**List of Programs**

| **SL** | **Program Description** | **Page** |
| --- | --- | --- |
| **No** |  |  |
| **1** | Design, Develop and Implement a menu driven Program in C for the | **6** |
|  | following Array operations |  |
|  | a. Creating an Array of N Integer Elements |  |
|  | b. Display of Array Elements with Suitable Headings |  |
|  | c. Inserting an Element (ELEM) at a given valid Position (POS) |  |
|  | d. Deleting an Element at a given valid Position(POS) |  |
|  | e. Exit. |  |
|  | Support the program with functions for each of the above operations |  |
| **2** | Design, Develop and Implement a Program in C for the following operations | **10** |
|  | 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 all occurrences of |  |
|  | PAT in STR with REP if PAT exists in STR. Report 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. |  |
| **3** | Design, Develop and Implement a menu driven Program in C for the | **12** |
|  | 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 |  |
| **4** | Design, Develop and Implement a Program in C for converting an Infix | **17** |
|  | Expression to Postfix Expression. Program should support for both |  |
|  | parenthesized and free parenthesized expressions with the operators: +, -, \*, |  |
|  | /, %(Remainder), ^(Power) and alphanumeric operands. |  |
|  | |  |
|  | |  |
| BMSIT&M, Bangalore, Dept of CSE | | Page 3 |

| **5** Design, Develop and Implement a Program in C for the following Stack | **19** |
| --- | --- |
| Applications |  |
| a. Evaluation of Suffix expression with single digit operands and operators: |  |
| +, -, \*, /, %, ^ |  |
| b. Solving Tower of Hanoi problem with n disks |  |
| **6** Design, Develop and Implement a menu driven Program in C for the | **22** |
| following operations on Circular QUEUE of Characters (Array |  |
| Implementation of Queue with maximum size MAX) |  |
| a. Insert an Element on to 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 |  |
| **7** Design, Develop and Implement a menu driven Program in C for the | **26** |
| following operations on Singly Linked List (SLL) of Student Data with the |  |
| fields: *USN, Name, Branch, Sem, PhNo* |  |
| a. Create a SLL of N Students Data by using *front insertion*. |  |
| b. Display the status of SLL and count the number of nodes in it |  |
| c. Perform Insertion and Deletion at End of SLL |  |
| d. Perform Insertion and Deletion at Front of SLL |  |
| e. Demonstrate how this SLL can be used as STACK and QUEUE |  |
| f. Exit |  |
| **8** Design, Develop and Implement a menu driven Program in C for the | **32** |
| following operations on Doubly Linked List (DLL) of Employee Data with |  |
| the fields: *SSN, Name, Dept, Designation, Sal, PhNo* |  |

a. Create a DLL of N Employees Data by using *end* *insertion*. b. Display the status of DLL and count thenumber of nodes in it c. Perform Insertion and Deletion at End of DLL

d. Perform Insertion and Deletion at Front of DLL

e. Demonstrate how this DLL can be used as Double Ended Queue f. Exit



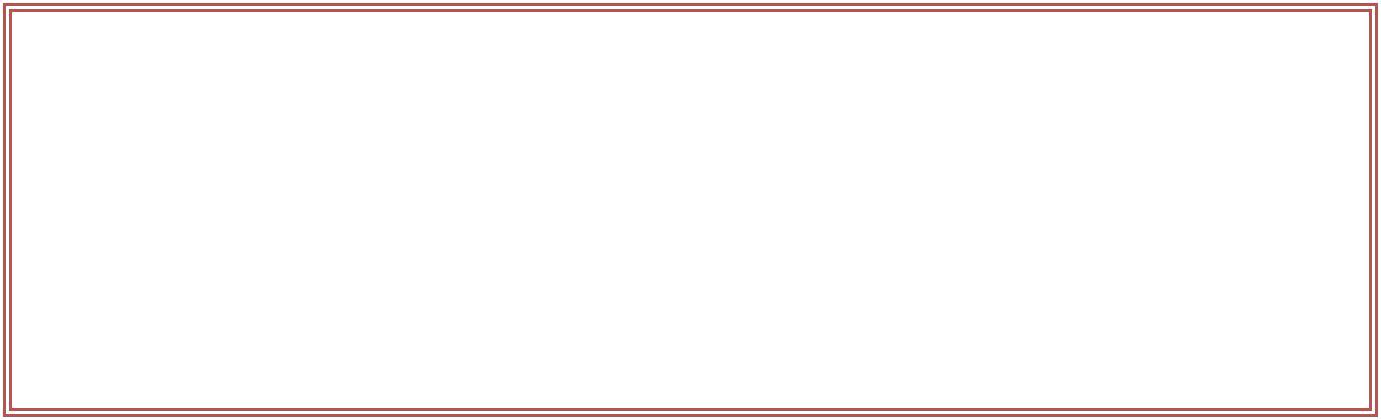
BMSIT&M, Bangalore, Dept of CSE Page 4

| **9** | Design, Develop and Implement a Program in C for the following operations | **38** |
| --- | --- | --- |
|  | on Singly Circular Linked List (SCLL) with header nodes |  |
|  | a. Represent and Evaluate a Polynomial P(x,y,z) = 6x2y2z- |  |
|  | 4yz5+3x3yz+2xy5z-2xyz3 |  |
|  | b. Find the sum of two 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 |  |
| **10** | Design, Develop and Implement a menu driven Program in C for the | **44** |
|  | following operations on Binary Search Tree (BST) of Integers |  |
|  | a. Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2 |  |
|  | b. Traverse the BST in Inorder, Preorder and Post Order |  |
|  | c. Search the BST for a given element (KEY) and report the appropriate |  |
|  | message |  |
|  | d. Delete an element(ELEM) from BST |  |
|  | e. Exit |  |
| **11** | Design, Develop and Implement a Program in C for the following operations | **50** |
|  | on Graph(G) of Cities |  |
|  | a. Create a Graph of N cities using Adjacency Matrix. |  |
|  | b. Print all the nodes reachable from a given starting node in a digraph using |  |
|  | BFS method |  |
|  | c. Check whether a given graph is connected or not using DFS method |  |
| **12** | Given a File of N employee records with a set K of Keys(4-digit) which | **54** |
|  | 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 →L as H(K)=K 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.



BMSIT&M, Bangalore, Dept of CSE Page 5

1. Design, Develop and Implement a menu driven Program in C for the following Array operations
   1. Creating an Array of N Integer Elements
   2. Display of Array Elements with Suitable Headings
   3. Inserting an Element (ELEM) at a given valid Position (POS)
   4. Deleting an Element at a given valid Position(POS)
   5. Exit.

Support the program with functions for each of the above operations

#include<stdio.h>

#include<stdlib.h>

int a[20];

int n,val,i,pos;

/\*Function Prototype\*/ void create();

void display();

void insert();

void delete();

int main()

{

int choice;

while(choice)

{

printf("\n\n--------MENU-----------\n");

printf("1.CREATE\n");

printf("2.DISPLAY\n");

printf("3.INSERT\n");

printf("4.DELETE\n");

printf("5.EXIT\n");

printf("-----------------------");

printf("\nENTER YOUR CHOICE:\t");

scanf("%d",&choice);

switch(choice)

{

case 1: create();

break;

case 2: display();

break;



BMSIT&M, Bangalore, Dept of CSE Page 6

case 3: insert();

break;

case 4: delete();

break;

case 5: exit(0);

break;

default:

printf("\nInvalid choice:\n");

break;

}

}

return 0;

}

//creating an array

void create()

{

printf("\nEnter the size of the array elements:\t");

scanf("%d",&n);

printf("\nEnter the elements for the array:\n");

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

{

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

}

}

//displaying an array elements

void display()

{

int i;

printf("\nThe array elements are:\n");

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

{

printf("%d\t",a[i]);

}

}

//inserting an element into an array

void insert()

{

printf("\nEnter the position for the new element:\t");



BMSIT&M, Bangalore, Dept of CSE Page 7

scanf("%d",&pos);

printf("\nEnter the element to be inserted :\t");

scanf("%d",&val);

for(i=n-1;i>=pos;i--)

{

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

}

a[pos]=val;

n=n+1;

}

//deleting an array element

void delete()

{

printf("\nEnter the position of the element to be deleted:\t"); scanf("%d",&pos);

val=a[pos];

for(i=pos;i<n-1;i++)

{

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

}

n=n-1;

printf("\nThe deleted element is =%d",val);

