Date : 13/4/2022

Lab Program 1.

Write a program to insert elements in an Array.

```
#include <stdio.h>
void main()
{
    int array[100], position, i, n, value;
printf("enter no of elements in array \n");
scanf("%d", &n);
    printf("Enter %d elements \n", n);
    for(i=0; i<n; i++)
      scanf("%d", &array[i]);
printf("enter the location where you want to
                    insert the element \n");
    scanf("%d", &position);
printf("enter the value to insert \n");
    scanf("%d", &value);
    for(i=n-1; i>=position-1; i--)
        array[i+1] = array[i];
  array[position-1] = value;
    printf("Resulted array is \n");
      for(i=0; i<=n; i++)
        printf("%d \n", array[i]);
    getch();
}
```

Output (1)

Enter no of element in array 5

Enter 5 elements 1

3

4

5

6

Enter the location you want to insert element 6

Enter the values to insert 9

The resulted array is 1

3

4

5

9

Lab Program 2

Write a program to delete an element in an array

```
#include <stdio.h>
void main()
{
    int array[100], position, c, n;
printf("Enter no of element in array \n");
scanf("%d", &n);
printf("Enter %d elements \n", n);
for (c=0; c<n; c++)
  scanf("%d", &array[c]);
printf("Enter the location you wish to delete element \n");
scanf("%d", &position);
  if (position >= n+1)
printf("Deletion not possible. \n");
else.
{
    for(c = position - 1; c<n-1; c++)
            array[c] = array[c+1];
printf("Resulted array is: \n");
for (c=0; c<n-1; c++)
    printf("%d \n", array[c]);
  }
getch();
}
```

Output (2)

```
Enter the no of element in the array
6
Enter 6 elements
6
7
8
4
3
2
Enter the location to delete the element
4
The resulted array is:
6
7
8
3
2
```

Date: 20/04/22

Lab 3

Write a program to search a number using binary Search.

```
#include <stdio.h>
#include <conio.h>
void main()
{
int A[5] = {10, 20, 30, 40, 50};
int key = 50, flag=0, low=0, high=4, mid;
while (low<=high)
{
    mid = (low+high)/2;
    If (key == A[mid])
    {
    flag=1;
    break;
    }
else if (key<A[mid])
    high = mid-1;
else
    low = mid+1;
}
if (flag == 1)
Printf( "key is found");
else
Printf ("key not found");
getch();
}
```

Output (3)

The key is found.

Perform binary Search

0	1	2	3	4	5
5	6	7	(8)	9	10
↑ Low		↑ Mid	key		↑ high

Step1: $Mid = \frac{Low + high}{2} = \frac{0+5}{2} = 2$

key = A[Mid] ; / 8 ≠ 7

key < A[Mid] / 8 < 7, false

key > A[Mid] / 8 > 7 ; True

Step2: low = mid + 1 / low = 2 + 1

low = 3

$Mid = \frac{3+5}{2} = 4$

key = A[Mid] / 8 ≠ 9

key < A[Mid] / 8 < 9 True

Step 3 high = mid + 1 / high = 3

$hmid = \frac{3+3}{2} = 3$

A[Mid] = 8 / key = A[Mid]

8 = 8 True

key found

20/04/22

20/04/22

Lab Program 4

Write a Program to find the location of a element using Linear Search.

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int a[5], i,x, n;
Printf("How many element ?");
  Scanf("%d",&n);
Printf("Enter array elements :\n");
 for(i=0; i<n; ++i)
    Scanf("%d",&a[i]);
Printf("\n Element to search :");
    Scanf("%d", &x);
 for(i=0; i<n; ++i)
    if(a[i]==x)
    break;
if(i<n)
  Printf("element found at index %d", i);
else
    Printf("element not found");
   getch();
}
```

Output (4)

```
How many elements? 5
Enter array elements:
5
6
7
8
9
Enter the element to search: 9
Element is found at index 4
```

Date: 26/4/22

Lab Program 5/ Write Program to sort the number in assending order using bubble sort.

