Data Structures Lab
(CA3206)

Name: SHUBHANSHU SINGH

Registration ID: 2021PGCACA054

Date: 02-02-2022

Facility : Dr. D.K. Shaw

Assignment 01

1. Write a C program to perform Bubble Sort for an array using pointer.

2. Write a C program to perform Merge Sort for an array using pointer.

3. Write a C program to perform Selection Sort for an array using pointer.

4. Write a C program to perform Matrix addition using pointer.

5. Write a C program to perform Matrix Multiplications using pointer.

1. Write a C program to perform Bubble Sort for an array using pointer.

Program :

```c
#include<stdio.h>
void print_array(int n,int *a){

  for(int i=0;i<n;i++)
   printf("%d ",*(a+i));
 }

void bubble_sort(int n,int *a){

  int flag=1;

   for(int i=0; i<n&&flag==1; i++)
   {
    flag=0;
     for(int j=0; j<n-(i+1); j++)
      if (*(a+j)>*(a+j+1))
      {
        int temp=*(a+j);
       *(a+j)=*(a+j+1);
       *(a+j+1)=temp;
       flag=1;
      }
   }
 }
int main(){

 printf("Enter the number of elements you wants : ");
  int n;
  scanf("%d",&n);

 int a[n];
 printf("Eenter elements : ");
 for(int i=0;i<n;i++)
  scanf("%d",&a[i]);
```

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

 printf("Array :\n");
  for(int i=0;i<n;i++)
   printf("%d ",a[i]);

  bubble_sort(n,a);

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

 printf("\nSorted Array :\n");
  print_array(n,a);

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

return 0;
}
```

Output :

```
Enter the number of elements you wants : 10
Eenter elements : 1 4 2 9 0 2 3 2 43 0
------------------------------
Array :
1 4 2 9 0 2 3 2 43 0
------------------------------
Sorted Array :
0 0 1 2 2 2 3 4 9 43
------------------------------
```

2. Write a C program to perform Merge Sort for an array using pointer.

Program :

```c
#include<stdio.h>
void print_array(int n, int *a){

  for(int i=0; i<n; i++)
   printf("%d ",*(a+i));
 }

void merge(int *a, int lb, int mid, int ub){

    int sub_array_one=mid-lb+1, sub_array_two=ub-mid;

    int left_array[sub_array_one],right_array[sub_array_two];

   for (int i=0; i<sub_array_one; i++)
        left_array[i]=*(a+lb+i);

    for (int j = 0; j < sub_array_two; j++)
        right_array[j]=*(a+mid+1+j);

    int one=0,two=0,merged_array = lb;

    while (one < sub_array_one && two < sub_array_two)
    {
      if (left_array[one] <= right_array[two])
       {
         *(a+merged_array) = left_array[one];
          one++;
       }
      else
       {
         *(a+merged_array)= right_array[two];
          two++;
       }
        merged_array++;
    }
```

```
    while(one<sub_array_one)
    {
      *(a+merged_array)=left_array[one];
       one++;
      merged_array++;
    }
    while(two<sub_array_two)
    {
      *(a+merged_array)=right_array[two];
       two++;
      merged_array++;
    }
}

void merge_sort(int *a, int lb, int ub){

      if (lb>=ub)
        return;

    int mid=lb+(ub-lb)/2;
    merge_sort(a, lb, mid);
    merge_sort(a, mid + 1,ub);
    merge(a,lb, mid, ub);
}

int main(){

 printf("Enter the number of elements you wants : ");

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

 int a[n];
 printf("Eenter elements : ");
 for(int i=0;i<n;i++)
  scanf("%d",&a[i]);

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

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

  int lb=0,ub=n-1;
  merge_sort(a, lb, ub);

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

 printf("\nSorted Array :\n");
  print_array(n, a);

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

return 0;
}
```

Output :

```
Enter the number of elements you wants : 10
Eenter elements : 1 2 4 5 2 8 0 75 4 90
------------------------------
Array :
1 2 4 5 2 8 0 75 4 90
------------------------------
Sorted Array :
0 1 2 2 4 4 5 8 75 90
------------------------------
```

3. Write a C program to perform Selection Sort for an array using pointer.

Program :

