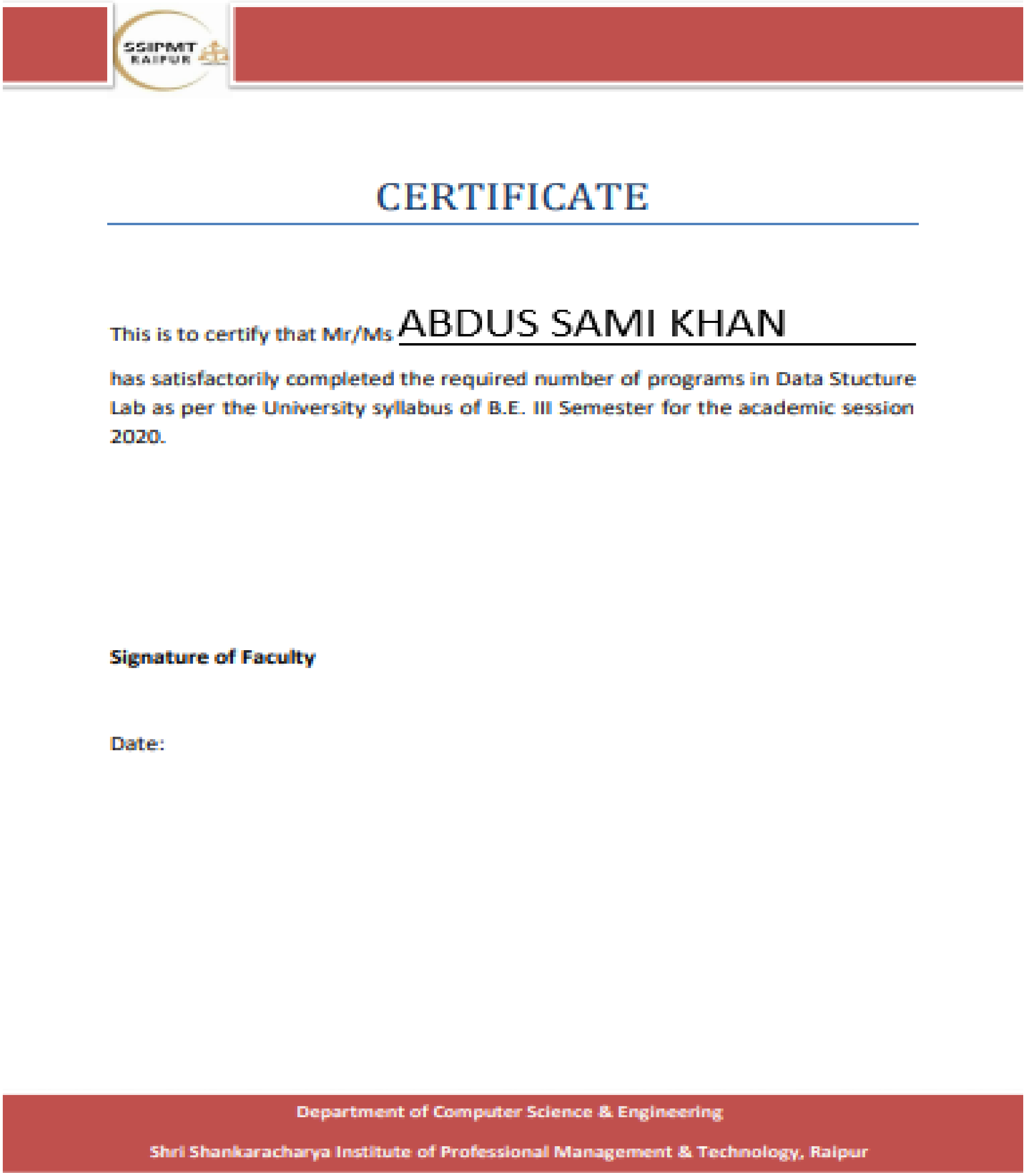
SSIPMT
RAIPUR

DATA STRUCTURE

LAB FILE

Department of Computer Science & Engineering

Shri Shankaracharya Institute of Professional Management & Technology, Raipur



SSIPMT RAIPUR

CERTIFICATE

This is to certify that Mr/Ms ABDUS SAMI KHAN

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:

Department of Computer Science & Engineering

Shri Shankaracharya Institute of Professional Management & Technology, Raipur

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.			

