**Reg. No: 21BCE1297** **Name: Vidhi Shah** **Date: 03/06/22**

# DA1 & DA2

## Q1

## Aim:

Write a C program to store the computer network details using a structure and pass this structure as an argument to a function that displays a list of messages identifying each pair of computers from the same locality. In the messages, the computers should be identified by their nicknames.

## Procedure:

### Input:

Number of computers, n
Nickname
IP Address

### Output:

Pair of computers from the same locality identified by their nickname

### Algorithm:

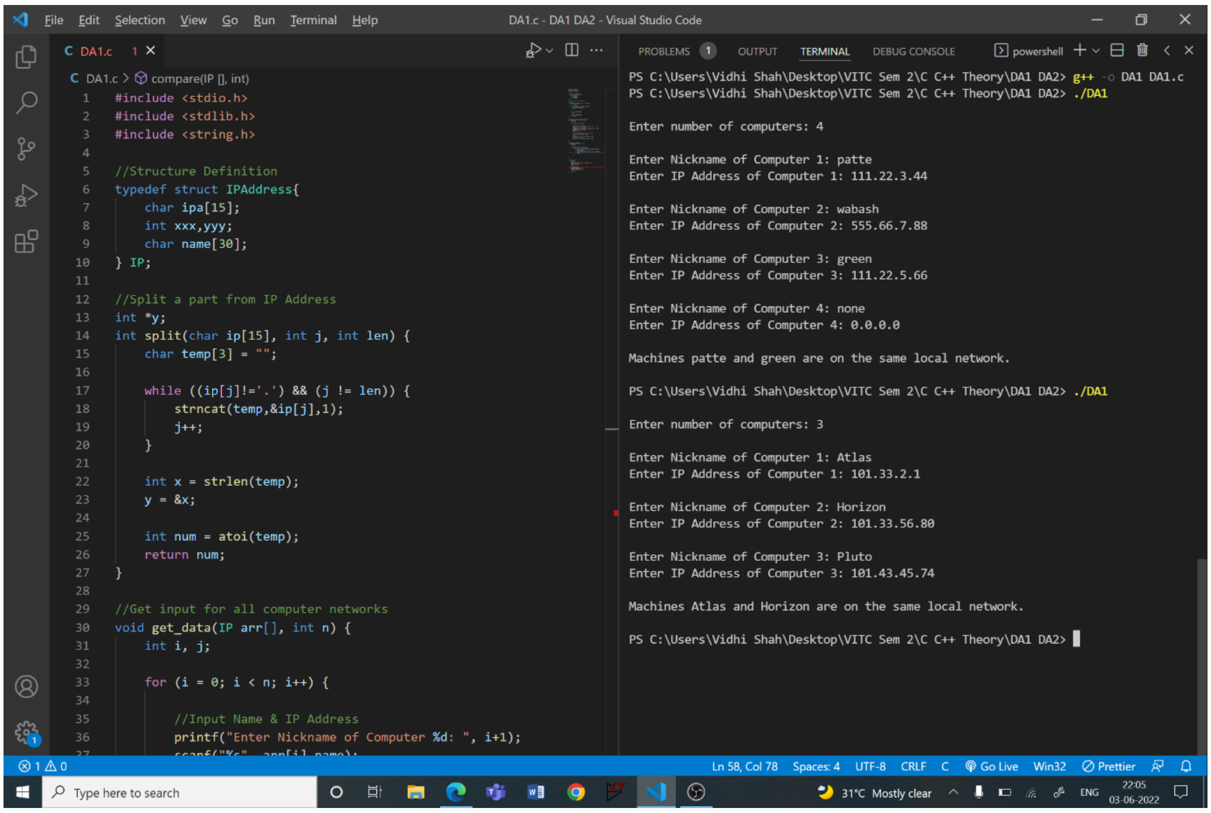
Step 1: Create a structure IPAddress with data types ipa(string), xxx(int), yyy(int) and name(string)
Step 2: Create a split function to split the first two part of the addresses by using '.' As a delimiter and string concatenation
Step 3: Create a get_data function that takes input for array of computer networks and use split function for each network
Step 4: Create a compare function to compare 'xxx' and 'yyy' component of each object in the array and print the nicknames of computers whose first two components match.