}

**Output**

inux:~/dslab # gedit array.c

linux:~/dslab # cc array.c

linux:~/dslab # ./a.out

--------MENU-----------

1.CREATE

2.DISPLAY

3.INSERT

4.DELETE

5.EXIT

-----------------------

ENTER YOUR CHOICE: 1

Enter the size of the array elements: 3

Enter the elements for the array:



BMSIT&M, Bangalore, Dept of CSE Page 8

10 25 30

ENTER YOUR CHOICE: 2

The array elements are:

10 25 30

ENTER YOUR CHOICE: 3

Enter the position for the new element: 1

Enter the element to be inserted: 20

ENTER YOUR CHOICE: 2

The array elements are:

10 20 25 30

ENTER YOUR CHOICE: 4

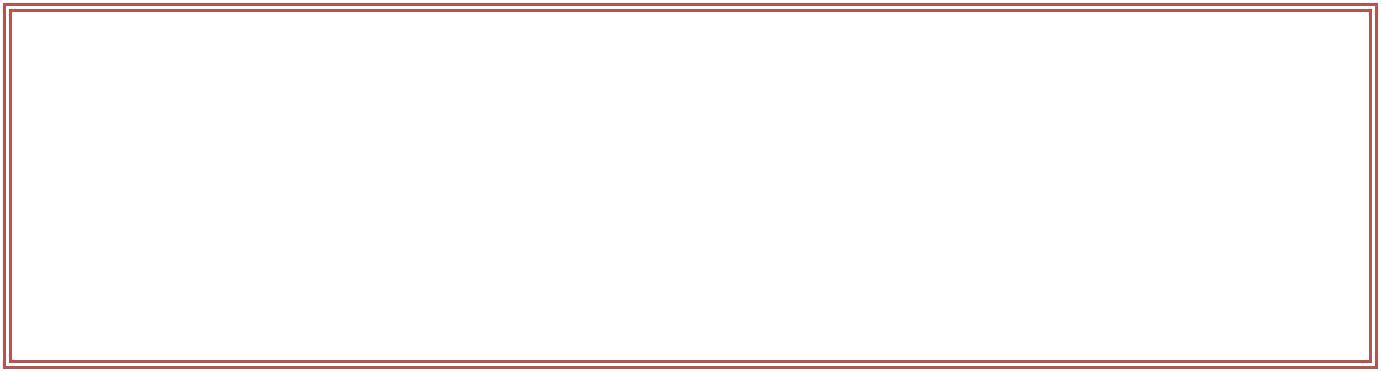
Enter the position of the element to be deleted: 3

The deleted element is =30

enter your choice: 5



BMSIT&M, Bangalore, Dept of CSE Page 9

1. Design, Develop and Implement a Program in C for the following operations on Strings
   1. Read a main String (STR), a Pattern String (PAT) and a Replace String (REP)
   2. Perform Pattern Matching Operation: Find and Replace all occurrences of PAT in STR with REP if PAT exists in STR. Report 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 main()

{

char STR[100],PAT[100],REP[100],ans[100];

int i,j,c,m,k,flag=0;

printf("\nEnter the MAIN string: \n");

gets(STR);

printf("\nEnter a PATTERN string: \n");

gets(PAT);

printf("\nEnter a REPLACE string: \n");

gets(REP);

i = m = c = j = 0;

while ( STR[c] != '\0')

{

* Checking for Match if ( STR[m] == PAT[i] )

{

i++;

m++;

flag=1;

if ( PAT[i] == '\0')

{

//copy replace string in ans string

for(k=0; REP[k] != '\0';k++,j++)

ans[j] = REP[k];

i=0;

c=m;

}

}

else //mismatch

{

ans[j] = STR[c];



BMSIT&M, Bangalore, Dept of CSE Page 10

j++;

c++; m = c; i=0;

}

}

if(flag==0)

{

printf("Pattern doesn't found!!!");

}

else

{

ans[j] = '\0';

printf("\nThe RESULTANT string is:%s\n" ,ans);

}

}

**Output**

linux:~/dslab # gedit string.c linux:~/dslab # cc string.c linux:~/dslab # ./a.out

Enter the MAIN string: good morning

Enter a PATTERN string: morning

Enter a REPLACE string: evening

The RESULTANT string is: good evening

linux:~/dslab # ./a.out Enter the MAIN string: hi vcet

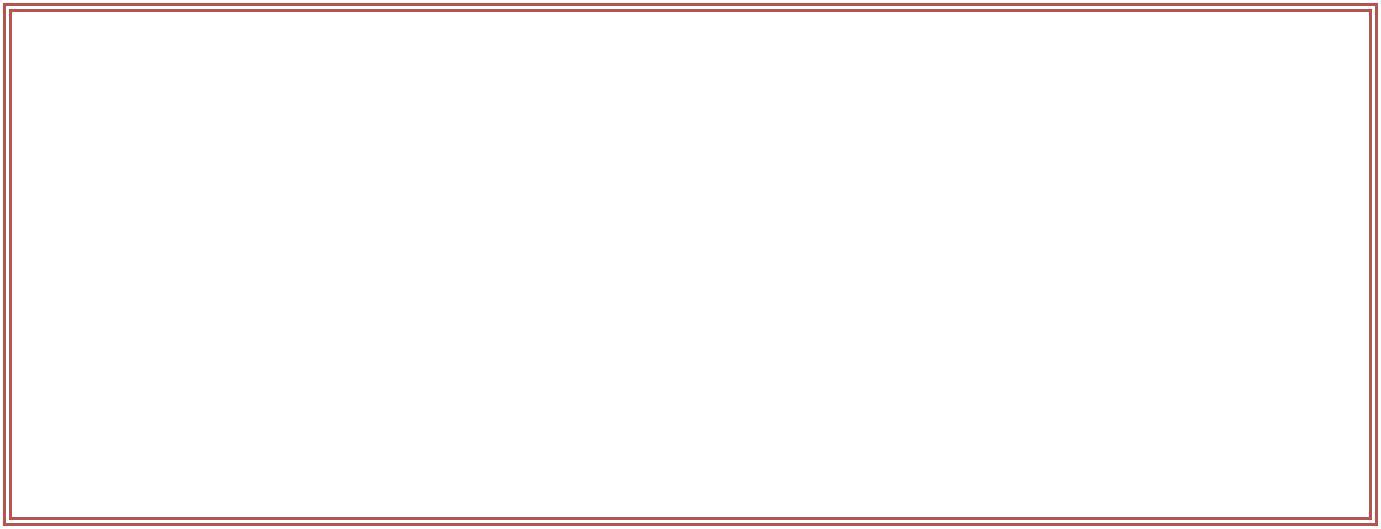
Enter a PATTERN string: bye

Enter a REPLACE string: hello

Pattern doesn't found!!!



BMSIT&M, Bangalore, Dept of CSE Page 11



1. 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)
   1. *Push* an Element on to Stack
   2. *Pop* an Element from Stack
   3. Demonstrate how Stack can be used to check *Palindrome*
   4. Demonstrate *Overflow* and *Underflow* situations on Stack
   5. Display the status of Stack
   6. Exit

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

#include<stdlib.h>

#include<stdio.h>

#include<string.h>

#define max\_size 5

int stack[max\_size],top=-1;

void push();

void pop();

void display();

void pali();

int main()

{

int choice; while(choice)

{

//printf("\n");

printf("\n\n--------STACK OPERATIONS-----------\n"); printf("1.Push\n");

printf("2.Pop\n");

printf("3.Palindrome\n");

printf("4.Display\n");

printf("5.Exit\n");

printf("-----------------------");

printf("\nEnter your choice:\t");

scanf("%d",&choice);

switch(choice)

{

case 1: push();



BMSIT&M, Bangalore, Dept of CSE Page 12

break;

case 2: pop();

break;

case3: palindrome();

break;

case 4: display();

break;

case 5: exit(0);

break;

default: printf("\nInvalid choice:\n");

break;

}

}

return 0;

}

void push() //Inserting element into the stack

{

int item,n;

if(top==(max\_size-1))

{

printf("\nStack Overflow:");

}

else

{

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

scanf("%d",&item);

top=top+1;

stack[top]=item;

}

}

void pop() //deleting an element from the stack

{

int item;

if(top==-1)

{

printf("Stack Underflow:");

}

else

{



BMSIT&M, Bangalore, Dept of CSE Page 13

item=stack[top];

top=top-1;

printf("\nThe poped element: %d\t",item);

}

}

void palindrome()

{

int digit,j,k,len=top+1,flag=0,ind=0;

int num[len],rev[len],i=0;

while(top!=-1)

{

digit= stack[top];

num[i]=digit;

top--;

i++;

