



Sri Lanka Institute of Information Technology

Foundations of Algorithms – IE2072

Submitted by:

Student Registration Number	Student Name
IT19065236	Maddumage M

Date of submission: 29 / 05 / 2021

Q1 –

Question 1

Given a string containing digits from 2-9 inclusive, return all possible letter combinations that the number could represent. Return the answer in any order.

A mapping of digit to letters (just like on the telephone buttons) is given below. Note that 1 does not map to any letters.



Example 1:

Input: digits = "23"

Output: ["ad","ae","af","bd","be","bf","cd","ce","cf"]

Example 2:

Input: digits = ""

Output: []

Example 3:

Input: digits = "2"

Output: ["a","b","c"]

Constraints:

- $0 \leq \text{digits.length} \leq 4$
- $\text{digits}[i]$ is a digit in the range ['2', '9'].

Q1 Source code :

Source code developed using C++ language

```
#include<iostream>
```

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
#include<stdlib.h>
```

```
#include<vector>
```

```
//declaring functions
```

```
void vecprint(vector<string> &pp);
```

```
void create_combination(string num, vector<string>&result, string  
pac[], int c, string l);
```

```
vector<string> comb(string num);
```

```
int main()
```

```
{  
    //declaring variables  
  
    string num;  
  
    //get values from the input  
  
    cout << "enter a number : ";  
  
    cin >> num ;  
  
    //calling the comb function to create combinations.  
  
    comb(num);  
  
    return 0;  
}
```

```
//print values that are stores in the vector  
void vecprint(vector<string> &pp)  
{  
    for(int i = 0 ; i < pp.size() ; i++)  
    {
```

```
        cout << pp.at(i) << "\n";

    }

}

//create letter combinations.

void create_combination(string num, vector<string>&result, string
pac[], int c, string l)
{

    if(c == num.length())
    {

        //insert results to result vector.

        result.push_back(l);

        //calling printing functions to print the values of the vector

        vecprint(result);
```

```
        return;  
    }
```

```
// used 0 to remove integer values from the string
```

```
string word = pac[num[c] - '0'];
```

```
for(int j = 0 ; j < word.length() ; j++)
```

```
{
```

```
    //implementing the recursive function
```

```
    create_combination(num, result, pac, c+1, l+word[j]);
```

```
}
```

```
}
```

```
vector<string> comb(string num)
```

```
{
```

```
    //declaring a vector variable
```

```
    vector<string>result;
```

```
//map data to a string array  
  
string pace[10] =  
{ "", "", "abc", "def", "ghi", "jkl", "mno", "pqrs", "tuv", "wxyz" };
```