## Code:

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

//Structure Definition
typedef struct IPAddress{
    char ipa[15];
    int xxx,yyy;
    char name[30];
} IP;

//Split a part from IP Address
int *y;
int split(char ip[15], int j, int len) {
    char temp[3] = "";

    while ((ip[j]!='.') && (j != len)) {
        strncat(temp,&ip[j],1);
        j++;
    }

    int x = strlen(temp);
    y = &x;

    int num = atoi(temp);
    return num;
}

//Get input for all computer networks
void get_data(IP arr[], int n) {
    int i, j;

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

        //Input Name & IP Address
        printf("Enter Nickname of Computer %d: ", i+1);
        scanf("%s", arr[i].name);
        printf("Enter IP Address of Computer %d: ", i+1);
        scanf("%s", arr[i].ipa);
        printf("\n");
```

```c
      //Call split function on the IP Address
        int len = strlen(arr[i].ipa);
        j = 0;
        arr[i].xxx = split(arr[i].ipa, j, len);
        j += *y + 1;
        arr[i].yyy = split(arr[i].ipa, j, len);
    }
}

//Cmpare 2 IP Addresses
void compare(IP arr[], int n) {
    int i, j;

    for (i = 0; i < n - 1; i++) {
        for (j = i + 1; j < n; j++) {
            if ((arr[i].xxx == arr[j].xxx) && (arr[i].yyy == arr[j].yyy)) {
                printf("Machines %s and %s are on the same local
network.\n\n", arr[i].name, arr[j].name);
            }
        }
    }
}
//Main function
int main() {
    int n, i;
    printf("\nEnter number of computers: ");
    scanf("%d", &n);
    printf("\n");

    IP cndetails[n];
    get_data(cndetails, n);
    compare(cndetails, n);

    return 0;
}
```

## **Output:**


File Edit Selection View Go Run Terminal Help
DA1.c - DA1 DA2 - Visual Studio Code
DA1.c
DA1.c > compare(IP [], int)
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE
powershell
Ln 58, Col 78 Spaces: 4 UTF-8 CRLF C Go Live Win32 Prettier
Type here to search
31°C Mostly clear ENG 22:05 03-06-2022

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

//Structure Definition
typedef struct IPAddress{
    char ipa[15];
    int xxx,yyy;
    char name[30];
} IP;

//Split a part from IP Address
int *y;
int split(char ip[15], int j, int len) {
    char temp[3] = "";

    while ((ip[j]!='.') && (j != len)) {
        strncat(temp,&ip[j],1);
        j++;
    }

    int x = strlen(temp);
    y = &x;

    int num = atoi(temp);
    return num;
}

//Get input for all computer networks
void get_data(IP arr[], int n) {
    int i, j;

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

        //Input Name & IP Address
        printf("Enter Nickname of Computer %d: ", i+1);
```

```
PS C:\Users\Vidhi Shah\Desktop\VITC Sem 2\C C++ Theory\DA1 DA2> g++ -o DA1 DA1.c
PS C:\Users\Vidhi Shah\Desktop\VITC Sem 2\C C++ Theory\DA1 DA2> ./DA1

Enter number of computers: 4

Enter Nickname of Computer 1: patte
Enter IP Address of Computer 1: 111.22.3.44

Enter Nickname of Computer 2: wabash
Enter IP Address of Computer 2: 555.66.7.88

Enter Nickname of Computer 3: green
Enter IP Address of Computer 3: 111.22.5.66

Enter Nickname of Computer 4: none
Enter IP Address of Computer 4: 0.0.0.0

Machines patte and green are on the same local network.

PS C:\Users\Vidhi Shah\Desktop\VITC Sem 2\C C++ Theory\DA1 DA2> ./DA1

Enter number of computers: 3

Enter Nickname of Computer 1: Atlas
Enter IP Address of Computer 1: 101.33.2.1

Enter Nickname of Computer 2: Horizon
Enter IP Address of Computer 2: 101.33.56.80

Enter Nickname of Computer 3: Pluto
Enter IP Address of Computer 3: 101.43.45.74

Machines Atlas and Horizon are on the same local network.

PS C:\Users\Vidhi Shah\Desktop\VITC Sem 2\C C++ Theory\DA1 DA2>
```

