DATA STRUCTURE

LAB FILE

Department of Computer Science & Engineering

Shri Shankaracharya Institute of Professional Management & Technology, Raipur

CERTIFICATE

This is to certify that Mr/Ms Ashish Kumar Sahu

has satisfactorily completed the required number of programs in Data Stucture Lab as per the University syllabus of B.E. III Semester for the academic session 2020.

Signature of Faculty

Date:

INDEX

S.No.	Program	Date of Performance	Date of Submission	Signature (of faculty
1	Write a program to perform insertion on array.			
2	Write a program to perform deletion on array.			
3	Write a program to perform reverse of array			
4	Write a program to perform binary search on linear array.			
5	Write a program to perform sequential search on linear array.			
6	Write a program to perform insertion sort on input array.			
7	Write a program to perform bubble sort on input array.			

8	Write a program to perform selection sort on input array.			
9	Write a program to implement stack and perform push and pop operation.			
10	Write a program to perform Traverse, insertion, deletion oven Singly link list.			
11	Write a program to convert infix to postfix expression.			
12	Write a program to perform Quick sort on input array.			

Topic :... Program No 1 ... Date :..... Page :..... 1 .....

Aim :- Write a program to perform insertion on array.

Code:

```
#include<iostream>
using namespace std;
void insertion(int list[], int lb, int ub, int max
               int item, int place)
{ int temp, i, j, k;
  if (place+1 > ub)
  { cout << "error";
    return;
  }
  if (ub > max)
  { cout << "error";
    return;
  }
  for(i=place; i<ub, i++)
  { temp = list[i];
    list[i] = item;
    item = temp;
  }
  cout << "\n after insertion \n";
  for(k=lb, k<ub; k++)
  { cout << list[k] << endl; }
}
```

```
int main()
{  int a[5], i, p, n;
   cout<<"enter array"<<endl;
   for(i=0; i<5; i++)
   {  cin>>a[i];
   }
   cout<<"\n enter place(between 1 to 5) and
       number\n";
   cin>>p>>n;
   insertion(a, 0, 5, 5, n, p-1);
   return 0;
}
```

Output :-

```
enter array
44
5
78
32
50
enter place(between 1 to 5) and number
4
888

after insertion
44
5
78
888
32
```

Aim :- Write a program to perform deletion on Array.

Code :-

```
#include<iostream>
using namespace std;
void deletion(int list[], int lb, int ub, int max, int place)
{ if(ub>max)
  { cout<<" error"; return; }
  if(place>max)
  { cout<<" error";
    return; }
  int i,j,k;
  for(i=place-1; i<=ub; i++)
  { int temp;
    list[i]=list[i+1];
  }
}
void traverse(int list[], int lb, int ub, int max)
{ if(ub>max)
  { cout<<"error";
    return; }
  for(int i=lb; i<ub, i++)
  { cout<<list[i]<<endl;
  }
```

```
}

int main()
{ int i,p,a[5];
  cout<<"enter array"<<endl;
  for(i=0;i<5;i++)
  { cin>>a[i];
  }
  cout<<"enter the place(between 1 to 5) for
      performing deletion:"<<endl;
  cin>>p;
  cout<<"before deletion, array is:"<<endl;
  traverse(a,0,5,5)
  deletion(a,0,5,5,p);
  cout<<"after deletion, array is:"<<endl;
  traverse(a,0,5,5);
}
```

OUTPUT:-

```
enter array
22
67
8
10
6
enter the Place for Performing deletion:
2
before deletion, array is:
22
67
8
10
6
after deletion, array is:
22
8
10
6
0
```

Aim :- Write a program to perform reverse on Array.