}

for(j=0;j<len;j++)

{

printf("Numbers= %d\n",num[j]);

}

printf("reverse operation :\n");

for(k=len-1;k>=0;k--)

{

rev[k]=num[ind];

ind++;

}

printf("reverse array : ");

for(k=0;k<len;k++)

{

printf("%d\n",rev[k]);

}



BMSIT&M, Bangalore, Dept of CSE Page 14

printf("check for palindrome :\n");

int length = 0;

for(i=0;i<len;i++)

{

if(num[i]==rev[i])

{

length = length+1;

}

}

if(length==len)

{

printf("It is palindrome number\n");

}

else

{

printf("It is not a palindrome number\n");

}

top = len-1;

}

void display()

{

int i; if(top==-1)

{

printf("\nStack is Empty:");

}

else

{

printf("\nThe stack elements are:\n" );

for(i=top;i>=0;i--)

{

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

}

}

}

**Output**

linux:~/dslab # gedit stack.c

linux:~/dslab # cc stack.c

linux:~/dslab # ./a.out



BMSIT&M, Bangalore, Dept of CSE Page 15

--------STACK OPERATIONS-----------

1.Push

2.Pop

3.Palindrome

4.Display

5.Exit

-----------------------

Enter your choice: 1

Enter the element to be inserted: 1

Enter your choice: 1

Enter the element to be inserted: 2

Enter your choice: 1

Enter the element to be inserted: 1

Enter your choice: 1

Enter the element to be inserted: 5

Enter your choice: 2

The poped element: 5

Enter your choice: 4

The stack elements are: 1 2 1

Enter your choice: 3

Numbers= 1 Numbers= 2 Numbers= 1

reverse operation :

reverse array :

1

2

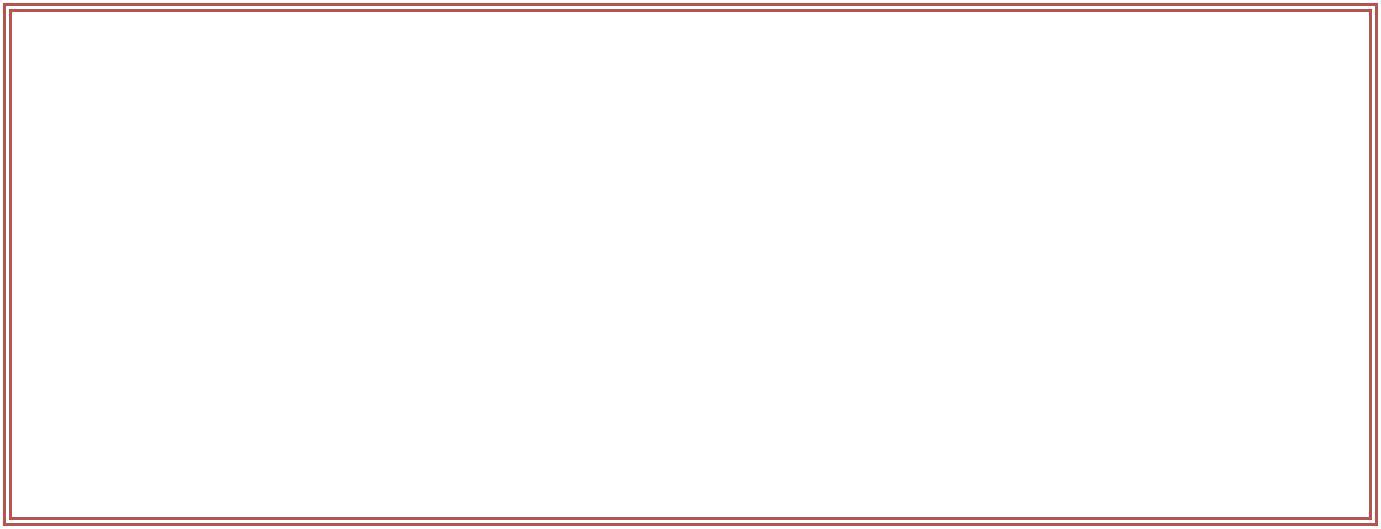
1

check for palindrome :

It is palindrome number



BMSIT&M, Bangalore, Dept of CSE Page 16

1. 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.

#define SIZE 50 /\* Size of Stack \*/

#include <ctype.h>

#include <stdio.h>

char s[SIZE];

int top = -1; /\* Global declarations \*/

push(char elem) /\* Function for PUSH operation \*/

{

s[++top] = elem;

}

char pop() /\* Function for POP operation \*/

{

return (s[top--]);

}

int pr(char elem) /\* Function for precedence \*/

{

switch (elem)

{

case '#': return 0;

case '(': return 1;

case '+':

case '-': return 2;

case '\*':

case '/':

case '%': return 3;

case '^': return 4;

}

}



BMSIT&M, Bangalore, Dept of CSE Page 17

void main() /\* Main Program \*/

{

char infx[50], pofx[50], ch, elem;

int i = 0, k = 0;

printf("\n\nRead the Infix Expression ? ");

scanf("%s", infx);

push('#');

while ((ch = infx[i++]) != '\0')

{

if (ch == '(') push(ch);

else if (isalnum(ch))

pofx[k++] = ch;

else if (ch == ')')

{

while (s[top] != '(')

pofx[k++] = pop();

elem = pop(); /\* Remove ( \*/

}

else /\* Operator \*/

{

while (pr(s[top]) >= pr(ch))

pofx[k++] = pop();

push(ch);

}

}

while (s[top] != '#') /\* Pop from stack till empty\*/

pofx[k++] = pop();

pofx[k] = '\0'; /\* Make pofx as valid string \*/

printf("\n\nGiven Infix Expn: %s Postfix Expn: %s\n", infx, pofx);

}

**Output**

linux:~/dslab # gedit intopost.c

linux:~/dslab # cc intopost.c

linux:~/dslab # ./a.out

Read the Infix Expression ? (a+b)\*c/d^5%1

Given Infix Expn: (a+b)\*c/d^5%1

Postfix Expn: ab+c\*d5^/1%



BMSIT&M, Bangalore, Dept of CSE Page 18

1. Design, Develop and Implement a Program in C for the following Stack Applications
   1. Evaluation of Suffix expression with single digit operands and operators: +, -, \*, /, %, ^
   2. Solving Tower of Hanoi problem with n disks

* Evaluation of Suffix Expression #include<stdio.h> #include<string.h> #include<stdlib.h> #include<math.h>

#define MAX 50

int stack[MAX];

char post[MAX];

int top=-1;

void pushstack(int tmp);

void calculator(char c);

void main()

{

int i;

printf("Insert a postfix notation :: "); //gets(post);

scanf("%s",post);

for(i=0;i<strlen(post);i++)

{

if(post[i]>='0' && post[i]<='9')

{

pushstack(i);

}

if(post[i]=='+' || post[i]=='-' || post[i]=='\*' || post[i]=='/' || post[i]=='^')

{

calculator(post[i]);

}

}

printf("\n\nResult :: %d",stack[top]);

}

void pushstack(int tmp)

{

top++; stack[top]=(int)(post[tmp]-48);

}

void calculator(char c)

{



BMSIT&M, Bangalore, Dept of CSE Page 19

int a,b,ans;

a=stack[top];

stack[top]='\0;

top--;

b=stack[top];

stack[top]='\0';

top--;

switch(c)

{

case '+': ans=b+a;

break;

case '-': ans=b-a;

break;

case '\*': ans=b\*a;

break;

case '/': ans=b/a;

break;

case '^': ans=pow(b,a);

break;

default: ans=0;

}

top++;

stack[top]=ans;

}

// Towers of Hanoi #include <stdio.h>

void towers(int, char, char, char); int main()

{

int num;

printf("Enter the number of disks : ");

scanf("%d", &num);

printf("The sequence of moves involved in the Tower of Hanoi are :\n"); towers(num, 'A', 'C', 'B');

return 0;

}

void towers(int num, char frompeg, char topeg, char auxpeg)

{

if (num == 1)



BMSIT&M, Bangalore, Dept of CSE Page 20

{

printf("\n Move disk 1 from peg %c to peg %c", frompeg, topeg); return;

}

towers(num - 1, frompeg, auxpeg, topeg);

printf("\n Move disk %d from peg %c to peg %c", num, frompeg, topeg); towers(num - 1, auxpeg, topeg, frompeg);