1

Aim:- write a program insertion on array

Code:-
```
     #include<iostream>
     using namespace std;

       int main ()
       {
          int a[20], n, x, i, pos = 0;
          count << "Enter size of array ";
          cin >> n;
      cout << "Enter array is order:\n";

        for ( i=0; i< n; ++i)
             cin >> a[i];
    cout <<"\n Enter element to insert:";
       cin >> x;
       for( i=0; i<n; ++i)
           if (a[i] <= x && x < a[i+1])
    {   pos = i+1;
               break;  }
     for ( i= n+1; i>pos ; --i)
             a[i] = a[i-1];
       a[pos] = x;
  cout<<"\n \n Array after inserting:\n";
       for (i= 0; i< n+1 ; i++)
           cout << a[i]<< " ";
    return 0;
    }
```

Output :-

```
Enter size of array : 5
Enter array in order :
    1  3  4  6  9

Enter element to insert : 2
Array after inserting :
1   2   3   4   6   9
```

Subject :   Date :

2

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

Code :-

```
# include<iostream>
# include<process.h>
using namespace std;

int main()
 {
     int a[50], x, n, i, j, b[50];
cout<<"Element of array:";
cin>>n;
   cout<<"\n Enter elements of array\n";

for (i=0; j=i<n; ++i)
       cin>>a[i];
  cout<<"\n Enter elemets to delete:";
  cin>>x;
for (i=0; j=0; i<n; ++i)
{     if (a[i]!=x)
          b[j++]=a[i]; }
if (j==n) { cout<<"\n Element not in array";
exit(0); }   else
 {    cout<<"\n New Array is";
   for (i=0; i<j; i++)
        cout<<b[i]<<" ";  }
  return 0;
 }
```

Output :-

```
Element you want to create : 5

 Enter elements of array
  14  8  3  6  9

  Enter element  to  delete : 6

 New array is  14  8  3  9
```

Subject : Date :

3

Aim:- Write a program to perform reverse of array.

Code:-

```
        #include <iostream>
        #include <algorithm>
  using namespace std;

void print (int arr [], int n)
{   for (int i = 0 ; i<n ; i++) {
      cout << arr[i] << " ";   }
  }

  void reverse (int arr [], int n)
  {   reverse (arr, arr + n);
   }
   int main ()
  {   int arr[] = {1,2,3,4, 5} ;

  int n = sizeof (arr)/sizeof(arr[0])) ;

     reverse(arr, n);
     print (arr, n) ;

     return 0;
   }
```

Output :-

```
5 4 3 2 1
```

4

<u>Aim</u>:- Write a program to perform Binary Search on linear array.

<u>Code</u>:-
```
#include <iostream>
    using namespace std;
    int main ()
      { int i, arr[10], num, first, last, middle;
 cout<<" Enter 10 elements : ";
        for (i=0 ; i<10 ; i++)
       cin >> arr [i];
 cout<<"\n Enter element : ";
      cin >> num;
      first = 0;
      last = 9;
     middle = (first + last)/2;
     while (first <= last)
   {  if (arr[middle]< num)
           first = middle +1;
     else if (arr [middle]==num)
  { cout<< "\n number, "<< num<<" pos "<< middle
      break;   }
      else
            last = middle -1;
  middle = (first+last)/2;     }
   if (first > last)
            cout<< "number, << num is not found
     cout<< endl;     return 0;
     }
```