Code :-

```
#include<iostream>
#include<conio.h>
using namespace std;
Void traverse(int lb, int ub, int list[], int max)
{  if(lb>ub)
   { cout<<" error " ;  return ;
   }
   if(ub>max)
   { cout<<" error" ; return;
   }
   for(int i=0; i<max; i++)
   { cout<<'\t'<<list[i];
   }
}
Void reverse(int lb, int ub, int list[], int max)
{  if(lb>ub)
   { cout<<" error" ;  return ;
   }
   if(ub>max)
   { cout<<" error" ; return;
   }
```

```
cout << "before performing reverse: " << endl;
traverse(lb, ub, list, max);
int temp;
for(int i=lb, i<=ub/2 ; i++)
{  temp = list[i];
   list[i] = list[ub-i+1];
   list[ub-i+1] = temp;
}
cout << endl << "after performing reverse: " << endl;
traverse(lb, ub, list, max);
}

int main()
{  int lb, ub, list[5], max;
   cout << " enter the array: " << endl;
   for(int i=0; i<5 ; i++)
   {  cin >> list[i];
   }
   cout << " enter lb, ub, max : " << endl;
   cin >> lb >> ub >> max;
   reverse(ub, ub, list, max);
   return 0;
}
```

Output:

```
enter the array:
2
5
8
90
12
enter lb, ub, max:
0
5
5
before deletion: performing reverse:
    2    5    8    90    12
after performing reverse:
    12    90    8    5    2
```

Aim :- Write a program to perform binary search on linear array.

Code :-

```
#include<iostream>
using namespace std;
int binary(int list[], int l, int r, int x)
{  int mid = l +((l+(r-1))/2);
   if (list[mid] ==x)
   {  cout << mid+1 <<endl;
      return mid+1;
   }
   if(x<list[mid])
   {  binary (list, l, mid, x);
   }
   if (x > list[mid])
   {  binary(list, mid, r, x);
   }
   return 0;
}

int main()
{
   int a[10] = {2, 44, 55, 69, 74, 88, 91, 119, 154, 200};
```

```
int i,x;
cout<<"Array is :"<<endl;
for(i=0; i<10; i++)
{
   cout<<a[i]<<"\t";
}
cout<<"\n enter number for binary search:"<<endl;
cin>>x;
cout<<"index of "<<x<<" is";
binary(a,0,10,x);
return 0;
}
```

Output :-

```
array is :
2 44 55 69 74 88 91 119 154 200
enter number for binary search:
74
index of 74 is: 5
```

```
array is :
2 44 55 69 74 88 91 119 154 200
enter number for binary search:
154
index of 154 is: 9
```

Topic :..Program No.:5..... Date :..... Page :..9.....

Aim :- Write a Program to Perform sequential Search on linear array.

Code :-

```
#include<iostream>
using namespace std;
int seq_srch(int list[], int lb, int ub, int max, int item)
{ if(lb>ub || ub>max)
  { cout<<"error";
    return 0;
  }
  int i, n=-2;
  for(i=lb; i<ub; i++)
  {   if (list[i] == item)
      { n=i;
        break;
      }
  }
  if(n==-2)
  { cout<<"item not found";
  }
  else
  { cout<<" item found at index"<<i+1;
  }
```

```
    return 0;
}

int main(){
    int a[5], x, n, i;
    cout<<" enter numbers: " << endl;
    for(i=0; i<5; i++)
    {   cin >> a[i];
    }
    cout<<" enter number for search: "<<endl;
    cin>>x;
    sel_srch(a,0,5,5,x);
    return 0;
}
```

Output:

```
enter numbers:
22
56
77
85
23
enter number for search:
85
item found at index 4
```

Topic :..Program No:..6..... Date :..... Page :..11.....

Aim :- Write a program to perform insertion short on input array.

Code :-

```
#include<iostream>
using namespace std;
void insertion(int list[], int lb, int ub, int max)
{   if (lb>ub || ub>max)
    { cout<<"error" ; return ;
    }
    int i, j, k, temp, flag ;
    for(i=lb+1 ; i<ub ; i++)
    { flag = list[i] ;
      for (j=i-1 ; j>=lb ; j--)
      { if (flag< list[i])
        { temp = list[j] ;
          list[j] = list[j+1] ;
          list[j+1] =temp
        }
      }
    }
    cout<<"\n shorted array is : \n" ;
    for(k=lb ; k<ub ; k++)
    { cout <<list[k] <<endl ;
    }
```

```
}
int main()
{ int i, a[5];
  cout<<"enter unshorted array"<<endl;
  for(i=0;i<5;i++)
  { cin>>a[i];
  }
  insertion(a,0,5,5);
  return 0;
}
```