}

**Output**

linux:~/dslab #gedit posteval.c

linux:~/dslab #gcc posteval.c –lm

linux:~/dslab # ./a.out

Insert a postfix notation :: 22^32\*+

Result :: 10

linux:~/dslab #gedit tower.c

linux:~/dslab #gcc tower.c

linux:~/dslab # ./a.out

Enter the number of disks : 3

The sequence of moves involved in the Tower of Hanoi are :

Move disk 1 from peg A to peg C

Move disk 2 from peg A to peg B

Move disk 1 from peg C to peg B

Move disk 3 from peg A to peg C

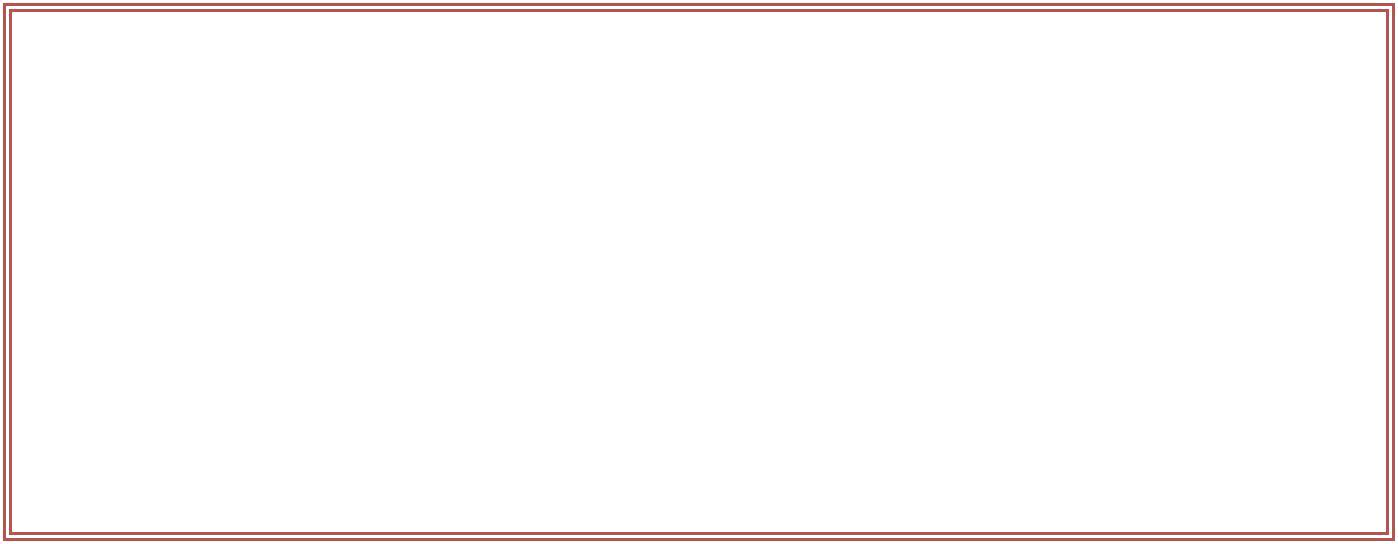
Move disk 1 from peg B to peg A

Move disk 2 from peg B to peg C

Move disk 1 from peg A to peg C



BMSIT&M, Bangalore, Dept of CSE Page 21

1. Design, Develop and Implement a menu driven Program in C for the following operations on

Circular QUEUE of Characters (Array Implementation of Queue with maximum size MAX)

a. Insert an Element on to 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> #define max 10

int q[10],front=0,rear=-1;

void main()

{

int ch;

void insert();

void delet();

void display();

printf("\nCircular Queue operations\n"); printf("1.insert\n2.delete\n3.display\n4.exit\n"); while(1)

{

printf("Enter your choice:");

scanf("%d",&ch);

switch(ch)

{

case 1: insert();

break;

case 2: delet();

break;

case 3: display();

break;

case 4: exit(1);

default: printf("Invalid option\n");

}

}

}



BMSIT&M, Bangalore, Dept of CSE Page 22

void insert()

{

int x;

if((front==0&&rear==max-1)||(front>0&&rear==front-1))

printf("Queue is overflow\n");

else

{

printf("Enter element to be insert:");

scanf("%d",&x);

if(rear==max-1&&front>0)

{

rear=0;

q[rear]=x;

}

else

{

if((front==0&&rear==-1)||(rear!=front-1))

q[++rear]=x;

}

}

}

void delet()

{

int a;

if((front==0)&&(rear==-1))

{

printf("Queue is underflow\n");

exit(1);

}

if(front==rear)

{

a=q[front];

rear=-1;

front=0;

}

else if(front==max-1)

{

a=q[front];

front=0;



BMSIT&M, Bangalore, Dept of CSE Page 23

}

else

a=q[front++];

printf("Deleted element is:%d\n",a);

}

void display()

{

int i,j;

if(front==0&&rear==-1)

{

printf("Queue is underflow\n");

exit(1);

}

if(front>rear)

{

for(i=0;i<=rear;i++)

printf("\t%d",q[i]);

for(j=front;j<=max-1;j++)

printf("\t%d",q[j]);

printf("\nrear is at %d\n",q[rear]);

printf("\nfront is at %d\n",q[front]);

}

else

{

for(i=front;i<=rear;i++)

{

printf("\t%d",q[i]);

}

printf("\nrear is at %d\n",q[rear]);

printf("\nfront is at %d\n",q[front]);

}

printf("\n");

}

**Output**

linux:~/dslab #gedit cirQ.c

linux:~/dslab #gcc cirQ.c



BMSIT&M, Bangalore, Dept of CSE Page 24

linux:~/dslab # ./a.out

Circular Queue operations

1.insert

2.delete

3.display

4.exit

Enter your choice:1

Enter element to be insert:10

Enter your choice:1

Enter element to be insert:20

Enter your choice:1

Enter element to be insert:30

Enter your choice:3 10 20 30

rear is at 30

front is at 10

Enter your choice:2

Deleted element is:10

Enter your choice:3 20 30

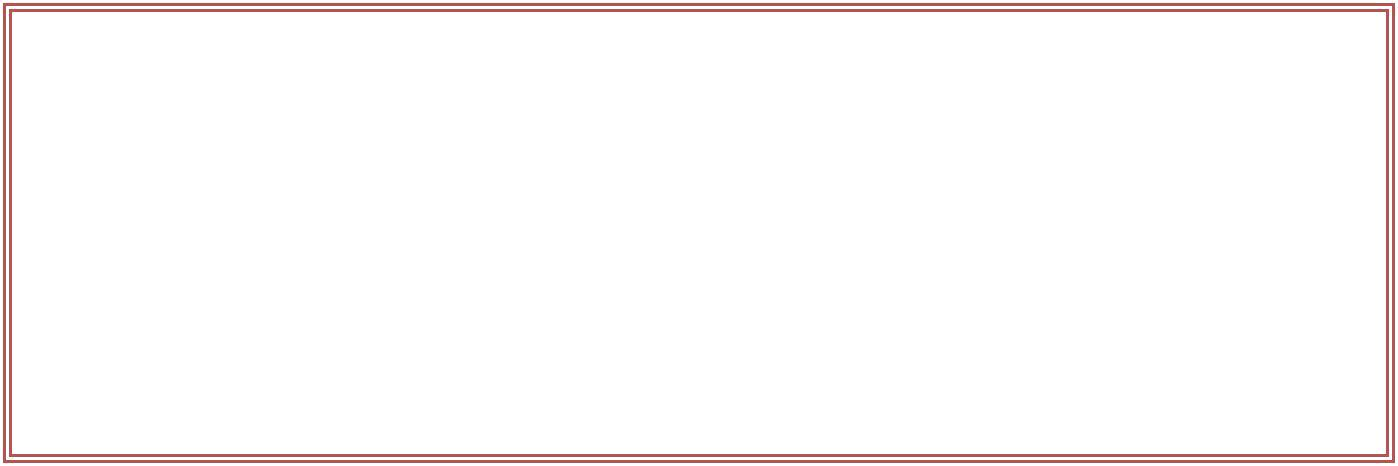
rear is at 30

front is at 20

Enter your choice:4



BMSIT&M, Bangalore, Dept of CSE Page 25



1. Design, Develop and Implement a menu driven Program in C for the following operations on Singly Linked List (SLL) of Student Data with the fields: *USN, Name, Branch, Sem, PhNo*
   1. Create a SLL of N Students Data by using *front insertion*.
   2. Display the status of SLL and count the number of nodes in it
   3. Perform Insertion and Deletion at End of SLL
   4. Perform Insertion and Deletion at Front of SLL
   5. Demonstrate how this SLL can be used as STACK and QUEUE
   6. Exit

