NAME - AKRITI CHOUDHARY

ROLL NUMBER - 2005776

SUBJECT - DSA LAB

DATE - 3/8/2021

CLASS - B14

BRANCH - CSE

Question1:WAP to find out the smallest and largest element stored in an array of n integers.

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

int main()
{

   int *arr;
   int n;

   puts("Enter the size of array \n");
   scanf("%d", &n);

   arr = (int *)(malloc(n * sizeof(int)));

   puts("Enter the elements of array \n");

   for (int i = 0; i < n; ++i)
   {
      scanf("%d", &arr[i]);
   }

   int max = arr[0];
   int min = arr[0];

   for (int i = 0; i < n; ++i)
   {
      if (max < arr[i])
      {
         max = arr[i];
      }
      if (min > arr[i])
      {
         min = arr[i];
      }
   }
   printf("Maximum value = %d \nMinimum value = %d \n", max, min);

   free(arr);
   puts("The dynamic memory has been freed");
}
```

```
PS D:\KIIT_NOTES\2nd year sem_3\dsa_lab\3_8_2021> ./LargestAndSmallest
Enter the size of array

10
Enter the elements of array

78 65 56 4 98 098 100 65 7 23
Maximum value = 100
Minimum value = 4
The dynamic memory has been freed
PS D:\KIIT_NOTES\2nd year sem_3\dsa_lab\3_8_2021>
```

Question2: WAP to reverse the contents of an array of n elements.

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

int main()
{

   int *arr;
   int n;

   puts("Enter the size of array \n");
   scanf("%d", &n);

   arr = (int *)(malloc(n * sizeof(int)));

   puts("Enter the elements of array \n");

   for (int i = 0; i < n; ++i)
   {
      scanf("%d", &arr[i]);
   }
   int temp = 0;
   for(int i = 0 ; i < n/2 ; ++i){
      temp = arr[i];
      arr[i] = arr[n - i - 1];
      arr[n - i - 1] = temp;
   }
   puts("The reversed array :");
   for (int i = 0; i < n; ++i)
   {
      printf("%d ", arr[i]);
   }
   puts("");
   free(arr);
   puts("The dynamic memory has been freed");

}
```

```
PS D:\KIIT_NOTES\2nd year sem_3\dsa_lab\3_8_2021> ./reverse
Enter the size of array

7
Enter the elements of array

1 2 3 4 5 6 7
The reversed array :
7 6 5 4 3 2 1
The dynamic memory has been freed
PS D:\KIIT_NOTES\2nd year sem_3\dsa_lab\3_8_2021>
```

Question3: WAP to search an element in an array of n numbers.

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

int main()
{

   int *arr;
   int n;

   puts("Enter the size of array ");
   scanf("%d", &n);

   arr = (int *)(malloc(n * sizeof(int)));

   puts("Enter the elements of array ");

   for (int i = 0; i < n; ++i)
   {
      scanf("%d", &arr[i]);
   }

   int key;
   puts("Enter the value to search for :");
   scanf("%d" ,&key);

   for (int i = 0; i < n; ++i)
   {
      if(arr[i] == key){
         printf("%d was found at index %d \n" ,key , i + 1);
      }
   }

   free(arr);
   puts("The dynamic memory has been freed");
}
```

```
PS D:\KIIT_NOTES\2nd year sem_3\dsa_lab\3_8_2021> ./linearSearch
Enter the size of array
10
Enter the elements of array
23 45 1 09 23 100 87 543 2 7
Enter the value to search for :
543
The 543 was found at index 8
The dynamic memory has been freed
PS D:\KIIT_NOTES\2nd year sem_3\dsa_lab\3_8_2021>
```

Question4: WAP to sort an array of n numbers.

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

int main()
{

   int *arr;
   int n;

   puts("Enter the size of array ");
   scanf("%d", &n);

   arr = (int *)(malloc(n * sizeof(int)));

   puts("Enter the elements of array ");

   for (int i = 0; i < n; ++i)
   {
      scanf("%d", &arr[i]);
   }
   //Sorting
   int temp = 0;
   for(int i = 0 ; i < n ; ++i){
      for(int j = 0; j < n - i -1 ; ++j){
         if(arr[j] > arr[j+1]){

            temp = arr[j];
            arr[j] = arr[j+1];
            arr[j+1] = temp;

         }
      }
   }
   puts("The sorted array in ascending order :");
   for (int i = 0; i < n; ++i)
   {
      printf("%d ", arr[i]);
   }
   puts("");
   free(arr);
   puts("The dynamic memory has been freed");

}
```

```
PS D:\KIIT_NOTES\2nd year sem_3\dsa_lab\3_8_2021> ./bubbleSort
Enter the size of array
5
Enter the elements of array
90 5 45 76 2
The sorted array in ascending order :
2 5 45 76 90
The dynamic memory has been freed
PS D:\KIIT_NOTES\2nd year sem_3\dsa_lab\3_8_2021>
```

