Day-36 of the #101 days of coding challenge

⇨ Sorting Algorithms---

1-> Bubble Sort---

Bubble sort is a sorting algorithm that compares two adjacent elements and swaps them until they are in the intended order.

➔ If static array data is Given the find the length

```
  // find array's length
int size = sizeof(data) / sizeof(data[0]);
```

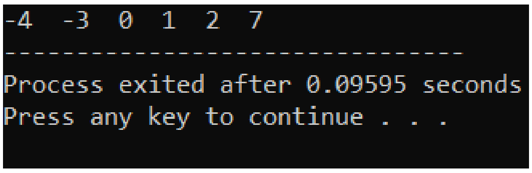
Code:-

```
#include<iostream>

using namespace std;

int bubbleSort(int *arr, int n)
{
   int i, j;

   for(i = 1; i<=n; i++)
   {
         for(j = 0; j<n-i; j++) // here leaves last elements in each iteration
         {
               if(arr[j] > arr[j+1])
```

```
          {
              int temp = arr[j];
              arr[j]   = arr[j+1];
              arr[j+1] = temp;
          }
      }
  }

  // showing the data
  for(i = 0; i<n; i++)
  {
      cout<<arr[i]<<"  ";
  }
}

int main()
{
  int arr[] = {-3, -4, 0, 7, 2, 1};

  // finding the length
  int size = sizeof(arr) / sizeof(arr[0]);

  bubbleSort(arr, size);
  return 0;
```

```
}

//-3, -4, 0, 7, 2, 1
//-4, -3, 0, 7, 2, 1
//-4, -3 ,0, 7, 2, 1;
//-4, -3, 0, 2, 7, 1
//-4, -3, 0, 2, 1, 7 (here in each iteration
maximum elements and then just less then
coming to the end so need to skip every time
last element)
```

Output:-



```
-4  -3  0  1  2  7
--------------------------------
Process exited after 0.09595 seconds
Press any key to continue . . .
```

2- Selection Sort---

Selection sort is a sorting algorithm that selects the smallest element from an unsorted list in each iteration and places that element at the beginning of the unsorted list.

Algorithms:-

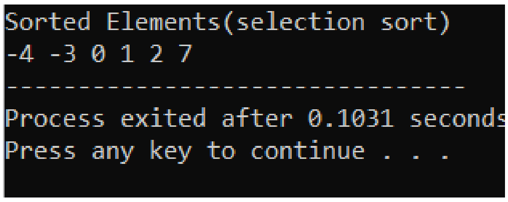
Compare `minimum` with the second element. If the second element is smaller than `minimum`, assign the second element as `minimum`.

Compare `minimum` with the third element. Again, if the third element is smaller, then assign `minimum` to the third element otherwise do nothing. The process goes on until the last element.

Code:-

```
#include<iostream>

using namespace std;

void swap(int *a, int *b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}

void selectionSort(int arr[], int n)
{
    int i, j;
    
    for(i = 0; i<n; i++)
    {
            int min_valueAtIndex = i;  // first elements suppose to the minimum element
            for(j = i+1; j<n; j++)
            {
                    if(arr[j] < arr[min_valueAtIndex])
                    {
                            min_valueAtIndex = j;
                    }
            }
            
            // now after getting the minimum elements into the each iteration need to swap from max to min index
            swap(&arr[min_valueAtIndex], &arr[i]);
    }
    
    // displaying the elements;
    
    for(i = 0; i<n; i++)
    {
            cout<<arr[i]<<" ";
    }
}
int main()
{
    int arr[] = {-3, -4, 0, 7, 2, 1};

    // finding the length
    int size = sizeof(arr) / sizeof(arr[0]);

    cout<<"Sorted Elements(selection sort)"<<endl;
    selectionSort(arr, size);
    return 0;
}
```

Output:-



```
Sorted Elements(selection sort)
-4 -3 0 1 2 7
--------------------------------
Process exited after 0.1031 seconds
Press any key to continue . . .
```