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int count=0;

struct node

{

int sem,phno;

char name[20],branch[10],usn[20];

struct node \*next;

}\*first=NULL,\*last=NULL,\*temp=NULL,\*temp1;

void create()

{

int sem,phno;

char name[20],branch[10],usn[20];

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

printf("\n Enter usn,name, branch, sem, phno of student : ");

scanf("%s %s %s %d %d", usn, name,branch, &sem,&phno);

strcpy(temp->usn,usn);

strcpy(temp->name,name);

strcpy(temp->branch,branch);

temp->sem = sem;

temp->phno = phno;

temp->next=NULL;

count++;

}

void insert\_atfirst()

{

if (first == NULL)



BMSIT&M, Bangalore, Dept of CSE Page 26

{

create();

first = temp;

last = first;

}

else

{

create();

temp->next = first;

first = temp;

}

}

void insert\_atlast()

{

if(first==NULL)

{

create(); first = temp;

last = first;

}

else

{

create();

last->next = temp;

last = temp;

}

}

void display()

{

temp1=first;

if(temp1 == NULL)

{

printf("List empty to display \n"); return;

}

printf("\n Linked list elements from begining : \n"); while (temp1!= NULL)

{

printf("%s %s %s %d %d\n", temp1->usn, temp1->name,temp1->branch,temp1->sem,temp1->phno );



BMSIT&M, Bangalore, Dept of CSE Page 27

temp1 = temp1->next;

}

printf(" No of students = %d ", count);

}

int deleteend()

{

struct node \*temp;

temp=first;

if(temp->next==NULL)

{

free(temp);

first=NULL;

}

else

{

while(temp->next!=last) temp=temp->next;

printf("%s %s %s %d %d\n", last->usn, last->name,last->branch, last->sem, last->phno );

free(last);

temp->next=NULL;

last=temp;

}

count--;

return 0;

}

int deletefront()

{

struct node \*temp;

temp=first;

if(temp->next==NULL)

{

free(temp);

first=NULL; return 0;

}

else

{

first=temp->next;



BMSIT&M, Bangalore, Dept of CSE Page 28

printf("%s %s %s %d %d", temp->usn, temp->name,temp->branch,temp->sem, temp->phno ); free(temp);

}

count--; return 0;

}

void main()

{

int ch,n,i; first=NULL;

temp = temp1 = NULL;

printf("-----------------MENU----------------------\n");

printf("\n 1 create a SLL of n emp");

printf("\n 2 - Display from beginning");

printf("\n 3 - Insert at end");

printf("\n 4 - delete at end");

printf("\n 5 - Insert at beg");

printf("\n 6 - delete at beg");

printf("\n 7 - exit\n");

printf("-------------------------------------------\n");

while (1)

{

printf("\n Enter choice : ");

scanf("%d", &ch);

switch (ch)

{

case 1:

printf("\n Enter no of students : ");

scanf("%d", &n);

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

insert\_atfirst();

break;

case 2: display();

break;

case 3: insert\_atlast();

break;

case 4:deleteend();

break;

case 5:insert\_atfirst();

break;



BMSIT&M, Bangalore, Dept of CSE Page 29

case 6: deletefront();

break;

case 7: exit(0);

default: printf("wrong choice\n");

}

}

}

**Output**

linux:~/dslab #gedit slink.c

linux:~/dslab #gcc slink.c

linux:~/dslab # ./a.out

–---------------MENU----------------------

1. – create a SLL of n emp
2. -Display from beginning
3. -Insert at end
4. -delete at end
5. -Insert at beg
6. -delete at beg
7. -exit

------------------------------------------------

Enter choice : 1

Enter no of students : 2

Enter usn,name, branch, sem, phno of student : 007 vijay CSE 3 121

Enter usn,name, branch, sem, phno of student : 100 yashas CSE 3 911

Enter choice : 2

Linked list elements from begining : 100 yashas CSE 3 911 007 vijay CSE 3 121

No of students = 2

Enter choice : 3

Enter usn,name, branch, sem, phno of student : 001 raj CSE 3 111

Enter choice : 2

Linked list elements from begining : 100 yashas CSE 3 911 007 vijay CSE 3 121

001 raj CSE 3 111 No of students = 3



BMSIT&M, Bangalore, Dept of CSE Page 30

Enter choice : 4 001 raj CSE 3 111 Enter choice : 2

Linked list elements from begining : 100 yashas CSE 3 911

007 vijay CSE 3 121 No of students = 2

Enter choice : 5

Enter usn,name, branch, sem, phno of student : 003 harsh cse 3 111

Enter choice : 2

Linked list elements from begining : 003 harsh cse 3 111 100 yashas CSE 3 911

007 vijay CSE 3 121 No of students = 3

Enter choice : 6 003 harsh cse 3 111

Enter choice : 2

Linked list elements from begining : 100 yashas CSE 3 911

007 vijay CSE 3 121 No of students = 2

Enter choice : 7



BMSIT&M, Bangalore, Dept of CSE Page 31



1. 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*

* 1. Create a DLL of N Employees Data by using *end insertion*.
  2. Display the status of DLL and count the number of nodes in it
  3. Perform Insertion and Deletion at End of DLL
  4. Perform Insertion and Deletion at Front of DLL
  5. Demonstrate how this DLL can be used as Double Ended Queue
  6. Exit

#include<string.h> int count=0;

struct node

{

struct node \*prev; int ssn,phno;

float sal;

char name[20],dept[10],desg[20];

struct node \*next;

}\*h,\*temp,\*temp1,\*temp2,\*temp4;

void create()

{

int ssn,phno; float sal;

char name[20],dept[10],desg[20];

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

temp->prev = NULL;

temp->next = NULL;

printf("\n Enter ssn,name,department, designation, salary and phno of employee : "); scanf("%d %s %s %s %f %d", &ssn, name,dept,desg,&sal, &phno);

temp->ssn = ssn;

strcpy(temp->name,name);

strcpy(temp->dept,dept);

strcpy(temp->desg,desg);

temp->sal = sal;

temp->phno = phno;

count++;

}



BMSIT&M, Bangalore, Dept of CSE Page 32

void insertbeg()

{

if (h == NULL)

{

create(); h = temp;

temp1 = h;

}

else

{

create();

temp->next = h;

h->prev = temp;

h = temp;

}

}

void insertend()

{

if(h==NULL)

{

create();

h = temp;

temp1 = h;

}

else

{

create();

temp1->next = temp;

temp->prev = temp1;

temp1 = temp;

}

}

void displaybeg()

{

temp2 =h;

if(temp2 == NULL)

{

printf("List empty to display \n");



BMSIT&M, Bangalore, Dept of CSE Page 33

return;

}

printf("\n Linked list elements from begining : \n");

while (temp2!= NULL)

{

printf("%d %s %s %s %f %d\n", temp2->ssn, temp2->name,temp2->dept,temp2->desg,temp2->sal, temp2->phno );

temp2 = temp2->next;

}

printf(" No of employees = %d ", count);

}

int deleteend()

{

struct node \*temp; temp=h;

if(temp->next==NULL)

{

free(temp);

h=NULL; return 0;

}

else

{

temp2=temp1->prev; temp2->next=NULL;

printf("%d %s %s %s %f %d\n", temp1->ssn, temp1->name,temp1->dept, temp1->desg,temp1->sal, temp1->phno );

free(temp1);

}

count--; return 0;

}

int deletebeg()

{

struct node \*temp; temp=h;

if(temp->next==NULL)

{

free(temp);

h=NULL;

}

else



BMSIT&M, Bangalore, Dept of CSE Page 34

{

h=h->next;

printf("%d %s %s %s %f %d", temp->ssn, temp->name,temp->dept, temp->desg,temp->sal, temp->phno );

free(temp);

}

count--; return 0;

}

void main()

{

int ch,n,i; h=NULL;

temp = temp1 = NULL;

printf("-----------------MENU--------------------\n");

printf("\n 1 - create a DLL of n emp");

printf("\n 2 - Display from beginning");

printf("\n 3 - Insert at end");

printf("\n 4 - delete at end");

printf("\n 5 - Insert at beg");

printf("\n 6 - delete at beg");

printf("\n 7 - exit\n");

printf("------------------------------------------\n"); while (1)

{

printf("\n Enter choice : ");

scanf("%d", &ch);

switch (ch)

{

case 1:

printf("\n Enter no of employees : ");

scanf("%d", &n);

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

insertend();

break;

case 2:

displaybeg();

break;

case 3:

insertend();