Output :-

```
enter unshorted array
34
67
11
09
55
 shorted array is :
9
11
34
55
67
```

Topic : Program No. 7 Date : Page : 13

Aim :- Write a program to perform bubble short on input array ;

Code :-

```
#include<iostream>
using namespace std;
void bubble(int list[], int lb, int ub, int max)
{  int i, j, temp, k;
   for(i=lb; i<ub-1; i++)
   {  for(j=lb; j<ub-1-i; j++)
      {  if (list[j] > list[j+1])
         {  temp = list[j];
            list[j] = list[j+1];
            list[j+1] = temp;
         }
      }
   }
   cout<<" shorted array is \n";
   for(k=lb; k<=ub; k++)
   {  cout<<list[k]<<endl;
   }
}
```

```
int main()
{
  int i,a[5];
  cout<<" enter unsorted array "<<endl;
  for(i=0;i<5;i++)
  { cin>>a[i];
  }
  bubble(a,0,5,5);
  return 0;
}
```

Output:

```
enter unshorted array
222
76
33
91
9
 Shorted array is
9
33
76
91
222
```

Aim :- Write a program to perform selection short on input array:

Code :-

```
#include<iostream>
Using namespace std;
Void selection(int list[], int lb, int ub, int max)
{
  int i,j,k,small,temp;
  for(i=lb, i<ub-1, i++)
  {  small = list[i];
     for(j=i+1; j<ub; j++)
     {  if(small > list[j]){
           temp = small;
           small = list[j];
           list[j] = temp;
        }
     }
     list[i] = small;
  }
  cout<<"\n shorted array is \n";
  for(k=lb; k<ub; k++)
  {  cout<<list[k]<<endl;
  }
}
```

```
int main()
{
   int i;a[5];
   cout<<" enter unshorted array"<<endl;
   for(i=0; i<5; i++)
   {  cin>>a[i];
   }
   selection(a,0,5,5);
   return 0;
}
```

Output :-

```
enter unshorted array
23
56
1
89
0

Shorted array is
0
1
23
56
89
```

Topic :..Program No: 9..... Date :..... Page :..16.1

Aim:- Write a program to implement stack and perform Push and Pop Operation.

Code:-

```
#include<iostream>
using namespace std;
#define max 100
class stackdemo
{  int top;
   public:
      int a[max];
      stackdemo()
      {  top = -1;
      }
      bool push(int x)
      int pop();
      int display();
      bool isempty();
};
bool stackdemo::push(int x)
{  if(top>=(max-1))
   {  cout<<"stack overflow";
      return false;
   }
```

```
        else
        { a[++top] = x ;
          cout << x << " pushed into stack \n";
          return true;
        }
}
int stackdemo :: Pop ()
{  if(top < 0)
   {  cout << "stack underflow";
      return 0;
   }
   else
   { int x = a[top--];
     return x;
   }
}
int stackdemo :: display()
{  if (top < 0)
   {  cout << "stack is empty";
      return 0;
   }
   else
   { int x = a[top];
     return x;
   }
}
```

```
bool Stackdemo::isempty()
{  return (top<0);
}

int main()
{ Stackdemo s;
   s.push(10);
   s.push(20);
   s.push(30);
   cout<<s.pop()<<" popped from stack \n";
   cout<<s.display()<<endl;
   cout<<s.isempty();
   return 0;
}
```

OUTPUT:

```
10 Pushed into stack
20 Pushed into stack
30 Pushed into stack.
30 Popped from stack.
20
0
```

Aim :- Write a program to perform Traverse, insertion, deletion over Singly link list.