Question5: Given an unsorted array of size n, WAP to find number of elements between two elements a and b (both inclusive).
Input : arr = [1, 2, 2, 7, 5, 4]
a=2 b=5
Output : 4

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

int main()
{
        int n;
   int i;
   int x,y;
   int index_x=0,index_y=0;

        printf("Enter the size of the array :\n");
        scanf("%d",&n);

        int *arr=(int*)malloc(n*sizeof(int));

        puts("Enter the elements into the array :");

        for(i=0;i<n;i++)
        scanf("%d",&arr[i]);

        puts("Enter the number x :");
        scanf("%d",&x);

        puts("Enter the number y :");
        scanf("%d",&y);

        for(i=0;i<n;i++)
        {
                if(arr[i]==x)
                {
                        index_x=i;
                        break;
                }
        }

        for(i=n-1;i>=0;i--)
        {
                if(arr[i]==y)
                {
                        index_y=i;
                        break;
                }
        }

        printf("Number of integers from x to y : %d \n",((index_y - index_x) + 1));
```

```
    free(arr);
    puts("The dynamic memory has been freed");

        return 0;


}
```

```
PS D:\KIIT_NOTES\2nd year sem_3\dsa_lab\3_8_2021> ./q5
Enter the size of the array :
10
Enter the elements into the array :
3 4 4 5 8 9 9 9 6 7
Enter the number a :
4
Enter the number b :
9
Number of integers from a to b : 7
The dynamic memory has been freed
PS D:\KIIT_NOTES\2nd year sem_3\dsa_lab\3_8_2021>
```

Question6: Given an array, WAP to print the next greater element (NGE) for every element. The Next greater Element for an element x is the first greater element on the right side of x in array. Elements for which no greater element exist, consider next greater element as -1.
Sample Input & Output
For the input array [2, 5, 3, 9, 7}, the next greater elements for each element are as follows.
Element NGE
2 5
5 9
3 9
7 -1

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

int main()
{
```

```
    int *arr;
    int n;

    puts("Enter the size of array ");
    scanf("%d", &n);

    arr = (int *)(malloc(n * sizeof(int)));

    puts("Enter the elements of array ");

    for (int i = 0; i < n; ++i)
    {
        scanf("%d", &arr[i]);
    }

    puts("Displaying the next greater element :");
    printf("element\tNGE\n");
    for (int i = 0; i < n; ++i)
    {
        for (int j = i; j < n; ++j)
        {
            if (arr[i] < arr[j])
            {
                printf("%d\t%d\n", arr[i], arr[j]);
                break;
            }
            if(j == n-1){
                printf("%d\t%d\n", arr[i], -1);
                }
        }
    }
    puts("");
    free(arr);
    puts("The dynamic memory has been freed");
}
```

```
PS D:\KIIT_NOTES\2nd year sem_3\dsa_lab\3_8_2021> ./NGE
Enter the size of array
5
Enter the elements of array
2 5 1 7 9
Displaying the next greater element :
element NGE
2       5
5       7
1       7
7       9
9       -1

The dynamic memory has been freed
PS D:\KIIT_NOTES\2nd year sem_3\dsa_lab\3_8_2021>
```

Question7: WAP to swap three numbers in cyclic order using Call by Reference. In other words, WAP that takes three variable (a, b, c) in as separate parameters and rotates the values stored so that value a goes to be, b, to c and c to a.

```
#include <stdio.h>

void swap(int *x , int *y){
   int temp;
   temp = *x;
   *x = *y;
   *y = temp;
}

int main(){
   int a , b , c;

   puts("Enter the three values ");
   scanf("%d%d%d" ,&a ,&b ,&c);

   swap(&a ,&b);
   swap(&a ,&c);

   puts("The three swaped values ");
   printf("a = %d \nb = %d \nc = %d \n",a , b ,c);
}
```

```
PS D:\KIIT_NOTES\2nd year sem_3\dsa_lab\3_8_2021> ./threeSwap
Enter the three values
5 10 15
The three swaped values
a = 15
b = 5
c = 10
PS D:\KIIT_NOTES\2nd year sem_3\dsa_lab\3_8_2021>
```

Question8:Let A be n*n square matrix array. WAP by using appropriate user defined functions for the following:
a) Find the number of nonzero elements in A
b) Find the sum of the elements above the leading diagonal.
c) Display the elements below the minor diagonal.
d) Find the product of the diagonal elements