BMSIT&M, Bangalore, Dept of CSE Page 35

break;

case 4:

deleteend();

break;

case 5:

insertbeg();

break;

case 6:

deletebeg();

break;

case 7: exit(0);

default: printf("wrong choice\n");

}

}

}

**Output**

linux:~/dslab #gedit dlink.c

linux:~/dslab #gcc dlink.c

linux:~/dslab # ./a.out

--------------MENU--------------------

–Create a DLL of n emp

* Display from beginning
* Insert at end
* Delete at end
* Insert at beg
* Delete at beg
* exit

------------------------------------------

Enter choice : 1

Enter no of employees : 2

Enter ssn,name,department, designation, salary and phno of employee : 1 RAJ SALES MANAGER 15000 911

Enter ssn,name,department, designation, salary and phno of employee :

2 RAVI HR ASST 10000 123



BMSIT&M, Bangalore, Dept of CSE Page 36

Enter choice : 2

Linked list elements from begining :

1 RAJ SALES MANAGER 15000.000000 911

2 RAVI HR ASST 10000.000000 123 No of employees = 2 Enter choice : 3

Enter ssn,name,department, designation, salary and phno of employee : 3 RAM MARKET MANAGER 50000 111

Enter choice : 2

Linked list elements from begining :

1 RAJ SALES MANAGER 15000.000000 911

2 RAVI HR ASST 10000.000000 123

3 RAM MARKET MANAGER 50000.000000 111 No of employees = 3 Enter choice : 4

3 RAM MARKET MANAGER 50000.000000 111 Enter choice : 2

Linked list elements from begining :

1 RAJ SALES MANAGER 15000.000000 911

2 RAVI HR ASST 10000.000000 123 No of employees = 2 Enter choice : 5

Enter ssn,name,department, designation, salary and phno of employee :

0 ALEX EXE TRAINEE 2000 133

Enter choice : 2

Linked list elements from begining :

0 ALEX EXE TRAINEE 2000.000000 133

1 RAJ SALES MANAGER 15000.000000 911

2 RAVI HR ASST 10000.000000 123

No of employees = 3

Enter choice : 6

0 ALEX EXE TRAINEE 2000.000000 133

Enter choice : 2

Linked list elements from begining :

1 RAJ SALES MANAGER 15000.000000 911

2 RAVI HR ASST 10000.000000 123

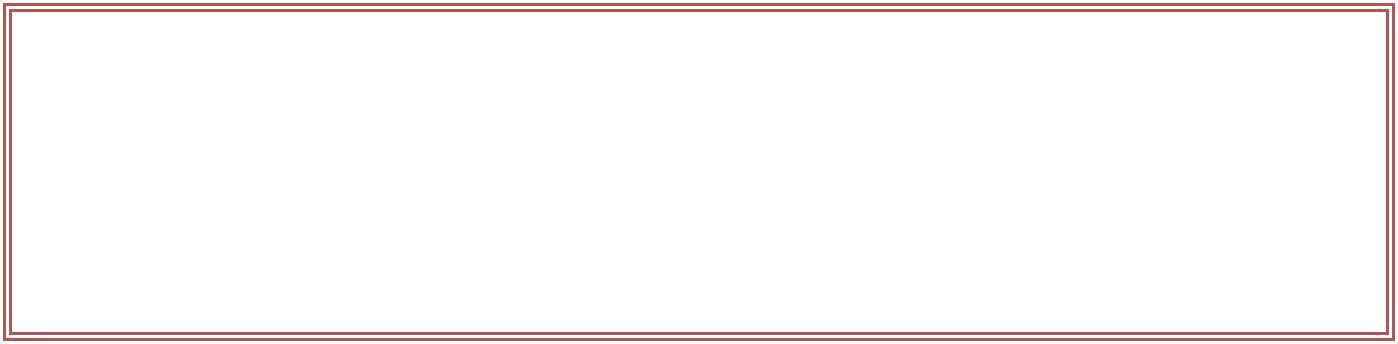
No of employees = 2

Enter choice : 7

Exit



BMSIT&M, Bangalore, Dept of CSE Page 37

1. Design, Develop and Implement a Program in C for the following operations on Singly Circular Linked List (SCLL) with header nodes
   1. Represent and Evaluate a Polynomial P(x,y,z) = 6x2y2z-4yz5+3x3yz+2xy5z-2xyz3
   2. Find the sum of two 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<math.h>

#include<stdlib.h>

#include<math.h>

typedef struct node

{

int expo,coef;

struct node \*next;

}node;

node \* insert(node \*,int,int);

node \* create();

node \* add(node \*p1,node \*p2);

int eval(node \*p1);

void display(node \*head);

node \*insert(node\*head,int expo1,int coef1)

{

node \*p,\*q;

p=(node \*)malloc(sizeof(node));

p->expo=expo1;

p->coef=coef1;

p->next=NULL;

if(head==NULL)

{

head=p;

head->next=head;

return(head);

}

if(expo1>head->expo)

{

p->next=head->next;



BMSIT&M, Bangalore, Dept of CSE Page 38

head->next=p;

head=p;

return(head);

}

if(expo1==head->expo)

{

head->coef=head->coef+coef1;

return(head);

}

q=head;

while(q->next!=head&&expo1>=q->next->expo)

q=q->next;

if(p->expo==q->expo)

q->coef=q->coef+coef1;

else

{

p->next=q->next;

q->next=p;

}

return(head);

}

node \*create()

{

int n,i,expo1,coef1; node \*head=NULL;

printf("\n\nEnter no of terms of polynomial==>"); scanf("%d",&n);

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

{

printf("\n\nEnter coef & expo==>"); scanf("%d%d",&coef1,&expo1); head=insert(head,expo1,coef1);

}

return(head);

}



BMSIT&M, Bangalore, Dept of CSE Page 39

node \*add(node \*p1,node \*p2)

{

node \*p;

node \*head=NULL;

printf("\n\n\nAddition of polynomial==>");

p=p1->next;

do

{

head=insert(head,p->expo,p->coef);

p=p->next;

}

while(p!=p1->next);

p=p2->next;

do

{

head=insert(head,p->expo,p->coef);

p=p->next;

}while(p!=p2->next);

return(head);

}

int eval(node \*head)

{

node \*p; int x,ans=0;

printf("\n\nEnter the value of x=");

scanf("%d",&x);

p=head->next;

do

{

ans=ans+p->coef\*pow(x,p->expo);

p=p->next;

}while(p!=head->next);

return(ans);

}

void display(node \*head)

{

node \*p,\*q; int n=0;

q=head->next;

p=head->next;



BMSIT&M, Bangalore, Dept of CSE Page 40

do

{

n++; q=q->next;

} while(q!=head->next); printf("\n\n\tThe polynomial is==>"); do

{

if(n-1)

{

printf("%dx^(%d) + ",p->coef,p->expo);

p=p->next;

}

else

{

printf(" %dx^(%d)",p->coef,p->expo);

p=p->next;

}

n--;

} while(p!=head->next);

}

void main()

{

int a,x,ch;

node \*p1,\*p2,\*p3; p1=p2=p3=NULL;

while(1)

{

printf("\n\t----------------<< MENU >>---------------"); printf("\n\tPolynomial Operations :");

printf("\n\t\t\t\t2 1.Add");

printf("\n\t\t\t\t2.Evaluate");

printf("\n\t\t\t\t3.Exit");

printf("\n\t------------------------------------------- ");

printf("\n\n\n\tEnter your choice==>");

scanf("%d",&ch);

switch(ch)

{

case 1 : p1=create();

display(p1);

p2=create();



BMSIT&M, Bangalore, Dept of CSE Page 41

display(p2);

p3=add(p1,p2);

display(p3);

break;

case 2 : p1=create();

display(p1);

a=eval(p1);

printf("\n\nValue of polynomial=%d",a);

break;

case 3 : exit(0);

break;

default :printf("\n\n\tinvalid choice");

break;

}

}

}

**Output**

linux:~/dslab #gedit poly.c

linux:~/dslab #gcc poly.c

linux:~/dslab # ./a.out

-----------------<< MENU >>---------------

Polynomial Operations:

1. Add
2. Evaluate
3. Exit

---------------------------------------------------

Enter your choice==>1

Enter no of terms of polynomial==>3

Enter coef & expo==>

4

3

Enter coef & expo==> 2 2

Enter coef & expo==> 5 1

The polynomial is==>5x^(1) + 2x^(2) + 4x^(3)

Enter no of terms of polynomial==>3

Enter coef & expo==> 4 1

Enter coef & expo==>3 2

Enter coef & expo==> 5 3

The polynomial is==>4x^(1) + 3x^(2) + 5x^(3)

