Practical 1 (A)

Aim: Insert an element at a specific position in an array.

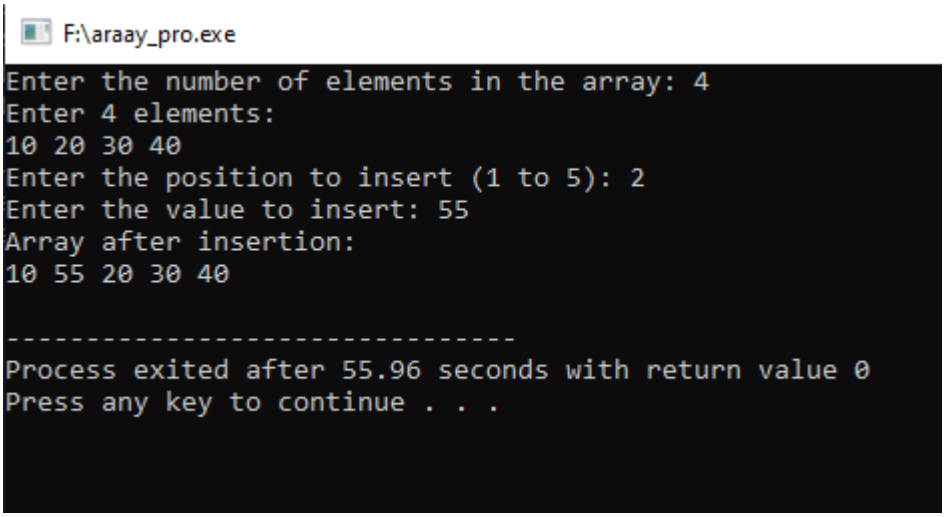
An array is a collection of similar items stored in a single line in memory.

Aim: Insert an element at a specific position in an array.

Code:

```
#include <iostream>
using namespace std;

int main() {
   int arr[100], n, pos, value;

   // Input the number of elements
   cout << "Enter the number of elements in the array: ";
   cin >> n;

   // Input the elements
   cout << "Enter " << n << " elements:\n";
   for(int i = 0; i < n; i++) {
      cin >> arr[i];
   }

   // Input the position and value to insert
   cout << "Enter the position to insert (1 to " << n+1 << "): ";
   cin >> pos;
   cout << "Enter the value to insert: ";
   cin >> value;

   // Check for valid position
   if(pos < 1 || pos > n+1) {
      cout << "Invalid position!" << endl;
      return 1;
   }
   // Shift elements to the right to make space
   for(int i = n; i >= pos; i--) {
      arr[i] = arr[i - 1];
   }
   // Insert the new element
   arr[pos - 1] = value;
   n++; // Increase the size of the array
   // Output the updated array
   cout << "Array after insertion:\n";
   for(int i = 0; i < n; i++) {
      cout << arr[i] << " ";
   }
   cout << endl;

   return 0;
}
```

Output:


F:\araay_pro.exe
Enter the number of elements in the array: 4
Enter 4 elements:
10 20 30 40
Enter the position to insert (1 to 5): 2
Enter the value to insert: 55
Array after insertion:
10 55 20 30 40

Process exited after 55.96 seconds with return value 0
Press any key to continue . . .

Practical 1 (B)

Aim: Delete an element from a specific position in an array.