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

int countNonZero(int *s[], int size)
{
   int count = 0;
   for (int i = 0; i < size; ++i)
   {
      for (int j = 0; j < size; ++j)
      {
         if (s[i][j] != 0)
         {
            count++;
         }
      }
   }
   return count;
}
int SumAboveLeadingDiagonal(int *s[], int size)
{
   int sum = 0;
   for (int i = 0; i < size; ++i)
   {
      for (int j = i + 1; j < size; ++j)
      {
         sum += s[i][j];
      }
   }
   return sum;
}
void DisplayBelowMinorDiagonal(int *s[], int size)
{
   int sum = 0;
   for (int i = 1; i < size; ++i)
   {
      for (int j = size - i; j < size; ++j)
      {
         printf("%d ", s[i][j]);
      }
      puts(" ");
   }
}
void ProductOfDiagonal(int *s[], int size)
{
   int leadingPro = 1;
   int MinorPro = 1;
   for (int i = 0; i < size; ++i)
   {
      for (int j = 0; j < size; ++j)
```

```
      {
         if (i == j)
         {
            leadingPro *= s[i][j];
         }
         if (i == (size - j - 1))
         {
            MinorPro *= s[i][j];
         }
      }
   }
   printf("Product of leading diagonal =  %d\n", leadingPro);
   printf("Product of Minor diagonal =  %d\n", MinorPro);
}

int main()
{

   puts("Enter the number of elements in a square matrix");
   int n;
   scanf("%d", &n);

   int **arr = (int **)malloc(n * sizeof(int *));
   for (int i = 0; i < n; i++)
      arr[i] = (int *)malloc(n * sizeof(int));

   puts("Enter the elements in the square matrix");
   for (int i = 0; i < n; ++i)
   {
      for (int j = 0; j < n; ++j)
      {
         scanf("%d", &arr[i][j]);
      }
   }
   printf("Menu: \n 1 : Number of non zero elements in the matrix \n 2 : Sum of the elements above the
leading diagonal \n 3 :Elements below the minor diagonal\n 4: Product of diagonals\n");
   int ch;
   printf("Enter choice :\n");
   scanf("%d", &ch);
   switch (ch)
   {
   case 1:
      printf("Number of non zero elements in the matrix : %d \n", countNonZero(arr, n));
      break;
   case 2:
      printf("Sum of the elements above the leading diagonal : %d \n", SumAboveLeadingDiagonal(arr,
n));
      break;
   case 3:
      printf("Elements below the minor diagonal \n");
      DisplayBelowMinorDiagonal(arr, n);
      break;
   case 4:
      printf("Product of diagonals: \n");
      ProductOfDiagonal(arr, n);
      break;
   default:
      printf("The choice is invalid \n");
```

```
        break;
    }

    free(arr);
    return 0;
}
```

```
Enter the number of elements in a square matrix
3
Enter the elements in the square matrix
1 2 3
4 5 6
7 8 9
Menu:
 1 : Number of non zero elements in the matrix
 2 : Sum of the elements above the leading diagonal
 3 :Elements below the minor diagonal
 4: Product of diagonals
Enter choice :
1
Number of non zero elements in the matrix : 9
PS D:\KIIT_NOTES\2nd year sem_3\dsa_lab\3_8_2021>
```

```
Enter the number of elements in a square matrix
3
Enter the elements in the square matrix
1 2 3
4 5 6
7 8 9
Menu:
 1 : Number of non zero elements in the matrix
 2 : Sum of the elements above the leading diagonal
 3 :Elements below the minor diagonal
 4: Product of diagonals
Enter choice :
2
Sum of the elements above the leading diagonal : 11
PS D:\KIIT_NOTES\2nd year sem_3\dsa_lab\3_8_2021>
```

```
Enter the elements in the square matrix
1 2 3
4 5 6
7 8 9
Menu:
 1 : Number of non zero elements in the matrix
 2 : Sum of the elements above the leading diagonal
 3 :Elements below the minor diagonal
 4: Product of diagonals
Enter choice :
3
Elements below the minor diagonal
6
8 9
PS D:\KIIT_NOTES\2nd year sem_3\dsa_lab\3_8_2021>
```

```
Enter the number of elements in a square matrix
3
Enter the elements in the square matrix
1 2 3
4 5 6
7 8 9
Menu:
 1 : Number of non zero elements in the matrix
 2 : Sum of the elements above the leading diagonal
 3 :Elements below the minor diagonal
 4: Product of diagonals
Enter choice :
4
Product of diagonals:
Product of leading diagonal =  45
Product of Minor diagonal =  105
PS D:\KIIT_NOTES\2nd year sem_3\dsa_lab\3_8_2021>
```