```
#include <stdio.h>
#include <conio.h>
void main()
{
int array[100], n, i, j, swap;
Printf("Enter no of elements");
scanf("%d", &n);
Printf("Enter %d nos: ", n);
for (i=0; i<n; i++)
 scanf("%d", &array[i]);
 for (i=0; i<n-1; i++)
{
 for (j=0; j<n-i-1; j++)
 {
  If (array[j] > array[j+1])
 {
  swap = array[j];
  array[j] = array[j+1];
  array[j+1] = swap;
 }
}
 Printf("SORTED ARRAY IS: ");
  for (i=0; i<n; i++)
 Printf("%d", array[i]);
getch();
}
```

Output (5)

```
Enter no of elements : 6
Enter 6 Numbers to the array: 5
9
7
1
0
6
SORTED ARRAY IS : 0 1 5 6 7 9
```

Working of Bubble Sort

15	16	6	8	5

15 16 6 8 5

15 16 6 8 5

15 6 16 8 5

15 6 8 16 5

15 6 8 5 16

Pass 1.

15 6 8 5 16

6 15 8 5 16

6 8 15 5 16

6 8 5 15 16

pass 2

6 8 5 15 16

6 5 8 15 16

5 6 8 15 16

Pass 3

Pass 4

Date: 27/4/22

Lab 6 Write A program to sort numbers in descending order using selection sort.

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int i, j, size, a[10], min, temp;
clrscr();
Printf("Enter the size of an array \n");
scanf("%d", &size)
Printf("enter the elements into an array :\n");
for(i=0; i<size; i++)
 {
 scanf("%d", &a[i]);
 }
Printf("Array elements before sorting \n");
 for(i=0; i<size; i++)
Printf("%d \t", a[i]);
 for(i=0; i<size; i++)
{
  min = i;
 for(j=i+1; j<size; j++)
{
  if(a[j] <a[min])
    min=j;
}
```

```
        temp = a[i];
        a[i] = a[mim];
        a[mim] = temp;
    }
    Printf("\n After sorting\n");
        for (i=0; i<size; i++)
        Printf("%d \t", a[i]);
        getch();
    }
```

Output (6)

```
Enter the size of an array : 7
Enter the element into the array 5
2
0
9
7
4
1
Array element before sorting
5 2 0 9 7 4 1
After Sorting
9 7 5 4 2 1 0
```

Date: 4/05/22

Lab Program 7

Write a Program to Sort the numbers in ascending order using Insertion sort.

```
#include <stdio.h>
#include <conio.h>
  void main()
  {
    int n, i, j, a[20], temp;
    clrscr();
  Printf ("Enter the size of an array \n");
  scanf ("%d", &n);
  Printf ("\n Enter the elements to array");
    for (i=0; i<n; i++)
  scanf ("%d" &a[i];
    Printf ("Before Sorting");
      for (i=0; i<n; i++)
    Printf ("%d\t", a[i]);
      for (i=1; i<n; i++)
    {
      temp = a[i];
  for (j=i; j>0 && temp<a[j-1]; j--)
  {
    a[j] = a[j-1];
  }
```

Output (7)

Enter the size of an array 6

Enter the elements to array 1

6

9

0

3

8

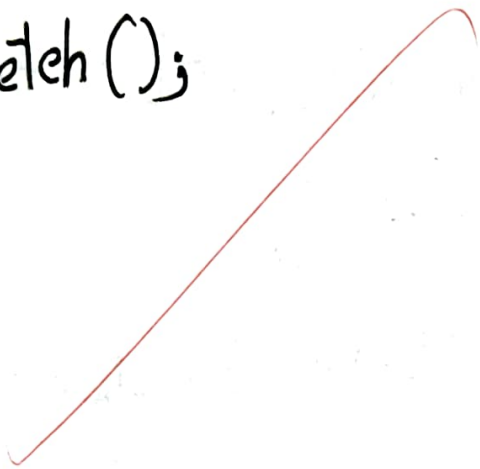
Before Sorting

1 6 9 0 3 8

After Sorting

0 1 3 6 8 9

```
        a[j] = temp;
    }
    Printf ("\n After sorting");
        for (i=0; i<n; i++)
    Printf ("%d\t", a[i]);
        getch ();
}
```



