

Ahsanullah University of Science and Technology  
Department of Computer Science and Engineering  
Examination: Assignment 5 Session: Spring 2021  
Year/Semester: 1/1 (C1) Course No: CSE1102  
Course Title: Elementary Structured Programming Lab  
ID NO:20210104111

1.WAP to insert an element into a position of an array. The element and the insertion point are inputs from the user.

Solution:

```
#include <stdio.h>

int main()
{
    int size, pos, i, num;
    printf("Enter any size of array :\n");
    scanf("%d", &size);
    int arr[size], arr_2[size + 1];
    printf("Enter the number and the position of the element:\n");
    scanf("%d %d", &num, &pos);
    printf("Enter the array:\n");
    for (i = 0; i < size; i++)
    {
        scanf("%d", &arr[i]);
    }
    for (i = 0; i < size; i++)
    {
        printf("%d", arr[i]);
    }
    printf("\n");
    for (i = 0; i <= size; i++)
    {
        if (i < (pos - 1))
        {
            arr_2[i] = arr[i];
        }
        else if (i == (pos - 1))
        {
            arr_2[i] = num;
        }
        else
        {
            arr_2[i] = arr[i - 1];
        }
    }
}
```

```

    }
    for (i = 0; i <= size; i++)
    {
        printf("%d", arr_2[i]);
    }

    return 0;
}

```

1. Find k-th maximum and k-th minimum from an array, where k is the user input.

Solution:

```

#include <stdio.h>

int main()
{
    int size, i, j, m, k;
    scanf("%d", &size);
    int arr[size];
    for (i = 0; i < size; i++)
    {
        scanf("%d", &arr[i]);
    }
    scanf("%d", &k);
    for (i = 0; i < size; i++)
    {
        for (j = i; j < size; j++)
        {
            if (arr[i] < arr[j])
            {
                m = arr[i];
                arr[i] = arr[j];
                arr[j] = m;
            }
        }
    }
    for (i = 0; i < size; i++)
    {
        printf("%d ", arr[i]);
    }
    printf("\nmaximum = %d\nminimum = %d", arr[k - 1], arr[size - k]);
    return 0;
}

```

2. 3. WAP to reverse an array of floats.

Solution:

```
#include <stdio.h>

int main()
{int n,i,j,k;
float m;
scanf("%d",&n);
float array[n];
for(i=0;i<n;i++)
{
    scanf("%f",&array[i]);
}
for(i=0,j=n-1;i<=n/2;i++,j--)
{
    m=array[i];
    array[i]=array[j];
    array[j]=m;
}
for(i=0;i<n;i++)
{
    printf("%f ",array[i]);
}

return 0;
}
```

3. WAP to rotate an integer array towards right by the given number of times.

Solution:

```
#include<stdio.h>

int main()
{
    int size,i,j,m,k,p,s=0;
    scanf("%d %d",&size,&k);
    p=size-k;
    m=k;
    int array[size],a[size];
    for(i=0;i<size;i++)
    {
        scanf("%d",&array[i]);
    }
    for(i=0;i<size;i++)
```

```

{
    if(i<k)
    {
        a[i]=array[p];
        p++;
    }
    else if(i>=k&& i<2*k)
    {
        a[i]=array[s];
        s++;
    }
    else
    {
        a[i]=array[m];
        m++;
    }
}
for(i=0;i<size;i++)
{
    printf("%d ",a[i]);
}
return 0;
}

```

4. WAP to delete all elements in between two specified positions.

Solution:

```

#include <stdio.h>
int main()
{
    int size, i, p, r1, r2, m;
    scanf("%d", &size);
    int a[size];
    for (i = 0; i < size; i++)
    {
        scanf("%d", &a[i]);
    }
    scanf("%d%d", &r1, &r2);
    p = r1;
    int b[size - (r2 - r1) + 1];
    for (i = 0; i < size; i++)
    {

```

```

        if (i < r1)
        {
            b[i] = a[i];
        }
        else if (i >= (r2 - 1))
        {
            b[p] = a[i];
            p++;
        }
    }
    for (i = 0; i < (size - (r2 - r1) + 1); i++)
    {
        printf("%d ", b[i]);
    }

    return 0;
}

```

5. WAP to search the first occurrence of a given sub array within another array.

Solution:

```

#include <stdio.h>
int main()
{
    int n, m, c = 0, i = 0, j;
    scanf("%d %d", &n, &m);
    int a[n], b[m];
    for (i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }
    for (i = 0; i < m; i++)
    {
        scanf("%d", &b[i]);
    }
    i = 0;
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < m; j++)
        {
            if (a[j] == b[i])
            {
                i++;
                c++;
            }
        }
    }
}

```

```

        if (c == m)
        {
            printf("%d", j - m + 1);
        }
    }
}
return 0;
}

```

6. WAP to find out the sum of non diagonal elements of a two dimensional array.

Solution:

```

#include <stdio.h>

int main()
{int n,i,j,sum=0,p=0;
scanf("%d",&n);
int a[n][n];
for(i=0;i<n;i++)
{
    for(j=0;j<n;j++)
    {
        scanf("%d",&a[i][j]);
    }
}
for(i=0;i<n;i++)
{
    for(j=0;j<n;j++)
    {
        if((i!=j)&&((n-1-i)!=j)&&((n-1-j)!=i)))
        {
            sum+=a[i][j];
        }
    }
}
printf("%d",sum);

return 0;
}

```

7. WAP to display the upper and lower triangle of a matrix.

Solution:

```
#include <stdio.h>

int main()
{
    int n, i, j, sum = 0, p = 0;
    scanf("%d", &n);
    int a[n][n];
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
        {
            scanf("%d", &a[i][j]);
        }
    }
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
        {
            if ((i != j) && ((n - 1 - i) != j) && ((n - 1 - j) != i))
            {
                sum += a[i][j];
            }
        }
    }
    printf("%d", sum);

    return 0;
}
```

10. Using a two dimensional array  $A[n \times n]$ , WAP to prepare a one dimensional array  $B[n^2]$  that will have all the elements of  $A$  as if they are stored in column major form.

Solution:

```
#include <stdio.h>

int main()
{
    int n, i, j, p = 0;
    scanf("%d", &n);
    int a[n][n], b[n * n];
    for (i = 0; i < n; i++)
```

```
{  
    for (j = 0; j < n; j++)  
    {  
        scanf("%d", &a[i][j]);  
    }  
}  
for (i = 0; i < n; i++)  
{  
    for (j = 0; j < n; j++)  
    {  
        b[p] = a[j][i];  
        p++;  
    }  
}  
for (i = 0; i < n * n; i++)  
{  
    printf("%d", b[i]);  
}  
  
return 0;  
}
```