Program 1: Matrix Multiplication without Functions

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

int main() {
   int a[10][10], b[10][10], mul[10][10] = {0};
   int r, c, i, j, k;

   // Input dimensions
   printf("Enter the number of rows and columns: ");
   scanf("%d %d", &r, &c);

   // Input first matrix
   printf("Enter elements of the first matrix:\n");
   for (i = 0; i < r; i++) {
      for (j = 0; j < c; j++) {
         scanf("%d", &a[i][j]);
      }
   }

   // Input second matrix
   printf("Enter elements of the second matrix:\n");
   for (i = 0; i < r; i++) {
      for (j = 0; j < c; j++) {
         scanf("%d", &b[i][j]);
      }
   }

   // Multiply matrices
   for (i = 0; i < r; i++) {
      for (j = 0; j < c; j++) {
         for (k = 0; k < c; k++) {
            mul[i][j] += a[i][k] * b[k][j];
         }
      }
   }

   // Print the result
   printf("Resultant Matrix:\n");
   for (i = 0; i < r; i++) {
      for (j = 0; j < c; j++) {
         printf("%d\t", mul[i][j]);
      }
      printf("\n");
   }
   return 0;
}
```

Program 2: Program to perform Matrix Multiplication using Function

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

// Function to read a matrix
void readMatrix(int matrix[10][10], int rows, int cols) {
   for (int i = 0; i < rows; i++) {
      for (int j = 0; j < cols; j++) {
         scanf("%d", &matrix[i][j]);
      }
   }
}

// Function to multiply two matrices
void multiplyMatrix(int a[10][10], int b[10][10], int result[10][10], int r1, int c1, int c2) {
   // Initialize result matrix
   for (int i = 0; i < r1; i++) {
      for (int j = 0; j < c2; j++) {
         result[i][j] = 0;
      }
   }

   // Perform multiplication
   for (int i = 0; i < r1; i++) {
      for (int j = 0; j < c2; j++) {
         for (int k = 0; k < c1; k++) {
            result[i][j] += a[i][k] * b[k][j];
         }
      }
   }
}

// Function to display a matrix
void displayMatrix(int matrix[10][10], int rows, int cols) {
   for (int i = 0; i < rows; i++) {
      for (int j = 0; j < cols; j++) {
         printf("%d\t", matrix[i][j]);
      }
      printf("\n");
   }
}
```

```
int main() {
   int a[10][10], b[10][10], result[10][10];
   int r1, c1, r2, c2;

   // Input dimensions of the first matrix
   printf("Enter the number of rows and columns for the first matrix: ");
   scanf("%d %d", &r1, &c1);

   // Input dimensions of the second matrix
   printf("Enter the number of rows and columns for the second matrix: ");
   scanf("%d %d", &r2, &c2);

   // Check if multiplication is possible
   if (c1 != r2) {
      printf("Matrix multiplication not possible. Columns of the first matrix must be equal to rows of
the second matrix.\n");
      return 1;
   }

   // Input matrices
   printf("Enter elements of the first matrix:\n");
   readMatrix(a, r1, c1);
   printf("Enter elements of the second matrix:\n");
   readMatrix(b, r2, c2);

   // Perform matrix multiplication
   multiplyMatrix(a, b, result, r1, c1, c2);

   // Display the result
   printf("Resultant Matrix:\n");
   displayMatrix(result, r1, c2);

   return 0;
}
```

Program 3: Write a program in C to find minimum number in an array.

```
#include<stdio.h>
#include<conio.h>
int main()
{
int array[5]={33,30,34,31,32},j;
printf("\n array elements are:\t");
for(j=0;j<5;j++)
{
    printf("%d\t",array[j]);
}
int min=array[0];
for(j=1;j<5;j++)
{
    if(array[j]<min)
    {
    min=array[j];
    }
}
printf("\n smallest number in 5-element integer array
                                                is:\t%d",min);
return 1;
}
```

Sets the minimum number is the first element of array.

Loop will continue 5 times

Finds smallest element and store it in **min** variable.

