

**Shri Ramdeobaba College of Engineering and Management**  
**Department of Electronics and Computer Science Engineering**

---

- 6) Initialize array and apply it to solve 1D and 2D array problems.
- Write a C program to obtain transpose of a 3 x 5 matrix.
  - Write a C program to multiply two matrices whose elements as well as number of rows and columns are entered through keyboard.
  - Write a C program to implement Selection Sort on 10 numbers entered through keyboard.

```
a) #include <stdio.h>
int main()
{
    int i, j, transpose[5][3];
    int matrix[3][5]= {
        1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
    };
    for (i= 0; i < 3; i++)
    {
        for (j = 0; j < 5; j++)
            transpose[j][i] = matrix[i][j];
    }
    printf("Entered matrix is:\n");
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 5; j++)
            printf("%d\t", matrix[i][j]);
        printf("\n");
    }
    printf("Transpose of the matrix is:\n");
    for (i = 0; i < 5; i++)
    {
        for (j = 0; j < 3; j++)
            printf("%d\t", transpose[i][j]);
        printf("\n");
    }
    return 0;
}
```

```
∴ b) #include<stdio.h>
int main()
{
    int m, n, p, q, i, j, k;
    int a[10][10], b[10][10], res[10][10];
    printf("Enter the order of first matrix\n");
    scanf("%d%d", & m, & n);
    printf("Enter the order of second matrix\n");
    scanf("%d%d", & p, & q);
    if (n != p)
    {
        printf("Matrix is incompatible for multiplication\n");
    }
    else
```

```
{
printf("Enter the elements of Matrix-A:\n");
for (i = 0; i < m; i++)
{
    for (j = 0; j < n; j++)
    {
        scanf("%d", & a[i][j]);
    }
}
printf("Enter the elements of Matrix-B:\n");
for (i = 0; i < p; i++)
{
    for (j = 0; j < q; j++)
    {
        scanf("%d", & b[i][j]);
    }
}
for (i = 0; i < m; i++)
{
    for (j = 0; j < q; j++)
    {
        res[i][j] = 0;
        for (k = 0; k < p; k++)
        {
            res[i][j] += a[i][k] * b[k][j];
        }
    }
}

printf("The product of the two matrices is:-\n");
for (i = 0; i < m; i++)
{
    for (j = 0; j < q; j++)
    {
        printf("%d\t", res[i][j]);
    }
    printf("\n");
}
}
return 0;
}
```

c) Set the first element of the array as minimum.

Minimum = 20

Compare the minimum with the next element, if it is smaller than minimum assign this element as minimum. Do this till the end of the array.

Comparing with 12 :  $20 > 12$  , minimum = 12

Comparing with 23 :  $12 < 23$  , minimum = 12

Comparing with 55 :  $12 < 55$  , minimum = 12

Comparing with 21 :  $12 < 21$  , minimum = 12

Place the minimum at the first position( index 0) of the array.

**Shri Ramdeobaba College of Engineering and Management**  
**Department of Electronics and Computer Science Engineering**

---

Array = {12, 20, 23, 55, 21}

for the next iteration, start sorting from the first unsorted element i.e. the element next to where the minimum is placed.

Array = {12, 20, 23, 55, 21}

Searching starts from 20, next element where minimum is placed.

Iteration 2 :

Minimum = 20

Comparing with 23 :  $20 < 23$  , minimum = 20

Comparing with 55 :  $20 < 55$  , minimum = 20

Comparing with 21 :  $20 < 21$  , minimum = 20

Minimum in place no change,

Array = {12, 20, 23, 55, 21}

Iteration 3 :

Minimum = 23.

Comparing with 55 :  $23 < 55$  , minimum = 23

Comparing with 21 :  $23 > 21$  , minimum = 21

Minimum is moved to index = 2

Array = {12, 20, 21, 55, 23}

Iteration 4 :

Minimum = 55

Comparing with 23 :  $23 < 55$  , minimum = 23

Minimum is moved to index 3 Array = {12, 20, 21, 23, 55}

```
#include <stdio.h>
int main()
{
    int arr[10]={6,12,0,18,11,99,55,45,34,2};
    int n=10;
    int i, j, position, swap;
    for (i = 0; i < (n - 1); i++)
    {
        position = i;
        for (j = i + 1; j < n; j++)
        {
            if (arr[position] > arr[j])
                position = j;
        }
        if (position != i)
        {
            swap = arr[i];
            arr[i] = arr[position];
            arr[position] = swap;
        }
    }
}
```

**Shri Ramdeobaba College of Engineering and Management**  
**Department of Electronics and Computer Science Engineering**

---

```
for (i = 0; i < n; i++)
printf("%d\t", arr[i]);
return 0;
}

/**
 * C program to create Simple Calculator using switch case
 */

#include <stdio.h>

int main()
{
    char op;
    float num1, num2, result=0.0f;

    /* Print welcome message */
    printf("WELCOME TO SIMPLE CALCULATOR\n");
    printf("-----\n");
    printf("Enter [number 1] [+ - * /] [number 2]\n");

    /* Input two number and operator from user */
    scanf("%f %c %f", &num1, &op, &num2);

    /* Switch the value and perform action based on operator*/
    switch(op)
    {
        case '+':
            result = num1 + num2;
            break;

        case '-':
            result = num1 - num2;
            break;

        case '*':
            result = num1 * num2;
            break;

        case '/':
            result = num1 / num2;
            break;

        default:
            printf("Invalid operator");
    }

    /* Prints the result */
    printf("%.2f %c %.2f = %.2f", num1, op, num2, result);

    return 0;
}
```