```
//check the length of the number is 0 or not.
```

```
if(num.length()==0)
```

```
{
```

```
    return result;
```

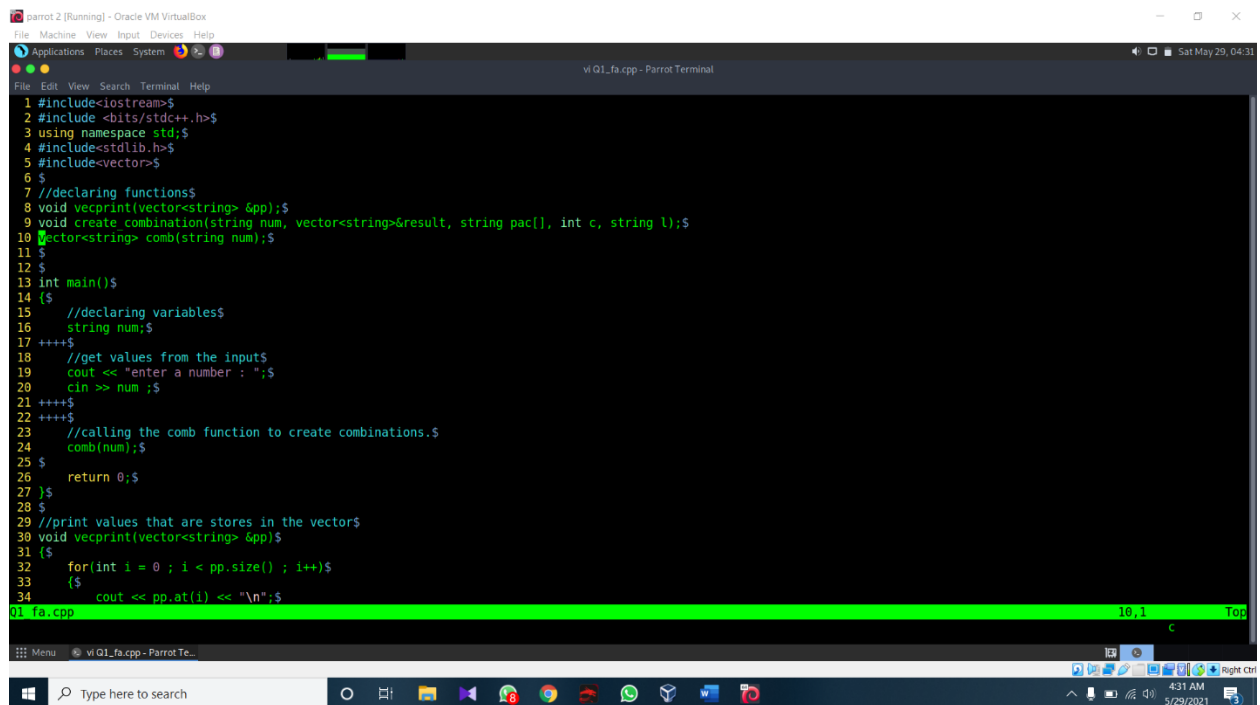
```
}
```

```
create_combination(num, result, pace, 0, "");
```

```
return result;
```

```
}
```

Image of source code



The image shows a screenshot of a Parrot VM terminal window titled "parrot 2 (Running) - Oracle VM VirtualBox". The terminal displays the source code for a C++ program named "Q1_fa.cpp". The code includes headers for `<iostream>`, `<bits/stdc++.h>`, `<vector>`, and `<string>`. It defines a `vecprint` function to print vector elements, a recursive `comb` function to generate combinations, and a `main` function that prompts the user for a number and prints the generated combinations. The code is as follows:

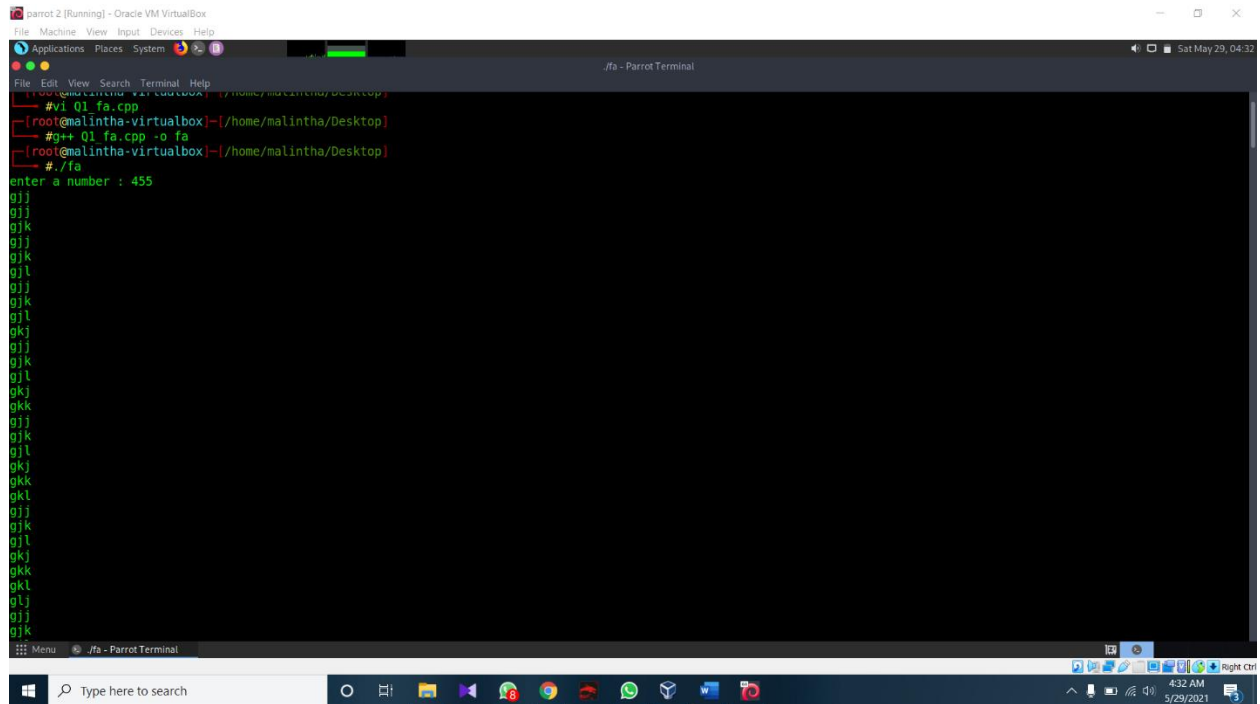
```
1 #include<iostream>$
2 #include <bits/stdc++.h>$
3 using namespace std;$
4 #include<stdlib.h>$
5 #include<vector>$
6 $
7 //declaring functions$
8 void vecprint(vector<string> &pp);$
9 void create combination(string num, vector<string>&result, string pac[], int c, string l);$
10 vector<string> comb(string num);$
11 $
12 $
13 int main()$
14 {$
15     //declaring variables$
16     string num;$
17     +++$
18     //get values from the inputs$
19     cout << "enter a number : ";$
20     cin >> num;$
21     +++$
22     +++$
23     //calling the comb function to create combinations.$
24     comb(num);$
25     $
26     return 0;$
27 }$
28 $
29 //print values that are stores in the vector$
30 void vecprint(vector<string> &pp)$
31 {$
32     for(int i = 0 ; i < pp.size() ; i++)$
33     {$
34         cout << pp.at(i) << "\n";$
35     }
```

The terminal window also shows a status bar at the bottom with the file name "Q1_fa.cpp", line numbers "10,1", and a "Top" button. The Windows taskbar is visible at the bottom of the screen, showing the time as 4:31 AM on 5/29/2021.