## Q2

### Aim:

In a library, the books are arranged vertically in the rack, one above the other. The book(s) can be added or removed only from the top and not in the middle. You have been assigned to add 10 books and remove the books until the book rack is empty. Develop the program using generic function.

### Procedure:

**Input:**

Title of 10 books

**Output:**

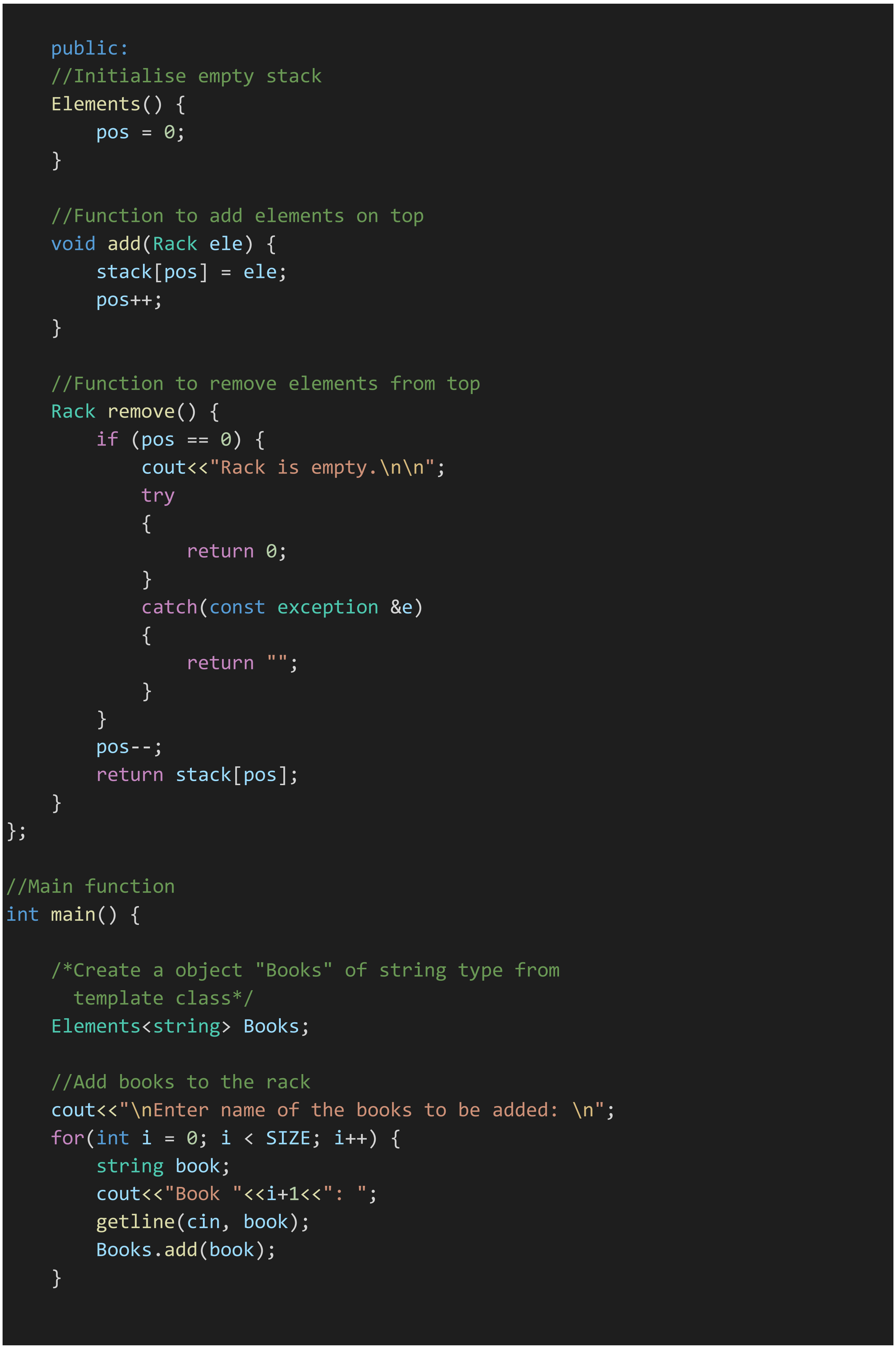
Books removed from top to bottom until rack is empty