Output :-

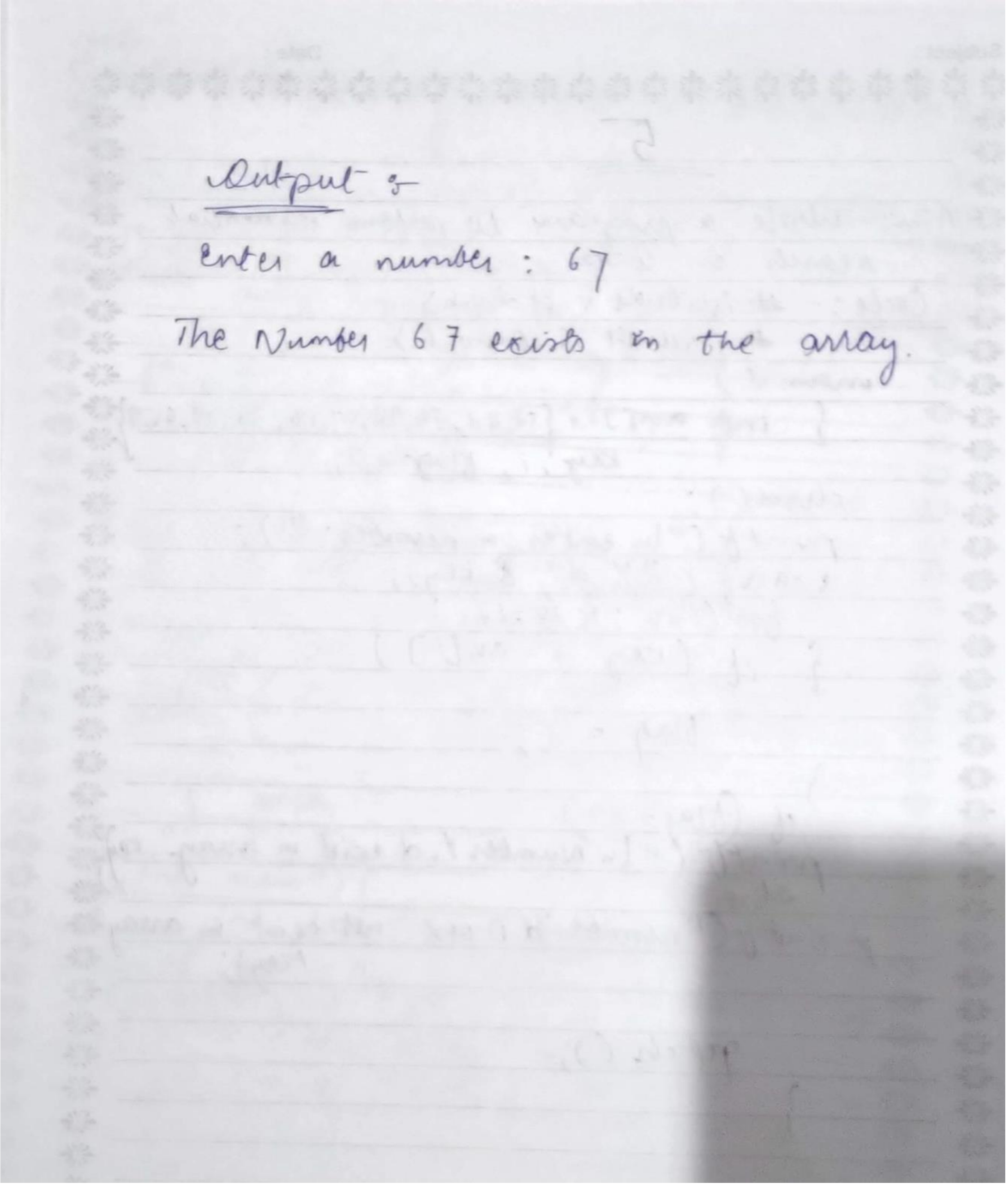
```
Enter 10 elements : 1
 2
 3
  4
  5
  6
  7
  8
  9
  10
 Enter lelement to be search : 8
The number, 8 found at position 8.
```

5

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

Code:-

```
#include < stdio.h>
#include <conio.h>
main ()
{ int arr[] = {12,23,78,98,67,56,45,19,65,9}
              key, i, flag = 0;
  clrscr();
  printf("\n enter a number: ");
  scanf("%d", &key);
   for (i=0 ; i< 10; i++)
  { if (key == arr[i] )

        flag = 1;
  }
   if (flag==1)
  printf("\n Number %d exist in array", key);
   else
 printf("number %d Does not exist in array",
                                key);

      getch();
  }
```