```
parrot.2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
vi Q1_fa.cpp - Parrot Terminal
File Edit View Search Terminal Help
34 cout << pp.at(i) << "\n";
35 +$
36 }$
37 +++++$
38 }$
39 $
40 //create letter combinations.$
41 void create_combination(string num, vector<string>&result, string pac[], int c, string l)$
42 {$
43 +++++$
44     if(c == num.length())$
45     {$
46         ++++++$
47         //insert results to result vector.$
48         result.push_back(l);$
49         ++++++$
50         //calling printing functions to print the values of the vector$
51         vecprint(result);$
52         ++++++$
53         return;$
54     }$
55 ++++++$
56 ++++++$
57 ++++++$
58     // used 0 to remove integer values from the string$
59     string word = pac[num[c] - '0'];$
60 +++++$
61     for(int j = 0 ; j < word.length() ; j++)$
62     {++$
63         //implementing the recursive functions$
64         create_combination(num, result, pac, c+1, l+word[j]);$
65     }$
66 +++++$
67 }$
01 fa.cpp 34,33 43%
```

```
parrot.2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
vi Q1_fa.cpp - Parrot Terminal
File Edit View Search Terminal Help
64     create_combination(num, result, pac, c+1, l+word[j]);$
65 }$
66 +++++$
67 }$
68 $
69 vector<string> comb(string num)$
70 {$
71     //declaring a vector variable$
72     vector<string>result;$
73 +++++$
74     //map data to a string array$
75     string pace[10] = {"", "a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z"};$
76 +++++$
77 +++++$
78     //check the length of the number is 0 or not.$
79     if(num.length()==0)$
80     {$
81         return result;$
82     }$
83 +++++$
84     create_combination(num, result, pace, 0, "");$
85     return result;$
86 +++++$
87 +++++$
88 $
89 }+$
90 $
91 $
92 $
93 $
94 $
95 $
96 $
97 $
01 fa.cpp 94,0-1 82%
```

Output of Q1



The screenshot shows a Parrot VM terminal window titled "parrot-2 [Running] - Oracle VM VirtualBox". The terminal is running a C++ program. The user has entered the number 455. The program's output is a sequence of characters: gjj, gjj, gjk, gjj, gjk, gjl, gjj, gjk, gjl, gkj, gjj, gjk, gjl, gkj, gkk, gjj, gjk, gjl, gkj, gkk, gkl, gjj, gjk, gjl, gkj, gkk, gkl, glj, gjj, and gjk.