Code:

```
#include<iostream>
Using namespace std;
Struct node
{  int data;
   node *next;
};
Class list
{  Private:
        node *head;
        node *tail;
   Public:
        list()
        {  head == NULL;
           tail == NULL;
        }
        void insert_start(int);
        void insert_end(int);
        Void insertat position (int, int);
        void delete_first();
        void delete_last();
        void deleteat position (int);
```

```
    Void traverse();
};
Void list:: insert_start(int item)
{  node *temp = new node;
   temp -> data = item;
   temp -> next = NULL;
   if(head = NULL)
   {   head = temp;
       tail = temp;
       temp = null;
   }
   else
   { temp -> next = head;
       head = temp;
   }
}
Void list:: insert_end(int item)
{  node *temp = new node;
   temp -> data = item;
   temp -> next = NULL;
   tail -> next = temp;
   tail = temp;
}
Void list:: insertatposition(int pos, int item)
{  node * temp = new node;
   node * temp1 = new node;
```

```
node *temp2 = new node;
int j;
temp2 = head;
for(j=1; temp2 != NULL; j++)
{ temp2 = temp2->next;
}
if(j<pos)
{cout<<" link list has only "<<j-1<<" contents so
you can't insert data at position "<<pos<<endl;
  return;
}
temp2 = head;
temp2->data = item;
temp2->next = NULL;
for(int i=1; i<pos; i++)
{ temp1 = temp2;
   temp2 = temp2->next;
}
temp->next = temp2;
temp1->next = temp;
}
void list::delete_first()
{  node *temp = new node;
   temp = head;
   temp = temp->next;
   head = temp;
}
```

```
Void delete_last()
{  node *temp1 = newnode;
   node *temp2 = new node;
   temp2 = head;
   while(temp2->next != NULL)
   {  temp1 = temp2;
      temp2 = temp2->next;
   }
   temp1->next = NULL;
   tail = temp1;
}
Void list:: deleteatposition(int pos)
{  node *temp1 = new node;
   node *temp2 = new node;
   node *temp3 = new node;
   int j;
   temp2 = head;
   for(j=1; temp2 != NULL; j++)
   {  temp2 = temp2->next;
   }
   if(j < pos)
   {  cout<<" linked list has only "<<j-1<<" nodes,
      So you can't delete node no. "<<pos<<endl;
      return;
   }
   temp2 = head;
```

```
    for(int i=1 ;i<pos ; i++)
    { temp1 = temp2;
      temp2 = temp2 -> next;
    }
    temp3 = temp2 -> next;
    temp1 -> next = temp3;
    temp2 -> next = NULL;
}
Void list :: traverse()
{ node *temp = new node;
  temp = head;
  while(temp != NULL)
  { cout<< temp -> data <<'\t';
    temp = temp -> next;
  }
}

int main()
{ int n, a, p;
  list var;
  char ch;
  while(1)
  { cout<<" enter 1 for insert at begin" << endl;
    cout<<" enter 2 for insert at last"<<endl;
    cout<<" enter 3 for insert at position"<<endl;
    cout<<"enter 4 for delete at begin"<<endl;
```

```
cout<<" enter 5 for delete at last"<<endl;
cout<<"enter 6 for delete at position"<<endl;
cout<<" enter 7 for ~~delete at~~ traversing"<<endl;
cin>>n;
switch(n)
  {  case 1:
     cout<<"enter number in"; cin>>a;
     var.insert_start(a); cout<<endl;
     break;
     case 2:
     cout<<"enter number"<<endl; cin>>a;
     var.insert_end(a);
     cout<<endl; break;
    case 3:
    cout<<" enter number and position"<<endl;
    cin>>a>>p; var.insertatposition(p,a);
    cout<<endl; break;
    case 4:
     var.delete_first(); cout<<endl; break;
   case 5:
     var.delete_last(); cout<<endl; break;
   case 6:
     cout<<"enter position:"<<endl; cin>>p;
     var.deleteatposition(p); cout<<endl; break;
   case 7:
     var.traverse(); cout<<endl; break;
```

```
        default:
        cout<<"option not found \n";
    }
cout<<"Put 'y' to get options otherwise 'n' \n";
cin>>ch;
if (ch=='n')
{ break;
}
}
return 0;
}
```

