Challenge 3.1

```cpp
#include <iostream>

#include <vector>

#include <string>


std::vector<int> linear_search_product(const std::vector<std::string>& product_list, const std::string& target_product) {

    std::vector<int> indices;


    for (int i = 0; i < product_list.size(); ++i) {

        if (product_list[i] == target_product) {

            indices.push_back(i);

        }

    }


    return indices;

}


int main() {

    std::vector<std::string> products = {"apple", "banana", "apple", "orange", "apple"};

    std::string target = "apple";

    std::vector<int> result = linear_search_product(products, target);


    if (result.empty()) {

        std::cout << "Product not found." << std::endl;

    } else {
```

```cpp
        std::cout << "Product found at indices: ";

        for (int index : result) {

            std::cout << index << " ";

        }

        std::cout << std::endl;

    }


    return 0;

}
```



```cpp
1    #include <iostream>
2    #include <vector>
3    #include <string>
4
5    std::vector<int> linear_search_product(const
     std::vector<std::string>& product_list, const std::string&
     target_product) {
6        std::vector<int> indices;
7
8        for (int i = 0; i < product_list.size(); ++i) {
9            if (product_list[i] == target_product) {
10               indices.push_back(i);
11           }
12       }
13
14       return indices;
15   }
16
17   int main() {
18       std::vector<std::string> products = {"apple", "banana",
     "apple", "orange", "apple"};
19       std::string target = "apple";
20       std::vector<int> result = linear_search_product(products,
     target);
```

Console:
```
sh -c make -s
./main
Product found at indices: 0 2 4
```

Ln 34, Col 1   History

Challenge3.2

```python
# create an empty dictionary

D = {}

n = int(input('How many student record you want to store?'))

 # create an empty list

 # Add student information to the list

ls = []

for i in range(0, n):

 x=input("Enter the student name.")

 y=input("Enter the Roll No of " + x+":")

 z=input("Enter the CGPA of " + x+":")

         # Add name and marks stored in x, y, z

         # respectively using tuple to the list

 ls.append((z,y,x,))

 # sort the elements of list

 # based on marks

ls = sorted(ls, reverse = True)

print('Sorted list of students according to their marks in   descending order')

space=''

print("CGPA", space*5, "ROLL.NO", space*5, "NAME")

for i in ls:

         # print name and marks stored in

         # second and first position

         # respectively in list of tuples.

   print(i[0],space*5,i[1],space*5,i[2])
```

```python
10    z=input("Enter the CGPA of " + x+":")
11    __# Add name and marks stored in x, y, z
12    # respectively using tuple to the list
13    ls.append((z,y,x,))
14    # sort the elements of list
15    # based on marks
16    ls = sorted(ls, reverse = True)
17    print('Sorted list of students according to their marks in
      descending order')
18    space=''
19    print("CGPA", space*5, "ROLL.NO", space*5, "NAME")
20    for i in ls:
21        # print name and marks stored in
22        # second and first position
23        # respectively in list of tuples.
24        print(i[0],space*5,i[1],space*5,i[2])
```

How many student record you want to store?5
Enter the student name.Arun
Enter the Roll No of Arun:cs01
Enter the CGPA of Arun:5.67
Enter the student name.Balaji
Enter the Roll No of Balaji:cs02
Enter the CGPA of Balaji:7.67
Enter the student name.Chandru
Enter the Roll No of Chandru:cs03
Enter the CGPA of Chandru:6.65
Enter the student name.Dheena
Enter the Roll No of Dheena:cs04
Enter the CGPA of Dheena:4.56
Enter the student name.Ezhil
Enter the Roll No of Ezhil:cs05
Enter the CGPA of Ezhil:8.78
Sorted list of students according to their marks in   descending order
CGPA ROLL.NO NAME
8.78 cs05 Ezhil
7.67 cs02 Balaji
6.65 cs03 Chandru
5.67 cs01 Arun
4.56 cs04 Dheena
›