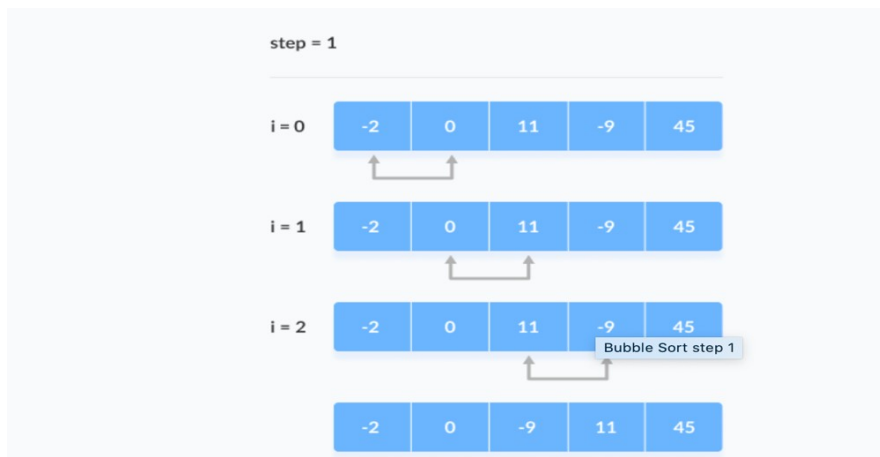
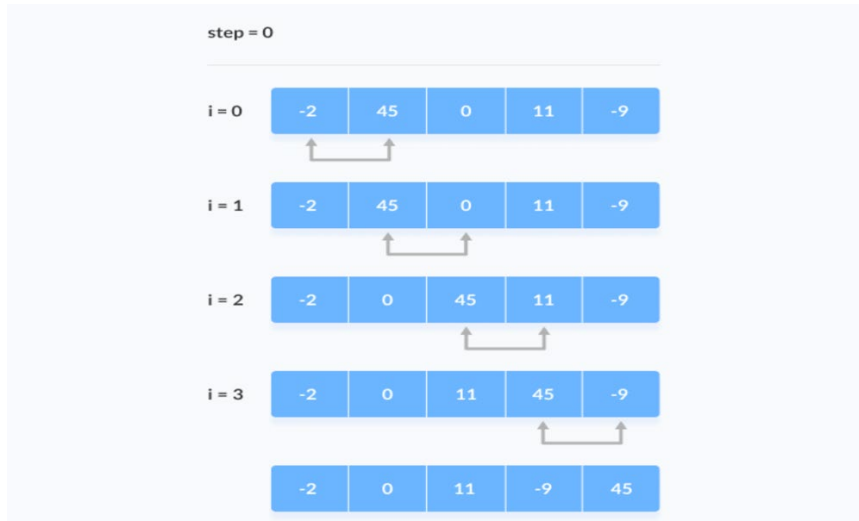


## Question Lab Exercise 2 Part 1

1. Study step 0, step 1, step 2, and step 3.





- Write an **algorithm** to perform bubble sort on a given array of integers.
- Write a program to perform bubble sort on an array of N elements. It uses the **algorithm** depicted in (a).

### Question 1a

Step 1: for (pass=1;(pass < sizearray) && !sort; pass++)

Step 2: for (int x=0; x < sizearray-pass; x++)

Step 3: if A[x] > A[x+1]

Temp = A[x]

A[x] = A[x+1]

A[x+1] = temp

[End of inner loop]

[End of outer loop]

Step 4: Exit program

### Question 1b

```
//QUESTION 1b
//NAME: BRENDAN DYLAN GAMPA ANAK JOSEPH DUSIT@DUSIT
//MATRIC NO: A20EC0021

#include <iostream>
using namespace std;

void bubbleswap(int arraycont[],int sizearray){
    int pass, temp, swap=0;
    bool sort=false;
    for(pass=1;(pass<sizearray) && !sort;pass++){
        sort=true;

        for(int x=0;x<sizearray-pass;x++){
            if(arraycont[x]>arraycont[x+1]){
                temp=arraycont[x];
                arraycont[x]=arraycont[x+1];
                arraycont[x+1]=temp;
                swap++;
            }
        }
        sort=false;
    }
}

//display content
for(int a=0;a<sizearray;a++){
    cout<<arraycont[a]<<" ";
}
cout<<endl;

cout<<"Number of pass: "<<pass<<endl;
cout<<"Number of swap: "<<swap<<endl;
}
```

```
int main(){
    int content[5]={-2,45,0,11,-9};
    int sizearray=5;

    bubbleswap(content,sizearray);

    return 0;
}
```

2. Below table illustrates the sorting of list L in ascending order using insertion sort technique. As we can see in the above illustration, four passes were required to sort a list of five elements. Hence, we can say that insertion sort require n-1 passes to sort an array of n elements.

Pass	Comparison	Resultant Array										
1	<table><tr><td>18</td><td>3</td><td>2</td><td>33</td><td>21</td></tr></table>	18	3	2	33	21	<table><tr><td>3</td><td>18</td><td>2</td><td>21</td><td>33</td></tr></table>	3	18	2	21	33
18	3	2	33	21								
3	18	2	21	33								
2	<table><tr><td>18</td><td>3</td><td>2</td><td>33</td><td>21</td></tr></table>	18	3	2	33	21	<table><tr><td>2</td><td>3</td><td>18</td><td>33</td><td>21</td></tr></table>	2	3	18	33	21
18	3	2	33	21								
2	3	18	33	21								
3	<table><tr><td>2</td><td>3</td><td>18</td><td>33</td><td>21</td></tr></table>	2	3	18	33	21	<table><tr><td>2</td><td>3</td><td>18</td><td>33</td><td>21</td></tr></table>	2	3	18	33	21
2	3	18	33	21								
2	3	18	33	21								
4	<table><tr><td>2</td><td>3</td><td>18</td><td>33</td><td>21</td></tr></table>	2	3	18	33	21	<table><tr><td>2</td><td>3</td><td>18</td><td>21</td><td>33</td></tr></table>	2	3	18	21	33
2	3	18	33	21								
2	3	18	21	33								
<div><div></div> → denotes the previously sorted sub array <div></div> → denotes the current selection</div>												

- Write an **algorithm** to perform insertion sort on a given array of integers.
- Write a program to perform insertion sort on an array of N elements. It uses the algorithm depicted in (a).

### Question 2a.

Step 1: for (pass=1; pass < listSize; pass++)

Step 2: while ((insertIndex > 0) && (element[insertIndex-1] > item))

Step 3: element [insertIndex] = element [insertIndex-1]

[end of while loop]

Step 4: element [insertIndex] = item

[end of for loop]

Step 5: program exit

## Question 2b

```
//QUESTION 2b
//NAME: BRENDAN DYLAN GAMPA ANAK JOSEPH DUSIT@DUSIT
//MATRIC NO: A20EC0021

#include <iostream>
using namespace std;

void Insertion (int element[],int listSize){
    int item,pass,insertIndex;

    for(pass=1; pass<listSize; pass++){
        item = element[pass];
        insertIndex = pass;

        while ((insertIndex > 0) && (element[insertIndex-1] > item)){
            element[insertIndex] = element[insertIndex-1];
            insertIndex--;
        }

        element[insertIndex] = item;
    }

    //Display content
    for(int i=0;i<listSize;i++){
        cout<<element[i]<<" ";
    }
}

main(){
    int element[5]={18,3,2,33,21},listSize=5;

    Insertion(element,listSize);
}
```