Addition of polynomial==>



BMSIT&M, Bangalore, Dept of CSE Page 42

The polynomial is==>9x^(1) + 5x^(2) + 9x^(3)

Enter your choice==>2

Enter no of terms of polynomial==>3

Enter coef & expo==>3 1

Enter coef & expo==> 4 2

Enter coef & expo==> 5 4

The polynomial is==>3x^(1) + 4x^(2) + 5x^(4)

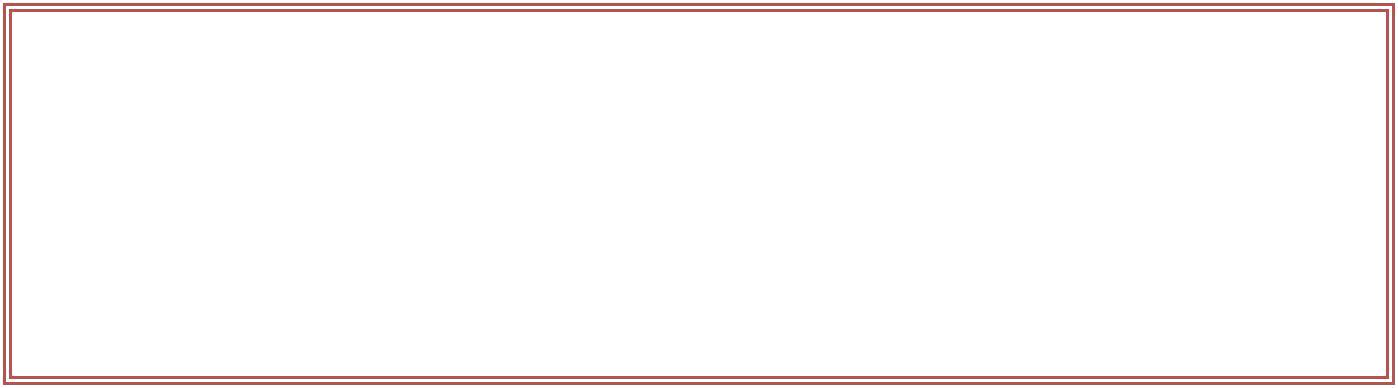
Enter the value of x=2

Value of polynomial=102 Enter your choice==>3

exit



BMSIT&M, Bangalore, Dept of CSE Page 43



1. Design, Develop and Implement a menu driven Program in C for the following operations on Binary Search Tree (BST) of Integers
   1. Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2
   2. Traverse the BST in Inorder, Preorder and Post Order
   3. Search the BST for a given element (KEY) and report the appropriate message
   4. Delete an element(ELEM) from BST
   5. Exit

* include <stdio.h>
* include <stdlib.h>

int flag=0;

typedef struct BST

{

int data;

struct BST \*lchild, \*rchild;

}node;

void insert(node \*, node \*);

void inorder(node \*);

void preorder(node \*);

void postorder(node \*);

node \*search(node \*, int, node \*\*);

void main()

{

int choice;

int ans =1;

int key;

node \*new\_node, \*root, \*tmp, \*parent;

node \*get\_node();

root = NULL;

printf("\nProgram For Binary Search Tree ");

do

{

printf("\n1.Create");

printf("\n2.Search");

printf("\n3.Recursive Traversals");

printf("\n4.Exit");

printf("\nEnter your choice :");

scanf("%d", &choice);



BMSIT&M, Bangalore, Dept of CSE Page 44

switch (choice)

{

case 1:

do

{

new\_node = get\_node();

printf("\nEnter The Element ");

scanf("%d", &new\_node->data);

if (root == NULL) /\* Tree is not Created \*/

root = new\_node;

else

insert(root, new\_node);

printf("\nWant To enter More Elements?(1/0)");

scanf("%d",&ans);

}

while (ans);

break;

case 2:

printf("\nEnter Element to be searched :");

scanf("%d", &key);

tmp = search(root, key, &parent);

if(flag==1)

{

printf("\nParent of node %d is %d", tmp->data, parent->data);

}

else

{

printf("\n The %d Element is not Present",key);

}

flag=0;

break;

case 3:

if (root == NULL)

printf("Tree Is Not Created");

else

{

printf("\nThe Inorder display : ");

inorder(root);

printf("\nThe Preorder display : "); preorder(root);

printf("\nThe Postorder display : "); postorder(root);

}

break;



BMSIT&M, Bangalore, Dept of CSE Page 45

}

} while (choice != 4);

}

/\* Get new Node \*/

node \*get\_node()

{

node \*temp;

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

temp->lchild = NULL;

temp->rchild = NULL;

return temp;

}

/\*This function is for creating a binary search tree \*/

void insert(node \*root, node \*new\_node)

{

if (new\_node->data < root->data)

{

if(root->lchild == NULL) root->lchild = new\_node; else insert(root->lchild, new\_node); }

if (new\_node->data > root->data)

{

if (root->rchild == NULL)

root->rchild = new\_node;

else

insert(root->rchild, new\_node);

}

}

/\* This function is for searching the node from binary Search Tree \*/

node \*search(node \*root, int key, node \*\*parent)

{

node \*temp;

temp = root;

while (temp != NULL)

{

if (temp->data == key)

{

printf("\nThe %d Element is Present", temp->data); flag=1;



BMSIT&M, Bangalore, Dept of CSE Page 46

return temp;

}

\*parent = temp;

if (temp->data > key)

temp = temp->lchild;

else

temp = temp->rchild;

}

return NULL;

}

/\* This function displays the tree in inorder fashion \*/

void inorder(node \*temp)

{

if (temp != NULL)

{

inorder(temp->lchild);

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

inorder(temp->rchild);

}

}

/\*This function displays the tree in preorder fashion \*/

void preorder(node \*temp)

{

if (temp != NULL)

{

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

preorder(temp->lchild);

preorder(temp->rchild);

}

}

/\*This function displays the tree in postorder fashion \*/ void postorder(node \*temp)

{

if (temp != NULL)

{

postorder(temp->lchild);

postorder(temp->rchild);

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

}

}



BMSIT&M, Bangalore, Dept of CSE Page 47

**Output**

linux:~/dslab #gedit bst.c

linux:~/dslab #gcc bst.c

linux:~/dslab # ./a.out

Program For Binary Search Tree

1.Create

2.Search

3.Recursive Traversals

4.Exit

Enter your choice :1

Enter The Element 15

Want To enter More Elements?(1/0)

1

Enter The Element 25

Want To enter More Elements?(1/0)

1

Enter The Element 35

Want To enter More Elements?(1/0)

1

Enter The Element 45

Want To enter More Elements?(1/0)

1

Enter The Element 5

Want To enter More Elements?(1/0)

1

Enter The Element 7

Want To enter More Elements?(1/0)

0

Enter your choice :2

Enter Element to be searched :7

The 7 Element is Present

Parent of node 7 is 5

1.Create

2.Search

3.Recursive Traversals



BMSIT&M, Bangalore, Dept of CSE Page 48

4.Exit

Enter your choice :2

Enter Element to be searched :88

The 88 Element is not Present

Enter your choice :3

The Inorder display : 5 7 15 25 35 45

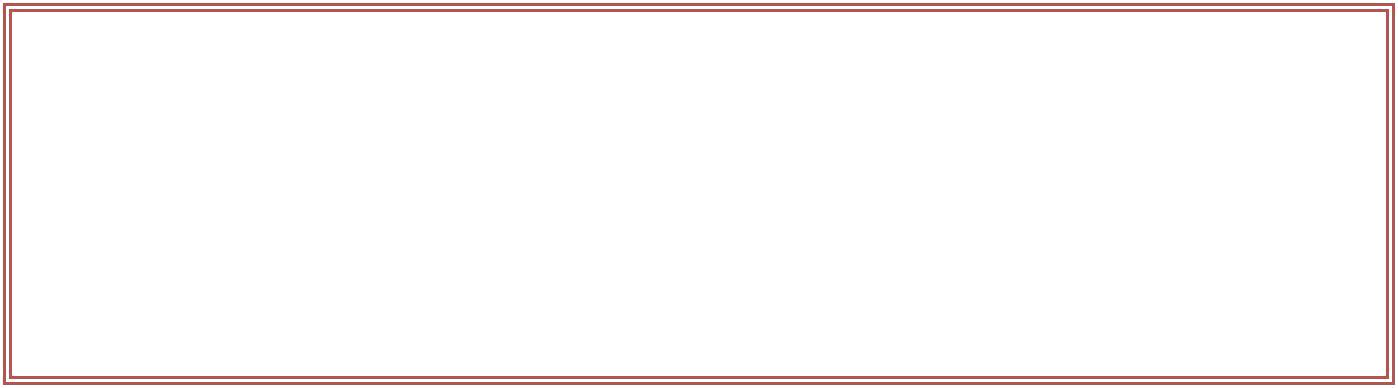
The Preorder display : 15 5 7 25 35 45

The Postorder display : 7 5 45 35 25 15

Enter your choice :4



BMSIT&M, Bangalore, Dept of CSE Page 49

1. Design, Develop and Implement a Program in C for the following operations on Graph(G) of Cities
   1. Create a Graph of N cities using Adjacency Matrix.
   2. Print all the nodes reachable from a given starting node in a digraph using BFS method
   3. Check whether a given graph is connected or not using DFS method