**Algorithm:**

```
Step 1: Create a template ‘Rack’ with class ‘Elements’
Step 2: Private data members: Array ‘stack’ of type Rack and integer type
        variable ‘pos’
Step 3: Initialise ‘pos’ to 0 with constructor
Step 4: Public member functions
            Step A: ‘Add’ function to add elements on top
            Step B: ‘Remove’ function to remove elements from top
Step 5: Main function
            Step A: Create object with string data type
            Step B: For loop to add books to rack
            Step C: While loop to remove books until rack is empty
```

### Code:

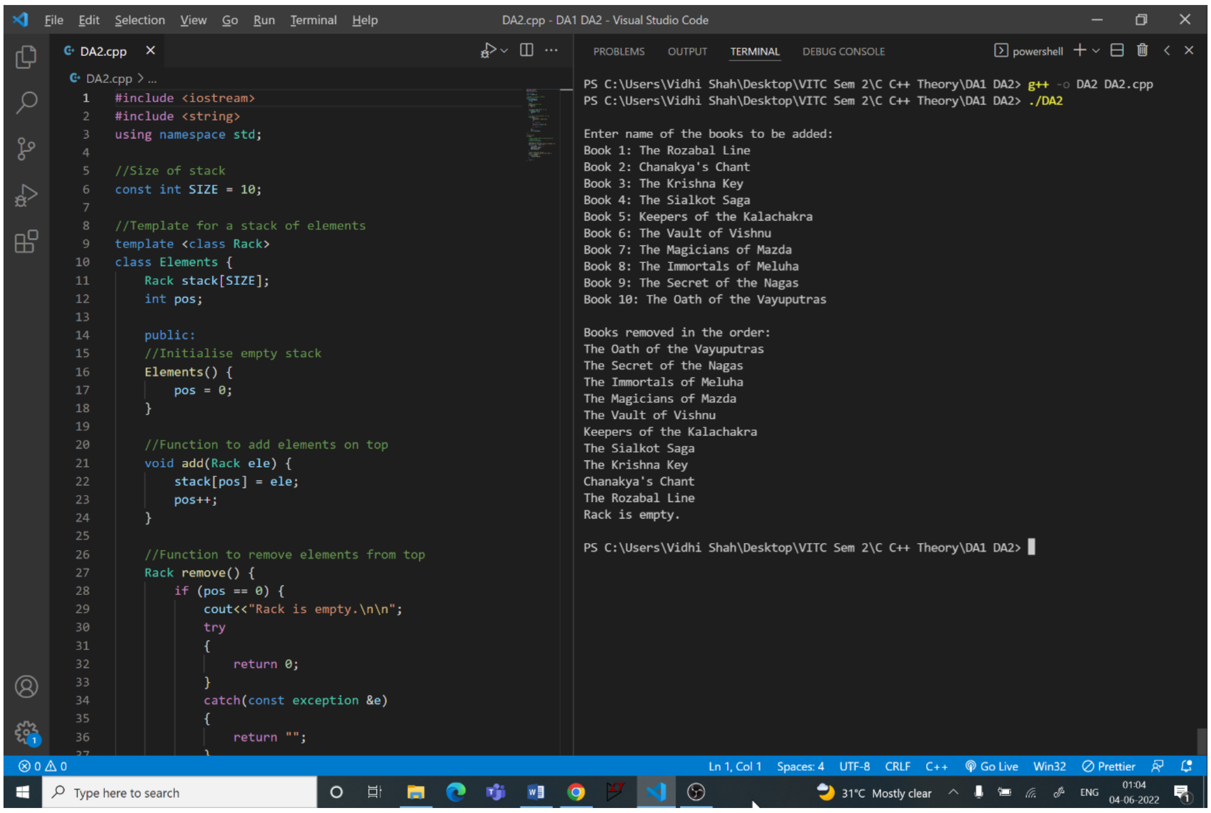
```
#include <iostream>
#include <string>
using namespace std;

//Size of stack
const int SIZE = 10;

//Template for a stack of elements
template <class Rack>
class Elements {
    Rack stack[SIZE];
    int pos;
```



```
  public:
  //Initialise empty stack
  Elements() {
    pos = 0;
  }

  //Function to add elements on top
  void add(Rack ele) {
    stack[pos] = ele;
    pos++;
  }

  //Function to remove elements from top
  Rack remove() {
    if (pos == 0) {
      cout<<"Rack is empty.\n\n";
      try
      {
        return 0;
      }
      catch(const exception &e)
      {
        return "";
      }
    }
    pos--;
    return stack[pos];
  }
};

//Main function
int main() {

  /*Create a object "Books" of string type from
   template class*/
  Elements<string> Books;

  //Add books to the rack
  cout<<"\nEnter name of the books to be added: \n";
  for(int i = 0; i < SIZE; i++) {
    string book;
    cout<<"Book "<<i+1<<": ";
    getline(cin, book);
    Books.add(book);
  }
```

```cpp
//Remove books from the rack
    string r = "\nBooks removed in the order:";
    while (!r.empty()) {
        cout<<r<<endl;
        r = Books.remove();
    }

    return 0;
}
```

