| **Name** | Mahadev Balla |
| --- | --- |
| **UID no.** | 2023300010 |
| **Experiment No.** | 9 |

| **AIM:** | To solve given problems related to pointers. |
| --- | --- |
| **Program 1** | |
| **PROBLEM STATEMENT :** | Write a program to reverse the position of elements in the array using pointers. |
| **ALGORITHM:** | 1) Start  2) Create a function scan(x, size) to read 'size' integers from the user and store them in array x.  3) Create a function print(x, size) to print the elements of array x.  4) Create a function reverse(x, y, size) to reverse the elements of array x and store them in array y.  a. Declare a pointer p pointing to the first element of array y.  b. Iterate i from 0 to size - 1:  i. Set \*(p + i) to the i-th element from the end of array x (x[size - 1 - i]).  (Note: \*(p + i) is equivalent to y[i])    5) In the main function:  a. Read the size of the array from the user.  b. Declare arrays a and b of size 'size'.  c. Call the scan function to read elements into array a.  d. Print "Original array:" followed by a call to the print function with array a and its size.  e. Call the reverse function with arrays a and b, and their size.  f. Print "Array after reversal of elements:" followed by a call to the print function with array b and its size.  g. Return 0 to indicate successful execution.  6) End |
| **PROGRAM:** | #include <stdio.h>  void scan(int x[],int size)  {  printf("Enter elements : \n");  for (int i=0;i<size;i++)  {  scanf(" %d",&x[i]);  }  }  void print(int x[],int size)  {  for (int i=0;i<size;i++)  {  printf(" %d ",x[i]);  }  }  void reverse(int x[], int y[], int size)  {  int \*p=y;  for(int i=0;i<size;i++)  {  \*p=x[size-1-i];  \*p++;  }  }  int main ()  {  int size;  printf("Enter the size of array : ");  scanf("%d",&size);  int a[size],b[size];  scan(a,size);    printf("Original array :");  print(a,size);    reverse(a,b,size);    printf("\nArray after reversal of elements :");  print(b,size);  printf("\n");    return 0;  } |
| **RESULT:** | |
| **Program 2** | |
| **PROBLEM STATEMENT :** | Write a program to calculate the addition of matrices using pointers. Dimensions of the matrix will be decided by the user. |
| **ALGORITHM:** | 1) Start  2) Create a function twoDArrayInput(arr, row, col) to read 'row' x 'col' integers from the user and store them in a 2D array arr.  a. Print "Enter array elements:".  b. Iterate i from 0 to row - 1:  i. Iterate j from 0 to col - 1:  - Read an integer from the user and store it in arr[i][j].  3) Create a function twoDArrayPrint(arr, row, col) to print the elements of a 2D array arr.  a. Iterate i from 0 to row - 1:  i. Iterate j from 0 to col - 1:  - Print arr[i][j] followed by a tab.  ii. Print a newline.  4) Create a function matrixAdd(arr1, arr2, row, col) to add two matrices arr1 and arr2 and print the result.  a. Declare a 2D array solMat to store the result of matrix addition.  b. Iterate i from 0 to row - 1:  i. Iterate j from 0 to col - 1:  - Set solMat[i][j] to arr1[i][j] + arr2[i][j].  5) In the main function:  a. Read the number of rows and columns for the matrices from the user.    b. Declare arrays arr1 and arr2 of size row x col.    c. Call twoDArrayInput for arr1 and print the resulting matrix.  d. Call twoDArrayInput for arr2 and print the resulting matrix.  e. Call matrixAdd with arr1, arr2, row, and col to perform matrix addition and print the result.  f. Return 0 to indicate successful execution.  6) End |
| **PROGRAM:** | #include<stdio.h>  void twoDArrayInput(int arr[10][10], int, int);  void twoDArrayPrint(int arr[10][10], int, int);  void matrixAdd(int arr1[10][10], int arr2[10][10], int , int);  int main() {  int arr1[10][10], arr2[10][10], row, col;  printf(" Matrix 1: \n");  printf("Enter No. of rows(<=10)\n");  scanf("%d",&row);  printf("Enter No. of coloumns(<=10)\n");  scanf("%d", &col);  twoDArrayInput(arr1, row, col);  printf("Matrix 1: \n");  twoDArrayPrint(arr1, row, col);  twoDArrayInput(arr2, row, col);  printf("Matrix 2: \n");  twoDArrayPrint(arr2, row, col);  matrixAdd(arr1,arr2, row, col);    return 0;  }  void twoDArrayInput(int arr[10][10], int row, int col) {  printf("Enter array elements: \n");  for(int i=0; i<row; i++) {  for(int j=0; j<col; j++) {  scanf("%d", &arr[i][j]);  }  }  }  void twoDArrayPrint(int arr[10][10], int row, int col) {  for(int i=0; i<row; i++) {  for(int j=0; j<col; j++) {  printf("%d \t", arr[i][j]);  }  printf("\n");  }  }  void matrixAdd(int arr1[10][10],int arr2[10][10], int row, int col) {  int solMat[10][10];  for(int i=0; i<row; i++) {  for(int j=0; j<col; j++) {  \*(\*(solMat+i)+j)=(\*(\*(arr1+i)+j))+(\*(\*(arr2+i)+j));  }  }  printf("Matrix after addition: \n");  twoDArrayPrint(solMat, row, col);  } |
| **RESULT:** | |
| **CONCLUSION:** | Studied the application of functions and pointers to solve the given functions. |