Output :-

```
enter a number : 67
The Number 67 exists in the array.
```

6

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

Code:-

```
#include <iostream>
     using namespace std;
   void print array(int array [], int size) {
for (int i=0; i< size; i++) {
   cout << array[i] << " ";  }
    cout << endl;
   }
void insertion sort (int array [], int size) {
  for (int step=1; step < size; step ++) {
   int Key = array [step];
   int j = step -1;
   while (Key < array [j] && j >= 0) {
  array [j+1] = array [j];
         --j;
     }  array [j+1]  = Key;
        }  }
  int  main () {
        int data [] = {9,5,1,4, 3};
    int size = size of (data)/ size of (data[0]);
insertion sort (data, size);
cout << "sorter array order:\n";
    print array (data, size);
 }
```

Output:

```
Sorted array in ascending order:
  1  3  4  5  9
```

8

Aim:- Write a program to perform selection sort on iput array.

Code:-
```
#include <iostream>
    using namespace std;
   int main()
 { int i,j,n, loc, temp, min, a[30];
 cout<<" Enter no. of elements :";
    cin>>n;
   cout<<"\n enter elements\n ";
   for (i=0; i<n; i++)
    { cin >> a[i];    }
  for (i=0; i<n-1; i++)
 {     min=a[i] ;
       loc = i;
        for (j=i+1; j<n; j++)
 { if (min> a[j])   {  min = a[j];
        loc= j; }   }
    temp = a[i];
         a[i] = a[loc]; a[loc] = temp ;
 } cout<<"\n Sorted list as follows";
   for (i=0; i<n; i++)
 {  cout<< a[i]<<" ";     }
      return 0;
    }
```

Output:-

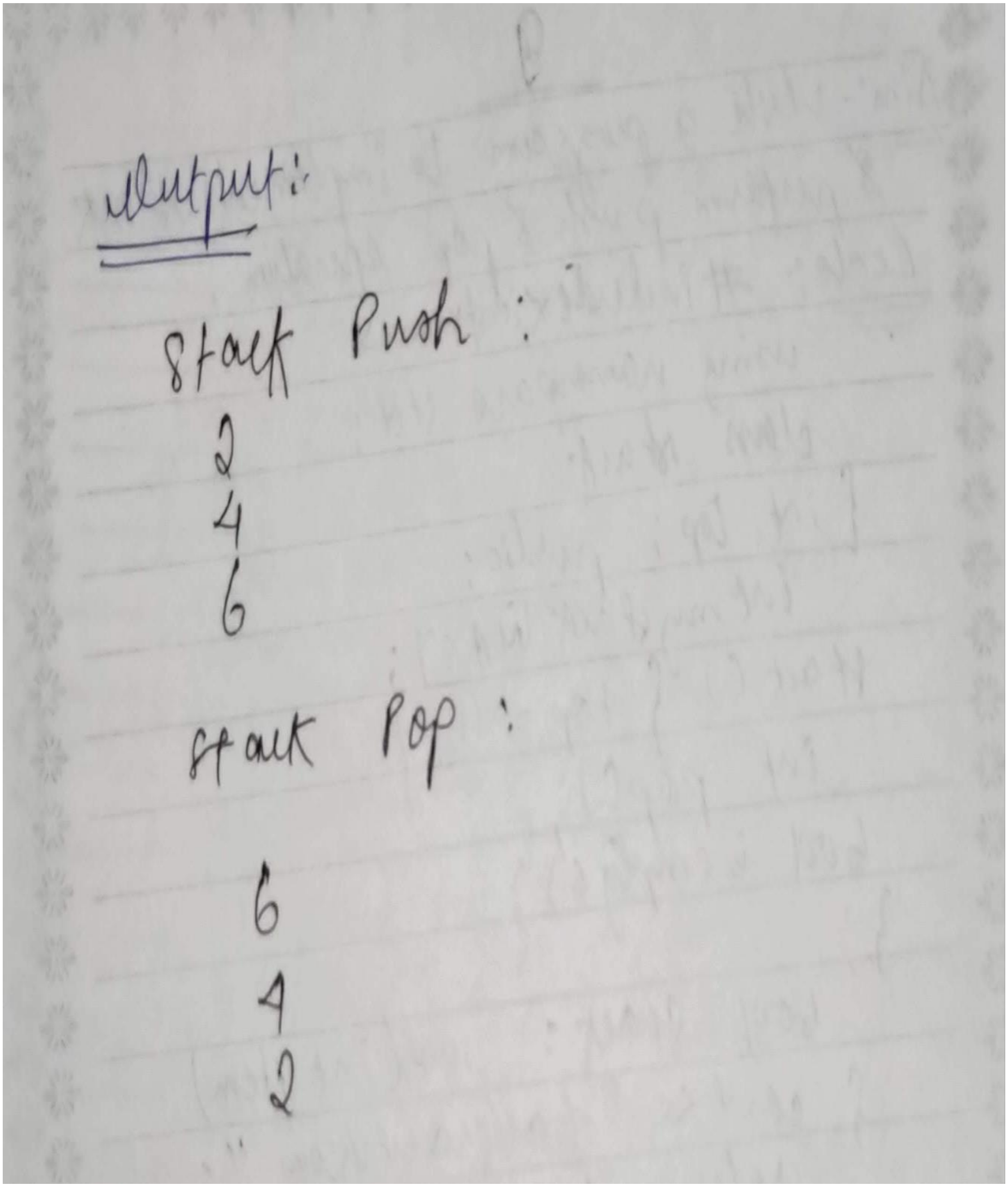
```
Enter no. of element =: 6

Enter the elements:
 18  3 10 7 8 1

Sorted list is as follows
 1  3  7  8  10  18
```

