06	04	30	N Sai Jahanavi
54	1KS18CS055	NAGARJUN N	10	10	07	02	29	Nagarjun N
55	1KS18CS056	MANDINI J K	10	10	10	09	39	Mandini J.K.
56	1KS18CS057	NARASIMHA MAIYA G S	10	10	10	10	40	Narasimha Maiga G.S.
57	1KS18CS058	NARASIMHARAJU R	10	10	10	02	32	Narasimha R.
58	1KS18CS059	NIKHIL.M	10	10	09	05	34	Nikhil M
59	1KS18CS060	NIKHIL VASAN	10	10	09	08	37	Nikhil Vasan
60	1KS18CS061	NIKIL B S	10	10	09	02	31	Nikil B.S.
61	1KS18CS062	NIKHITHA M	10	10	10	10	40	Nikhitha M
62	DIP	GOLLA YASWANTH	10	10	05	05	30	Golla Yaswanth
63	DIP	CHAITRA.K	10	10	10	10	40	Chaitra.K
64	DIP	B.K.SUSMITHA	10	10	10	05	35	Susmitha
65	DIP	AKSHAY.B.R	10	10	06	04	30	Akshay
66	DIP	RANJITHA.H.D	10	10	10	10	40	Ranjitha
67	DIP	BHAVANI.K G	10	10	10	03	33	Bhavani
68	DIP	KARTHIK PRAKASH	10	10	10	10	40	Karthik
69	DIP	RUSHI.C.S	10	10	05	02	27	Rushi
70	DIP	KALPITHA.A.J	10	10	10	03	33	Kalpitha
71	DIP	DHANALAKSHMI.B	10	10	10	05	35	Dhanalakshmi
72	DIP	RAMYIA						
73	CB	RAMYA	10	10	07	02	29	Ramyia.R

Faculty In-charge


26/11/2019