Output

```
C:\TURBOC3\sl1.exe

array elements are:     33      30      34      31      32
smallest number in 5-element integer array is: 30
```

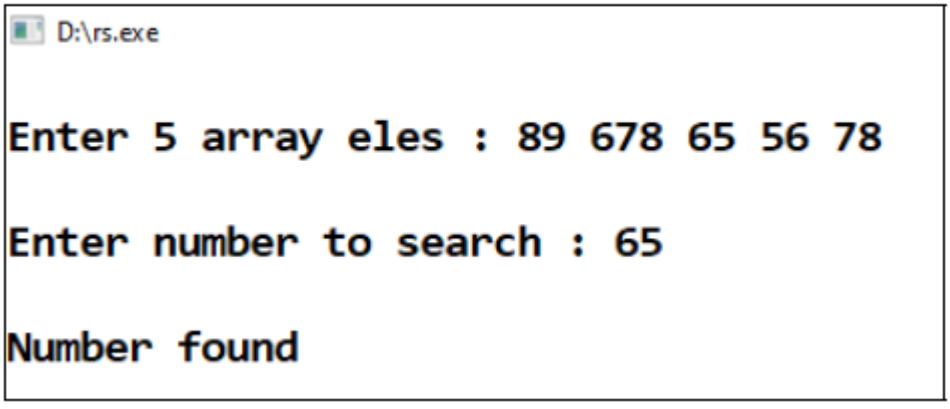
Program 4: Write a program to search a number within the array.

```
int main()
{
 int arr[5],num,i;
 printf("\nEnter 5 array eles : ");
 for(i=0;i<5;i++)
 {
    scanf("%d",&arr[i]);
 }
 printf("\nEnter number to search : ");
 scanf("%d",&num);

 for(i=0;i<5;i++)
 {
 if(num==arr[i])
 {
    printf("\nNumber found");
    break;
 }
}

if(i==5)
printf("\nNumber not found");
}
```

Output



Program 5: **Write a program in C to accept an ARRAY A with n elements and Separate it into two different arrays B and C in such a way that B contains Odd numbers and C contains Even numbers.** **i.e. if ARRAY A contains A = {3,2,4,2,5,7,8} then B = {3,5,7} and C = {2,4,2,8}.**

```
#include <stdio.h>
  void main()
  {

    long int ARR[10], OAR[10], EAR[10];
    int i, j = 0, k = 0, n;

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

    printf("Enter the elements of the array : ");
    for (i = 0; i < n; i++)
    {
      scanf("%d", &ARR[i]);

    }


    for (i = 0; i < n; i++)
    {
      if (ARR[i] % 2 == 0)
      {
        EAR[j] = ARR[i];
        j++;
      }
      else
      {
        OAR[k] = ARR[i];
        k++;
      }
    }

    printf("\nThe elements of OAR are : ");
    for (i = 0; i < k; i++)
    {
      printf("%d  ", OAR[i]);
    }

    printf("\nThe elements of EAR are : ");
    for (i = 0; i < j; i++)
    {
      printf("%d  ", EAR[i]);
    }

  }
```

Output

```
D:\rs.exe
Enter the size of array   : 5
Enter the elements of the array : 1 2 3 4 5

The elements of OAR are : 1  3  5
The elements of EAR are : 2  4
```

Program 6: Write a program which will accept 2 dimensional square matrix and find out transpose of it. Program should not make use of another matrix.

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

int main()
{
int matrix[2][2],i=0,j=0;
for (i=0;i<2;i++)
{
for(j=0;j<2;j++)
{
printf("Enter matrix[%d][%d] element : ",i, j);
scanf("%d", &matrix[i][j]);
}
}
printf("\n Original matrix:\n\t");
for (i=0;i<2;i++)
{
for (j=0;j<2;j++)
{
printf("%d\t",matrix[i][j]);
}
printf("\n\t");
}
printf("\n Transposed matrix:\n\t");
for (i=0;i<2;i++)
{
for (j=0;j<2;j++)
{
printf("%d\t",matrix[j][i]);
}
printf("\n\t");
}
}
```

Output