PROGRAM NO: 11 DATE: 02-11-2023

**AIM**

Write a program to implement any three sorting algorithm.

**ALGORITHM**

1. START
2. FUNC bubbleSort(arr, n):
3. DECLARE I, j, swapped, temp
4. FOR i=0 to i<n
5. SET swapped = 0
6. FOR j=0 to j<n-i-1
7. IF arr[j] > arr[j + 1]:
8. SWAP arr[j] and arr[j+1]
9. SET swapped = 1. END IF
10. END FOR
11. IF swapped == 0: RETURN
12. END FOR. END FUNC
13. FUNC selectionSort(arr, n):
14. DECLARE I, j, minIndex, minValue, temp
15. FOR i=0 to i<length:
16. SET minValue = arr[i], minIndex = i
17. FOR j=i+1 to j<length:
18. IF arr[j] < minValue:
19. SET minValue = arr[j], minIndex = j
20. END IF. END FOR
21. SWAP arr[i] and arr[minIndex]
22. END FOR. END FUNC
23. FUNC insertionSort(arr, n):
24. DECLARE i, value, index
25. FOR i=0 to i<length
26. SET value=arr[i], index = i
27. WHILE index > 0 and arr[index – 1] > value:
28. SET arr[index] = arr[index – 1]
29. DECREMENT index
30. END WHILE
31. SET arr[index] = value
32. END FOR. END FUNC
33. STOP

**CODE**

#include <iostream>

using namespace **std**;

void **bubbleSort**(int \*arr, int n) {

    int i, j, swapped, temp;

    for (i=0; i<n; i++) {

        swapped = 0;

        for (j=0; j<n-i-1; j++) {

            if (arr[j] > arr[j+1]) {

                temp = arr[j];

                arr[j] = arr[j+1];

                arr[j+1] = temp;

                swapped = 1;

            }

        }

        if (swapped == 0) {

            return;

        }

    }

}

void **selectionSort**(int \*arr, int length) {

    int i, j, minIndex, minValue, temp;

    for (i=0; i<length; i++) {

        minValue = arr[i];

        minIndex = i;

        for (j=i+1; j<length; j++) {

            if (arr[j] < minValue) {

                minValue = arr[j];

                minIndex = j;

            }

        }

*// swapping*

        temp = arr[i];

        arr[i] = arr[minIndex];

        arr[minIndex] = temp;

    }

}

void **insertionSort**(int \*arr, int length) {

    int i, value, index;

    for (i=1; i<length; i++) {

        value = arr[i];

        index = i;

        while (index > 0 && arr[index - 1] > value) {

            arr[index] = arr[index - 1];

            index--;

        }

        arr[index] = value;

    }

}

int **main**() {

    int length, choice;

    cout **<<** "Enter length of array: ";

    cin **>>** length;

    int \*arr = (int\*) **malloc**(length \* sizeof(int));

    for (int i=0; i<length; i++) {

        cout **<<** "Enter item " **<<** i+1 **<<** " : ";

        cin **>>** arr[i];

    }

    cout **<<** "Before Sorting: " **<<** **endl**;

    for (int i=0; i<length; i++) cout **<<** arr[i] **<<** ", ";

    cout **<<** **endl**;

    cout **<<** "--------- Sorting Methods ---------" **<<** **endl**;

    cout **<<** "1. Bubble Sort" **<<** **endl**;

    cout **<<** "2. Selection Sort" **<<** **endl**;

    cout **<<** "3. Insertion Sort" **<<** **endl**;

    cout **<<** "4. Exit" **<<** **endl**;

    cout **<<** "Enter choice: ";

    cin **>>** choice;

    if (choice == 1) {

**bubbleSort**(arr, length);

    } else if (choice == 2) {

**selectionSort**(arr, length);

    } else if (choice == 3) {

**insertionSort**(arr, length);

    }

    cout **<<** "After Sorting: " **<<** **endl**;

    for (int i=0; i<length; i++) cout **<<** arr[i] **<<** ", ";

    cout **<<** **endl**;

    return 0;

}

**OUTPUT**

Enter length of array: 6

Enter item 1 : 12

Enter item 2 : -1

Enter item 3 : 2

Enter item 4 : -4

Enter item 5 : 6

Enter item 6 : 3

Before Sorting:

12, -1, 2, -4, 6, 3,

--------- Sorting Methods ---------

1. Bubble Sort

2. Selection Sort

3. Insertion Sort

4. Exit

Enter choice: 1

After Sorting:

-4, -1, 2, 3, 6, 12,