**Program – 4**

**AIM – Write an algorithm and program to implement Selection Sort**

**Algorithm –**

beginBubbleSort(list)

for all elements of list

if list[i] > list[i+1]

swap(list[i], list[i+1])

end if

end for

return list

endBubbleSort

**Using Arrays**

**Source Code –**

#include <bits/stdc++.h>

using namespace std;

void swap(int \*xp, int \*yp)

{

int temp = \*xp;

\*xp = \*yp;

\*yp = temp;

}

voidbubbleSort(intarr[], int n)

{

inti, j;

for (i = 0; i< n-1; i++)

for (j = 0; j < n-i-1; j++)

if (arr[j] >arr[j+1])

swap(&arr[j], &arr[j+1]);

}

voidprintArray(intarr[], int size)

{

inti;

for (i = 0; i< size; i++)

cout<<arr[i] << " ";

cout<<endl;

}

int main()

{

intarr[] = {64, 34, 25, 12, 22, 11, 90};

int n = sizeof(arr)/sizeof(arr[0]);

bubbleSort(arr, n);

cout<<"Sorted array: \n";

printArray(arr, n);

return 0;

}

**Using recursion**

**Source Code –**

#include <bits/stdc++.h>

using namespace std;

voidbubbleSort(intarr[], int n)

{

// Base case

if (n == 1)

return;

for (inti=0; i<n-1; i++)

if (arr[i] >arr[i+1])

swap(arr[i], arr[i+1]);

bubbleSort(arr, n-1);

}

voidprintArray(intarr[], int n)

{

for (inti=0; i< n; i++)

printf("%d ", arr[i]);

printf("\n");

}

int main()

{

intarr[] = {64, 34, 25, 12, 22, 11, 90};

int n = sizeof(arr)/sizeof(arr[0]);

bubbleSort(arr, n);

printf("Sorted array : \n");

printArray(arr, n);

return 0;

}

**Using Recursion and Linked List**

**Source Code –**

#include <iostream>

using namespace std;

struct Node

{

int data;

struct Node\* next;

} Node;

struct Node\* swap(struct Node\* ptr1, struct Node\* ptr2)

{

struct Node\* tmp = ptr2->next;

ptr2->next = ptr1;

ptr1->next = tmp;

return ptr2;

}

intbubbleSort(struct Node\*\* head, int count)

{

struct Node\*\* h;

inti, j, swapped;

for (i = 0; i<= count; i++)

{

h = head;

swapped = 0;

for (j = 0; j < count - i - 1; j++)

{

struct Node\* p1 = \*h;

struct Node\* p2 = p1->next;

if (p1->data > p2->data)

{

\*h = swap(p1, p2);

swapped = 1;

}

h = &(\*h)->next;

}

if (swapped == 0)

break;

}

}

voidprintList(struct Node\* n)

{

while (n != NULL)

{

cout<< n->data << " -> ";

n = n->next;

}

cout<<endl;

}

voidinsertAtTheBegin(struct Node\*\* start\_ref, int data)

{

struct Node\* ptr1

= (struct Node\*)malloc(sizeof(struct Node));

ptr1->data = data;

ptr1->next = \*start\_ref;

\*start\_ref = ptr1;

}

int main()

{

intarr[] = { 78, 20, 10, 32, 1, 5 };

intlist\_size, i;

struct Node\* start = NULL;

list\_size = sizeof(arr) / sizeof(arr[0]);

for (i = 0; i<list\_size; i++)

insertAtTheBegin(&start, arr[i]);

cout<<"Linked list before sorting\n";

printList(start);

bubbleSort(&start, list\_size);

cout<<"Linked list after sorting\n";

printList(start);

return 0;

}