```c
#include<stdio.h>
void print_array(int n, int *a){

  for(int i=0; i<n; i++)
   printf("%d ",*(a+i));
 }

void section_sort(int *a,int n){

  for(int i=0;i<n;i++){
    int min_index=i;
    for(int j=i+1;j<n;j++)
     if(*(a+j)<*(a+min_index))
      min_index=j;
    int temp=*(a+i);
    *(a+i)=*(a+min_index);
    *(a+min_index)=temp;
  }
}

int main(){

 printf("Enter the number of elements you wants : ");

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

 int a[n];
 printf("Eenter elements : ");
 for(int i=0;i<n;i++)
  scanf("%d",&a[i]);

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

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

  section_sort(a,n);

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

 printf("\nSorted Array :\n");
  print_array(n, a);

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

 return 0;
}
```

Output :

```
Enter the number of elements you wants : 5
Eenter elements : 6 1 0 12 8
------------------------------
Array :
6 1 0 12 8
------------------------------
Sorted Array :
0 1 6 8 12
------------------------------
```

4. Write a C program to perform Matrix addition using pointer.

Program :

```c
#include<stdio.h>
int main()
{
    int i,j,r,c;
    printf("Enter number of row and column : ");
    scanf("%d %d",&r,&c);
    int a1[r][c],a2[r][c],add[r][c];

printf("\n-------------------------------\n");
  printf("Enter first Matrix : \n");
  for(i=0;i<r;i++)
   for(j=0;j<c;j++)
   scanf("%d",&a1[i][j]);

printf("-------------------------------\n");
  printf("Enter second Matrix : \n");
  for(i=0;i<r;i++)
   for(j=0;j<c;j++)
   scanf("%d",&a2[i][j]);

  for(i=0;i<r;i++)
   for(j=0;j<c;j++)
   *(*(add+i)+j)=*(*(a1+i)+j)+*(*(a2+i)+j);

printf("-------------------------------\n");
    printf("Addition of matrices is : \n");
    for(i=0; i<r; i++)
    {
     for(j=0; j<c; j++)
      printf("%d ",*(*(add+i)+j));
     printf("\n");
    }
printf("-------------------------------");

 return 0;
}
```

Output :

```
Enter number of row and column : 2 3

--------------------------------
Enter first Matrix :
1 2 3
0 5 1
--------------------------------
Enter second Matrix :
4 5 3
1 0 2
--------------------------------
Addition of matrices is :
5 7 6
1 5 3
--------------------------------
```

6.Write a C program to perform Matrix Multiplications using pointer.

Program :

```
#include<stdio.h>
int main()
{
   int i,j,r,c;

   printf("Enter number of row and column : ");
   scanf("%d %d",&r,&c);
   int a1[r][c],a2[r][c],multiply[r][c];

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

  printf("Enter first Matrix : \n");
  for(i=0;i<r;i++)
   for(j=0;j<c;j++)
   scanf("%d",&a1[i][j]);

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

  printf("Enter second Matrix : \n");
  for(i=0;i<r;i++)
   for(j=0;j<c;j++)
   scanf("%d",&a2[i][j]);

  int sum;
    for(int i=0; i<r; ++i)
     for(int j=0; j<c; ++j)
     {
     sum = 0;
      for(int k=0; k<r; ++k)
       sum=sum + *(*(a1+i)+k)* *(*(a2+k)+j);
        *(*(multiply+i)+j)=sum;
     }

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

```
    printf("Multiplication of matrices is : \n");
    for(i=0; i<r; i++)
    {
     for(j=0; j<c; j++)
      printf("%d ",*(*(multiply+i)+j));
     printf("\n");
    }

printf("--------------------------------");
   return 0;
}
```

Output :

```
Enter number of row and column : 3 3

--------------------------------
Enter first Matrix :
1 2 1
4 5 3
0 6 4
--------------------------------
Enter second Matrix :
7 5 1
9 0 0
6 4 3
--------------------------------
Multiplication of matrices is :
31 9 4
91 32 13
78 16 12
--------------------------------
```