| **Name** | Mahadev Balla |
| --- | --- |
| **UID no.** | 2023300010 |
| **Experiment No.** | 4 |

| **AIM:** | To solve given problems related to 1D arrays. |
| --- | --- |
| **Program 1** | |
| **PROBLEM STATEMENT :** | Linear Search |
| **ALGORITHM:** | 1) START  2) DEFINITION OF FUNCTION “PRINT” HAVING PARAMETERS  “(INT M[ ],INT SIZE)” :  INITIALIZE I=0  FOR(I<SIZE)  IF(M[ I ] != -1) PRINT “M[ I ]”, SET “I=I+1”  END IF  END FOR LOOP  3) DEFINITION OF FUNCTION “READ” HAVING PARAMETERS  “(INT M[ ],INT SIZE)” :  INITIALIZE I=0  FOR(I<SIZE)  INPUT “MARKS[ I ]”, SET “I=I+1”  END FOR LOOP  4) DEFINITION OF FUNCTION “SEARCH” OF INT RETURN-TYPE  HAVING PARAMETERS “(INT M[ ], INT A[], INT SIZE, INT TAR)” :  INITIALIZE ‘I=0, COUNTER=0’  FOR(I<SIZE)  IF(M[ I ]=TAR) SET ‘COUNTER=COUNTER+1, A[ I ]=I’  END IF  ELSE SET ‘A[ I ] = -1’  END ELSE  END FOR LOOP  RETURN COUNTER  5) DECLARE ‘SIZE, TARGET’  6) PRINT “Enter the size of array : ”  7) INPUT “SIZE”  8) DECLARE ‘ARR[SIZE], INDEX[SIZE]’  9) CALL THE FUNCTION ‘READ’ PASSING ARGUMENTS ‘ARR’  AND ‘SIZE’  10) PRINT “Enter the element to be searched : ”  11) INPUT “TARGET”  12) INITIALIZE REC = SEARCH(ARR,INDEX,SIZE,TARGET)  13) IF(REC != 0)  PRINT “ ‘TARGET’ is found ‘REC’ times at index : ”  CALL THE FUNCTION ‘PRINT’ PASSING ARGUMENTS ‘INDEX’  AND ‘SIZE’  END IF  ELSE  PRINT “ ‘TARGET’ is not present in array. ”  14) END |
| **PROGRAM:** | #include <stdio.h>  void print(int m[],int size)  {  for(int i=0;i<size;i++)  {  if(m[i]!=-1)  {  printf("%d,",m[i]);  }  }  }  void read(int m[],int size)  {  printf("Enter elements :\n");  for(int i=0;i<size;i++)  {  scanf("%d",&m[i]);  }  }  int search(int m[],int a[],int size,int tar)  {  int counter=0;  for(int i=0;i<size;i++)  {  if(m[i]==tar)  {  counter++;  a[i]=i;  }  else  {  a[i]=-1;  }    }  return counter;  }      int main()  {  int size,target;  printf("Enter the size of array : ");  scanf("%d",&size);    int arr[size],index[size];  read(arr,size);    printf("\nEnter the element to be searched : ");  scanf("%d",&target);    int rec=search(arr,index,size,target);    if(rec!=0)  {  printf("%d is found %d times at index : ",target,rec);  print(index,size);  }  else  {  printf("%d is not present in array.",target);  }  printf("\n");  return 0;  } |
| **RESULT:** | |
| **Program 2** | |
| **PROBLEM STATEMENT :** | Sorting of an Array |
| **ALGORITHM:** | 1) START  2) DEFINITION OF FUNCTION “PRINT” HAVING PARAMETER  “(INT MARKS[ ],INT N)” :  INITIALIZE I=0  FOR(I<N)  PRINT “MARKS[ I ]”, SET “I=I+1”  END FOR LOOP  3) DEFINITION OF FUNCTION “SCAN” HAVING PARAMETERS  “(INT MARKS[ ],INT N)” :  INITIALIZE I=0  FOR(I<N)  INPUT “MARKS[ I ]”, SET “I=I+1”  END FOR LOOP  4)DEFINITION OF FUNCTION “BUBBLE SORT” HAVING  PARAMETERS “(INT A[ ],INT N)” :  INITIALIZE J=0, TEMP=0, PASS=0;  FOR(PASS<N-1)  FOR(J<N-1-PASS)  IF(A[ J ] > A[ J+1 ]) SET “TEMP=A[J], A[J]=A[J+1], A[J+1]=TEMP”  END IF  END FOR LOOP  END FOR LOOP  5) DECLARE N  6) PRINT “Enter the size of array : ”  7) INPUT “N”  8) DECLARE AN ARRAY “ARR[N]”  9) CALL THE FUNCTION ‘SCAN’ PASSING ARGUMENTS ‘ARR’  AND ‘N’  10) PRINT “Original array :”  11) CALL THE FUNCTION ‘PRINT’ PASSING ARGUMENTS ‘ARR’  AND ‘N’  12) CALL THE FUNCTION ‘BUBBLE SORT’ PASSING ARGUMENTS  ‘ARR’ AND ‘N’  13) PRINT “Array after sorting :”  14) CALL THE FUNCTION ‘PRINT’ PASSING ARGUMENTS ‘ARR’  AND ‘N’  15) END |
| **PROGRAM:** | #include <stdio.h>  void print(int marks[],int n)  {  for(int i=0;i<n;i++)  {  printf(" %d ",marks[i]);  }  }  void scan(int marks[],int n)  {  printf("Enter elements: \n");  for(int i=0;i<n;i++)  scanf("%d",&marks[i]);  }  void bubblesort(int a[],int n)  {  int temp=0;  for(int pass=0;pass<n-1;pass++)  {  for(int j=0;j<n-1-pass;j++)  {  if(a[j]>a[j+1])  {  temp = a[j];  a[j] = a[j+1];  a[j+1] = temp;  }  }  }  }  int main ()  {  int n;  printf("Enter the size of array : ");  scanf("%d",&n);    int arr[n];  scan(arr,n);  printf("Original array :");  print(arr,n);  bubblesort(arr,n);  printf("\nArray after sorting :");  print(arr,n);  printf("\n");    return 0;    } |
| **RESULT:** | |
| **CONCLUSION:** | Studied the application of functions to solve problems related to 1D arrays. |