## Output:



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

//Size of stack
const int SIZE = 10;

//Template for a stack of elements
template <class Rack>
class Elements {
    Rack stack[SIZE];
    int pos;

    public:
    //Initialise empty stack
    Elements() {
        pos = 0;
    }

    //Function to add elements on top
    void add(Rack ele) {
        stack[pos] = ele;
        pos++;
    }

    //Function to remove elements from top
    Rack remove() {
        if (pos == 0) {
            cout<<"Rack is empty.\n\n";
            try
            {
                return 0;
            }
            catch(const exception &e)
            {
                return "";
```

```
PS C:\Users\Vidhi Shah\Desktop\VITC Sem 2\C C++ Theory\DA1 DA2> g++ -o DA2 DA2.cpp
PS C:\Users\Vidhi Shah\Desktop\VITC Sem 2\C C++ Theory\DA1 DA2> ./DA2

Enter name of the books to be added:
Book 1: The Rozabal Line
Book 2: Chanakya's Chant
Book 3: The Krishna Key
Book 4: The Sialkot Saga
Book 5: Keepers of the Kalachakra
Book 6: The Vault of Vishnu
Book 7: The Magicians of Mazda
Book 8: The Immortals of Meluha
Book 9: The Secret of the Nagas
Book 10: The Oath of the Vayuputras

Books removed in the order:
The Oath of the Vayuputras
The Secret of the Nagas
The Immortals of Meluha
The Magicians of Mazda
The Vault of Vishnu
Keepers of the Kalachakra
The Sialkot Saga
The Krishna Key
Chanakya's Chant
The Rozabal Line
Rack is empty.

PS C:\Users\Vidhi Shah\Desktop\VITC Sem 2\C C++ Theory\DA1 DA2>
```