|  |  |
| --- | --- |
| **Name** | Shubhan Singh |
| **UID no.** | 2022300118 |
| **Experiment No.** | 5 |

|  |  |
| --- | --- |
| **AIM:** | **To demonstrate the use of one-dimensional arrays to solve a given problem.** |
| **Program 1** | |
| **PROBLEM STATEMENT :** | **The selection sort algorithm sorts an array by repeatedly finding the minimum element (considering ascending order) from the unsorted part and putting it at the beginning. The algorithm maintains two sub-arrays in a given array. Implement it in C language.** |
| **ALGORITHM:** | Step 1: START  Step 2: Read number of elements to sort and store it in n.  Step 3: declare an array inputarray of width n.  Step 4: Read n input elements from user input and store them in inputarray.  Step 5: sort inputarray by executing sort(inputarray,n)  Step 6: print sorted array by executing print(inputarray,n)  Step 7: END  Algorithm for function sort(int \* arr, int n)(here, n is the size of passed array)  Step 1: set i=0  Step 2: set j=i+1  Step 3: if arr[i]>arr[j], swap the values of arr[i] and arr[j]  Step 4: if j<n, increment j and return to step 3.  Step 5: if i<n-1, increment I and return to step 2.  Algorithm for function printarray(int \* arr, int n)(n is the size of the passed array)  Step 1: set i=0  Step 2: print the value of arr[i]  Step 3: if i<n, increment i and return to step 2. |
| **PROGRAM:** | #include<stdio.h>  void sort (int *arr*[], int *n*){      int temp;      for(int i=0;i<*n*-1;i++){          for(int j=i+1;j<*n*;j++){              if(*arr*[i]>*arr*[j]){                  temp=*arr*[i];  *arr*[i]=*arr*[j];  *arr*[j]=temp;                  }          }      }  }  void printarray(int *arr*[], int *n*){      printf("The sorted array is: ");      for(int i=0;i<*n*;i++){          printf("%d ",*arr*[i]);      }  }  int main(){      int n;      printf("Enter number of elements to sort\n");      scanf("%d",&n);      int inputarray[n];      printf("Enter all array elements\n");      for(int i=0;i<n;i++){          scanf("%d",&inputarray[i]);      }      sort(inputarray,n);      printarray(inputarray,n);      return 0;  } |
| **RESULT:** | |
| **Program 2** | |
| **PROBLEM STATEMENT :** | **Perform search of a particular element in an array using binary search.** |
| **ALGORITHM:** | Step 1: START  Step 2: Read number of elements in array and store it in n.  Step 3: declare an array inputarray of width n.  Step 4: read n elements from user input and store them in inputarray.  Step 5: sort inputarray by executing sort(inputarray,n).  Step 6: read value to be searched and store it in x.  Step 7: execute binarysearch(inputarray,n,x) and store returned value, that is the position of the value in b.  Step 8: if b equals -1, print required element is not present in array.  Step 9: else print that required element is at bth position in sorted array.  Step 10: END  Algorithm for function sort(int \* arr, int n)(here, n is the size of passed array)  Step 1: set i=0  Step 2: set j=i+1  Step 3: if arr[i]>arr[j], swap the values of arr[i] and arr[j]  Step 4: if j<n, increment j and return to step 3.  Step 5: if i<n-1, increment I and return to step 2.  Algorithm for integer function binarysearch(int \*arr, int n, int x)(here n is the size of passed array and x is element to be searched)  Step 1: declare ub=n-1,lb=0 and m.  Step 2: set m to (lb+ub)/2  Step 3: if arr[m] is equal to x, return m and terminate execution of function.  Step 4 else if arr[m]>x, set ub to m-1.  Step 5: else if arr[m]<x, set lb to m+1  Step 6: if ub>=lb, return to step 2  Step 7: return -1 |
| **PROGRAM:** | #include<stdio.h>  void sort (int *arr*[], int *n*){      int temp;      for(int i=0;i<*n*-1;i++){          for(int j=i+1;j<*n*;j++){              if(*arr*[i]>*arr*[j]){                  temp=*arr*[i];  *arr*[i]=*arr*[j];  *arr*[j]=temp;              }          }      }  }  int binarysearch(int *arr*[], int *n*, int *x*){      int ub=*n*-1,lb=0,m;      m=(*n*-1)/2;      while(ub>=lb){          if(*arr*[m]==*x*){return m;}          else if(*arr*[m]>*x*){ub=m-1;}          else if(*arr*[m]<*x*){lb=m+1;}          m=(ub+lb)/2;      }      return -1;  }  int main(){      int n,x,b;      printf("Enter number of elements in array\n");      scanf("%d",&n);      int inputarray[n];      printf("Enter all array elements\n");      for(int i=0;i<n;i++){          scanf("%d",&inputarray[i]);      }      printf("Enter value to be searched\n");      scanf("%d",&x);      sort(inputarray,n);      b=binarysearch(inputarray,n,x);      if(b==-1){printf("This number is not present in the array\n");}      else{printf("%d is at position %d in the sorted array\n",x,b+1);}      return 0;  } |
| **RESULT:** | |