Output:

```
enter 1 for insert at begin
enter 2 for insert at last
enter 3 for insert at position
enter 4 for delete at begin
enter 5 for delete at last
enter 6 for delete at position
enter 7 for traversing
1
enter number
11

Put 'y' to get options otherwise 'n'
y
enter 1 for insertion at begin
enter 2 for insertion at last
enter 3 for insertion at position
enter 4 for deletion at begin
enter 5 for deletion at last
enter 6 for deletion at position
enter 7 for traversing
2
enter number.
22
```

```
Put 'y' to get option otherwise 'n'
y
enter 1 for insert at begin
enter 2 for insert at last
enter 3 for insert at position
enter 4 for delete at begin
enter 5 for delete at last
enter 6 for delete at position
enter 7 for traversing
3
enter number and position
33
3

Put 'y' to get options otherwise 'n'
y
enter 1 for insert at begin
enter 2 for insert at last
enter 3 for insert at position
enter 4 for delete at begin
enter 5 for delete at last
enter 6 for delete at position
enter 7 for traversing
7
11  22  33
Put y to get option otherwise n
```

y
enter 1 for insertion at begin
enter 2 for insert at last
enter 3 for insert at Position
enter 4 for delete at begin
enter 5 for delete at last
enter 6 for delete at Position
enter 7 for traversing
4
Put y to get options otherwise n
y
enter 1 for insert at begin
enter 2 for insert at last
enter 3 for insert at Position
enter 4 for delete at begin
enter 5 for delete at last
enter 6 for delete at Position
enter 7 for traversing.
5

Put 'y' to get options otherwise 'n'
y
enter 1 for insert at begin
enter 2 for insert at last
enter 3 for insert at Position
enter 4 for delete at begin

enter 5 for delete at last
enter 6 for delete at position
enter 7 for traversing
7
22
Put 'y' to get options otherwise 'n'
n

Topic : Program No: 11 Date : Page : 41

Aim :- Write a program to convert infix to postfix expression.

Code :-

```
#include<iostream>
#include<stack>
Using namespace std;
Void convertPostfix(char *a)
{  stack<char> s;
   char output[50], t;
   for(int i=0; a[i]!='\0'; i++)
   {  char ch = a[i];
      switch(ch)
      {  case '^':
         case '-':
         case '+':
         case '/':
         case '*': s.push(ch); break;
         case ')': t= s.pop();
                   s.pop();
                   cout<<t ; break;
      }
      if(isalpha(ch))
        cout<<ch;
   }
}
```

```
int main()
{
    char a[] = "(((a*b)+(c/d))-e)";
    convertPostfix(a);
    return 0;
}
```

output :-

ab*cd/+e-.

Topic :.Program No: 12..... Date :..... Page :.....24

Aim :- Write a program to perform Quick sort on input array.

Code :-

```
#include<iostream>
using namespace std;
void swap(int *a, int *b)
{   int t = *a;
    *a = *b;
    *b = t;
}
void Print(int array[], int size)
{   int i;
    for(i=0; i<size; i++)
    {  cout<<array[i]<<endl;
    }
}
int divide(int array[], int low, int high)
{  //int i;
   //for(i=0; i<size; i++)
    int pivot = array[high];
    int i = (low-1);
    for(int j = low; j<high; j++)
    {   if(array[j]<=pivot)
        {
```

```
            i++;
            swap(&array[i], &array[j]);
        }
    }
    print(array, 7);
    cout<<"..... \n";
    swap(&array[i+1], &array[high]);
    return(i+1);
}

void quicksort(int array[], int low, int high)
{   if(low<high)
    {   int pi = divide(array, low, high);
        quicksort(array, low, pi-1);
        quicksort(array, pi+1, high);
    }
}

int main()
{   int data[] = {8,7,6,1,0,9,2};
    int n = sizeof(data)/sizeof(data[0]);
    quicksort(data, 0, n-1);
    cout<<"Sorted array in ascending order:\n";
    print(data, n);
}
```

Output:

1 0 6 8 7 9 2
. . . . . .
1 0 2 8 7 9 6
. . . . . .
0 1 2 8 7 9 6
. . . . . .
0 1 2 6 7 9 8
. . . . . .

Sorted array in ascending order:
0 1 2 6 7 8 9