9

Aim:- Write a program to implement stack & perform push & pop operation.

Code:-

```
#include<iostream>
   using namespace std;
   class stack
  { int top; public:
     int mystack [MAX];
  stack () { top = -1; }
     int pop();
  bool isempty ();
  };
     bool stack :: push(int item)
   { cout<< "stack overflow ";
      return false;
   } else {
        cout<<item<<endl;
       return true ; } }
   { cout <<"stack underflow; return 0;
  } } }   bool stack :: isempty ( )
   { int main ()
      class stack stack;
  cout<<" Stack Push"<<endl;
  stack. push (2);
  stack. push (6);
 cout <<" stack Pop:"<< endl;
 while { cout<<stack.pop() << endl;
 } return 0;
     }
```



Output:

```
Stack Push :
  2
  4
  6

Stack Pop :

   6
   4
   2
```

Subject :   Date :

11

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

Code :-

```
# include <iostream>
# include <stack>
using namespace std;
void postfix (char *a)
{ stack <char> s;
   char output [50], t;
 { char ch = a[i];
      switch (ch)
   { case '^' :
      case '-' :
      case '+' :
      case '/' :
      case '*'   : s.push(ch):
                     break;
case ')' : t= s.top();
            s.pop();
               cout << t; break;
 } if (isalpha (ch))
           cout << ch;
     } } int main () {
 char a[] = "(((a*b) + (c/d)) -e)";
  Postfix (a);
        return 0;
    }
```

Output:

```
ab*cd/+e-
```

Subject :   Date :

12

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] << " ";
      cout << endl;  }
 for (int j = low; j < high; j++) {
   if (array[j] <= pivot) { i++;
    swap (&array [i], &array [j]); }
 } cout << "- - - -\n";
    return (i + 1);    }
void quicksort (int array [], int low,
                             int high) {
 if (low < high) {  quicksort(array, pi+1,
                               high);
} }    int main () { ind data[] = {8,7,6,1,2};
 int n = size of(data) / size of(data [0]);
     quick sort (data, 0, n-1);
cout<<" Sorted array order :\n"; print (data);
    }
```

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