Date: 6/5/22

Lab Pro 8

Write a program to perform push, pop, display operation in stack.

```
#include <stdio.h>
#include <process.h>
#include <stdlib.h>
#define MAX 5
int top = -1, stack[MAX];
  void push();
  void pop();
  void display();
 void main()
 {
   int ch;
   while (1)
 {
Printf("\n ** Stack Menu **");
Printf("\n\n1. Push\n2. Pop\n3. display\n4.
                                    Exit");
Printf("\n\n Enter your choice (1-4):");
 scanf("%d", &ch);
 Switch(ch)

{
 case 1: push();
 break;
```

```
        case 2: pop();
         break;
        case 3: display();
         break;
        case 4: exit(0);
        default: printf("\n Wrong choice !! ");
    }
}
void push()
{
  int val;
  if (top == MAX-1)
  {
    Printf ("\n Stack full! ");
  }
  else
  Printf ("\n Enter element to push: ");
   scanf ("%d", &val);
    top = top+1;
    Stack[top] = val;
  }
  void pop()
  {
    if (top == -1)
  {
    Printf ("\n Stack is empty! ");
  } else
   { Printf ("\n Deleted element is %d", stack[top]);
     top = top-1;
  }
}
```

```
void display()
{
  int i;
  if(top==-1)
  { Printf("\n Stack is empty!");
  }
  else
  { Printf("\n Stack is...\n");
    for(i=top; i>=0; --i)
    Printf("%d\n", stack[i]);
  }
}
```

OUTPUT (8)

```
** Stack Menu **
1. Push
2. Pop
3. Display
4. exit
Enter your choice (1-4): 1
Enter element to push: 60
90
80
50
** Stack Menu **
1. Push
2. Pop
```

3. Display
4. Exit

Enter your choice (1-4): 3

Stack is....

60
90
80
50

** Stack Menu **

1. Push
2. Pop
3. display
4. Exit

Enter your choice (1-4): 2

Deleted element is 60

** Stack Menu **

1. Push
2. Pop
3. Display
4. Exit

Enter your choice (1-4): 6

Wrong choice !!

** Stack Menu **

1. Push
2. pop
3. Display
4. Exit

Enter your choice (1-4): _

Lab Prog 9

Write a program to check the given matrix is sparse or not.

```
#include <stdio.h>
int main()
{
    int i, j, rows, columns, a[10][10], total = 0;
Printf("\n Enter Number of rows and columns :");
  scanf("%d %d", &i, &j);
Printf("\n Enter Matrix elements\n");
 for (rows = 0; rows < i; rows++)
{
    for (columns = 0; columns < j; columns++)
{
    scanf("%d", &a[rows][columns]);
 }
}
 for (rows = 0; rows < i; rows++)
{   for (columns = 0; columns < j; columns++)
 {
    if (a[rows][columns] == 0)
      { total ++;
      }
 }
}
```

```
        if (total > (rows * columns)/2)
        {
            Printf ("\n The matrix that you entered is sparse
        }
        else
        {
        Printf ("\n The matrix you entered is not a Sparse Matrix")
        }
        return 0;
}
```

Output (9)

```
Enter Number of row and column : 3

3

Enter matrix elements

1
3
5
6
9
7
5
0
8

The matrix you entered is not a Sparse Matrix.
```

Program NO. 10 Write a program to evaluate a postfix expression

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

// Stack type
struct stack
{
    int top;
    unsigned capacity;
    int * array;
};

struct Stack * createStack (unsigned capacity)
{
struct Stack *stack = (struct Stack *)malloc (sizeof (struct stack
if (!stack) return NULL;
{
struct Stack * stack = createStack (strlen (exp));
int i;
if (! stack) return -1;
for (i=0; exp[i]; ++i)
{
    if (isdigit(exp[i]))
     Push (stack, exp[i]-'0');
void push (struct Stack * stack, char op)
{
    stack -> array [++ stack -> top] = op;
}
    int evaluatePostfix (char * exp)
{
```

```
int isEmpty(struct Stack * stack)
{
  return stack -> top == -1;
}
char peek (struct Stack * stack)
{
  return stack -> array[stack -> top];
}
char pop(struct Stack* stack)
{
    if(!isEmpty(stack))
          return stack -> array[stack -> top--];
      return '$';
    }
    else
    {
     int val1 = pop(stack);
      int val2 = pop ( stack);
     switch (exp[i])
     {
     case '+': push(stack, val2+ val1); break;
     case '-': push(stack, val2 - val1); break;
     case '*': push(stack, val2 * val1); break;
     case '/': push(stack, val2 /val1); break;
    }
  }
}
return pop(stack);
}
   void main()
{
   char exp[] = "53+82-*";
     clrscr();
   Printf("Postfix evaluation: %d", evaluatePostfix(exp));
   getch();
}
```

output

```
postfix evaluation : 48
```

11/5/22