#include <stdio.h>

#include <stdlib.h>

int a[20][20],q[20],visited[20],reach[10],n,i,j,f=0,r=-1,count=0;

void bfs(int v)

{

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

if(a[v][i] && !visited[i])

q[++r]=i;

if(f<=r)

{

visited[q[f]]=1;

bfs(q[f++]);

}

}

void dfs(int v)

{

int i;

reach[v]=1;

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

{

if(a[v][i] && !reach[i])

{

printf("\n %d->%d",v,i);

count++;

dfs(i);

}

}

}

void main()

{

int v, choice;

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



BMSIT&M, Bangalore, Dept of CSE Page 50

scanf("%d",&n);

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

{

q[i]=0;

visited[i]=0;

}

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

reach[i]=0;

printf("\n Enter graph data in matrix form:\n");

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

for(j=1;j<=n;j++)

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

printf("1.BFS\n 2.DFS\n 3.Exit\n");

scanf("%d",&choice);

switch(choice)

{

case 1: printf("\n Enter the starting vertex:");

scanf("%d",&v);

bfs(v);

if((v<1)||(v>n))

{

printf("\n Bfs is not possible");

}

else

{

printf("\n The nodes which are reachable from %d:\n",v); for(i=1;i<=n;i++)

if(visited[i])

printf("%d\t",i);

}

break;

case 2:

dfs(1);

if(count==n-1)

printf("\n Graph is connected");

else

printf("\n Graph is not connected");



BMSIT&M, Bangalore, Dept of CSE Page 51

break;

case 3: exit(0);

}

}

**Output**

linux:~/dslab #gedit bfs.c

linux:~/dslab #gcc bfs.c

linux:~/dslab # ./a.out

Enter the number of vertices:5

Enter graph data in matrix form:

0 1 0 1 0

1 0 1 0 1

0 1 0 1 0

1 0 1 0 0

0 1 0 0 0

1.BFS

2.DFS

3.Exit

2

1->2

2->3

3->4

2->5

Graph is connected

Enter the number of vertices:5

Enter graph data in matrix form:

0 1 0 1 0

1 0 1 0 0

0 1 0 1 0

1 0 1 0 0

0 0 0 0 0

1.BFS

2.DFS

3.Exit



BMSIT&M, Bangalore, Dept of CSE Page 52

2

1->2

2->3

3->4

Graph is not connected

Enter the number of vertices:5

Enter graph data in matrix form:

0 1 1 0 0

0 0 0 1 0

0 0 0 0 0

0 0 1 0 0

0 0 1 0 0

1.BFS

2.DFS

3.Exit

1

Enter the starting vertex:1

The nodes which are reachable from 1: 2 3 4

Enter graph data in matrix form:

0 1 1 0 0

0 0 0 1 0

0 0 0 0 0

0 0 1 0 0

0 0 1 0 0

1.BFS

2.DFS

3.Exit

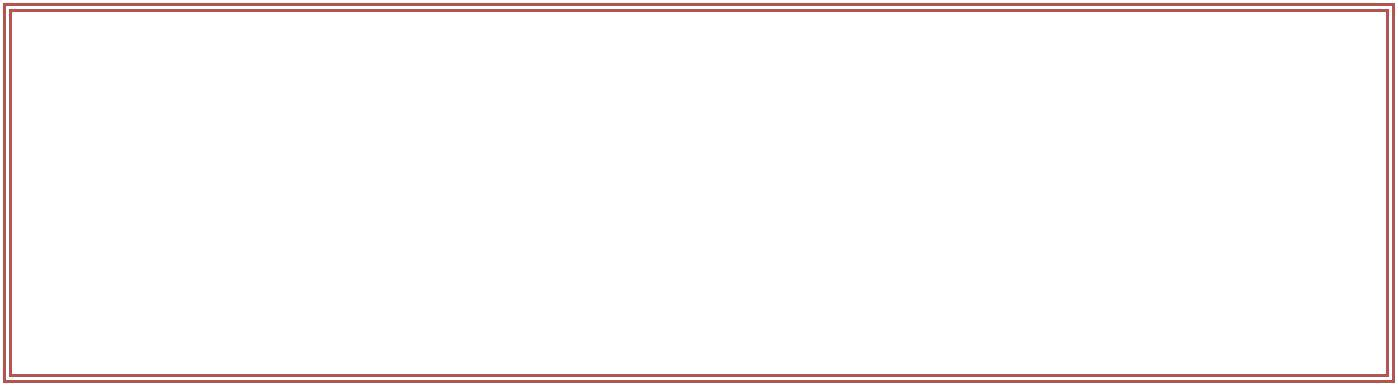
1

Enter the starting vertex:0

BFS is not possible



BMSIT&M, Bangalore, Dept of CSE Page 53



1. **G**iven a File of **N** employee records with a set **K** of Keys(4-digit) which uniquely determinethe 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 → L as H(K)=K 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>

#define MAX 100

int create(int);

void linear\_prob(int[], int, int);

void display (int[]);

void main()

{

int a[MAX],num,key,i;

int ans=1;

printf(" collision handling by linear probing : \n");

for (i=0;i<MAX;i++)

{

a[i] = -1;

}

do

{

printf("\n Enter the data");

scanf("%4d", &num);

key=create(num);

linear\_prob(a,key,num);

printf("\n Do you wish to continue ? (1/0) ");

scanf("%d",&ans);

}while(ans);

display(a);

}



BMSIT&M, Bangalore, Dept of CSE Page 54

int create(int num)

{

int key;

key=num%100;

return key;

}

void linear\_prob(int a[MAX], int key, int num)

{

int flag, i, count=0;

flag=0;

if(a[key]== -1)

{

a[key] = num;

}

else

{

printf("\nCollision Detected...!!!\n");

i=0;

while(i<MAX)

{

if (a[i]!=-1)

count++;

i++;

}

printf("Collision avoided successfully using LINEAR PROBING\n");

if(count == MAX)

{

printf("\n Hash table is full");

display(a);

exit(1);

}

for(i=key+1; i<MAX; i++)

if(a[i] == -1)

{

a[i] = num; flag =1;

break;

}



BMSIT&M, Bangalore, Dept of CSE Page 55

//for(i=0;i<key;i++)

i=0;

while((i<key) && (flag==0))

{

if(a[i] == -1)

{

a[i] = num;

flag=1;

break;

}

i++;

}

}

}

void display(int a[MAX])

{

int i,choice;

printf("1.Display ALL\n 2.Filtered Display\n");

scanf("%d",&choice);

if(choice==1)

{

printf("\n the hash table is\n");

for(i=0; i<MAX; i++)

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

}

else

{

printf("\n the hash table is\n");

for(i=0; i<MAX; i++)

if(a[i]!=-1)

{

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

continue;

}

}

}



BMSIT&M, Bangalore, Dept of CSE Page 56

**Output**

linux:~/dslab #gedit hash.c

linux:~/dslab #gcc hash.c

linux:~/dslab # ./a.out

collision handling by linear probing :

Enter the data1234

Do you wish to continue? (1/0) 1

Enter the data2548

Do you wish to continue? (1/0) 1

Enter the data3256

Do you wish to continue? (1/0) 1

Enter the data1299

Do you wish to continue? (1/0) 1

Enter the data1298

Do you wish to continue? (1/0) 1

Enter the data1398

Collision Detected...!!!

Collision avoided successfully using LINEAR PROBING Do you wish to continue? (1/0) 0

1.Display ALL

2.Filtered Display

2

the hash table is

1. 1398
2. 1234
3. 2548
4. 3256
5. 1298
6. 1299



BMSIT&M, Bangalore, Dept of CSE Page 57