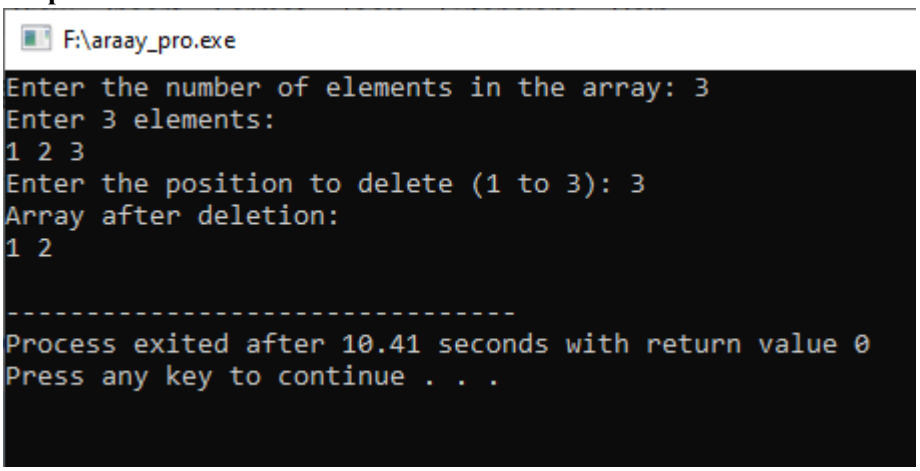
Code:

```
#include <iostream>
using namespace std;

int main() {
   int arr[100], n, pos;

   // Input the number of elements
   cout << "Enter the number of elements in the array: ";
   cin >> n;

   // Input the elements
   cout << "Enter " << n << " elements:\n";
   for(int i = 0; i < n; i++) {
      cin >> arr[i];
   }

   // Input the position to delete
   cout << "Enter the position to delete (1 to " << n << "): ";
   cin >> pos;

   // Check for valid position
   if(pos < 1 || pos > n) {
      cout << "Invalid position!" << endl;
      return 1;
   }

   // Shift elements to the left to delete the element
   for(int i = pos - 1; i < n - 1; i++) {
      arr[i] = arr[i + 1];
   }

   n--; // Decrease the size of the array

   // Output the updated array
   cout << "Array after deletion:\n";
   for(int i = 0; i < n; i++) {
      cout << arr[i] << " ";
   }
   cout << endl;

   return 0;
}
```

By. Prof. Krutika H. Churi

SDSM College,Palghar

Output:



By. Prof. Krutika H. Churi

SDSM College,Palghar

Practical 1 (C)

Aim: Search for an element in an array (linear search).

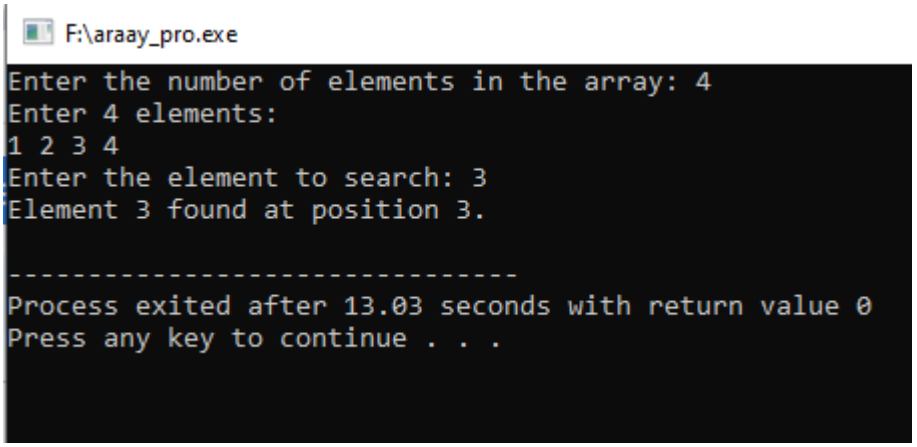
Code: #include <iostream>
using namespace std;

```
int main() {
   int arr[100], n, key, found = 0;

   // Input the number of elements
   cout << "Enter the number of elements in the array: ";
   cin >> n;

   // Input the elements
   cout << "Enter " << n << " elements:\n";
   for(int i = 0; i < n; i++) {
      cin >> arr[i];
   }

   // Input the element to search for
   cout << "Enter the element to search: ";
   cin >> key;

   // Perform linear search
   for(int i = 0; i < n; i++) {
      if(arr[i] == key) {
         cout << "Element " << key << " found at position " << i + 1 << "." << endl;
         found = 1;
         break;
      }
   }

   if(!found) {
      cout << "Element " << key << " not found in the array." << endl;
   }

   return 0;
}
```

Output:

By. Prof. Krutika H. Churi

SDSM College,Palghar



By. Prof. Krutika H. Churi

SDSM College,Palghar