```
File Edit View Search Terminal Help
[parrot@malintha-virtualbox:~/home/malintha/Desktop]
#vi Q1_fa.cpp
[parrot@malintha-virtualbox:~/home/malintha/Desktop]
#g++ Q1_fa.cpp -o fa
[parrot@malintha-virtualbox:~/home/malintha/Desktop]
#./fa
enter a number : 455
gjj
gjj
gjk
gjj
gjk
gjl
gjj
gjk
gjl
gkj
gjj
gjk
gjl
gkj
gkk
gjj
gjk
gjl
gkj
gkk
gkl
gjj
gjk
gjl
gkj
gkk
gkl
glj
gjj
gjk
```

Question 2

Given an input string (s) and a pattern (p), implement regular expression matching with support for '.' and '*' where:

- '.' Matches any single character.
- '*' Matches zero or more of the preceding element.

The matching should cover the entire input string (not partial).

Example 1:

Input: s = "aa", p = "a"

Output: false

Explanation: "a" does not match the entire string "aa".

Example 2:

Input: s = "aa", p = "a*"

Output: true

Explanation: '*' means zero or more of the preceding element, 'a'. Therefore, by repeating 'a' once, it becomes "aa".

Example 3:

Input: s = "ab", p = ".*"

Output: true

Explanation: ".*" means "zero or more (*) of any character (.)".

Example 4:

Input: s = "aab", p = "c*a*b"

Output: true

Explanation: c can be repeated 0 times, a can be repeated 1 time. Therefore, it matches "aab".

Example 5:

Input: s = "mississippi", p = "mis*is*p*."

Output: false

Constraints:

- $0 \leq s.length \leq 20$
- $0 \leq p.length \leq 30$
- s contains only lowercase English letters.
- p contains only lowercase English letters, '.', and '*'.
- It is guaranteed for each appearance of the character '*', there will be a previous valid character to match.

Q2 Source code

```
#include <stdio.h>
```

```
#include<iostream>
```

```
#include<string.h>
```

```
using namespace std;
```

```
//declaring the function
```

```
bool checker(string s, string p);
```

```
bool checkchars(string s);
```

```
bool checkpattn(string p);
```

```
int main()
```

```
{
```

```
    //declaring variables
```

```
    string s,p;
```

```
    //inserting inputs for string and pattern.
```

```
    cout << "please enter the string : ";
```

```
    cin >> s;
```

```
    cout << "enter the pattern : ";
```

```
    cin >> p;
```

```
    // validating the user inputs by callin checkchars and checkpattn  
    functions.
```

```
    if(checkchars(s) == 1 || checkpattn(p) == 1 || p[0] == '*' || s.length() <  
    0 || s.length() > 20 || p.length() < 0 || p.length() > 30)
```

```
{
```

```
cout << "invalid input";
```

```
return -1;
```

```
}
```

```
else
```

```
{
```

```
    //calling the checker function to check wheather the number is  
    match for the pattern
```

```
    if(checker(s,p) == 0)
```

```
    {
```

```
        cout << "False \n";
```

```
    }
```

```
else
```

```
if(checker(s,p) == 1)
```

```
{
```

```
    cout << "True \n";
```

```
}
```

```
}
```

```
return 0;
```

```
}
```

```
//check the pattern matches the string
```

```
bool checker(string s, string p)
```

```
{
```

```
    if((p.size() > 1) && p[1] == '*')
```

```
    {
```

```
        //check recursive function is true
```

```
        if(checker(s,p.substr(2)))
```

```
        {
```

```
            return true;
```

```
}
```

```
if((p[0] == '.'||p[0] == s[0]) && s.length() > 0)
```

```
{
```

```
    //implementing recursive function
```

```
    return checker(s.substr(1),p);
```

```
}
```

```
return false;
```

```
}
```

```
if(p.size() == 0)
```

```
{
```

```
    return s.size() == 0;
```



```
}
```

```
else
```

```
{
```

```
    if((p[0] == '.' || p[0] == s[0]) && s.size() > 0)
```

```
    {
```

```
        //substr - use to remove the check character of the string.
```

```
        //example : s = abb - s.substr(1) = bb
```

```
        //implementing the recursive function
```

```
        return checker(s.substr(1),p.substr(1));
```

```
    }
```

```
return false;
```

```
}
```

```
}
```

```
//check all the character are in lowercase in the strings
```

```
bool checkchars(string s)
```

```
{
```

```
    for(int i = 0 ; i < s.length() ; i++)
```

```
    {
```

```
        if(s[i] < 97 || s[i] > 122)
```

```
        {
```

```
            return true;
```

```
        }
```

```
    }
```

```
}
```

```
//check all the character are in lowercase in pattern
```

```
bool checkpattn(string p)
```

```
{
```

```
for (int i = 0 ; i < p.length() ; i++)  
{  
    if (isupper(p[i]) || isdigit(p[i]))  
    {  
        return true;  
    }  
  
}  
  
}
```

Image of source code

```
parrot.2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
vi Q2_fa.cpp - Parrot Terminal
File Edit View Search Terminal Help
1 #include <stdio.h>$
2 #include <iostream>$
3 #include <string.h>$
4 using namespace std;$
5 $
6 //declaring the functions$
7 bool checker(string s, string p);$
8 bool checkchars(string s);$
9 bool checkpattn(string p);$
10 $
11 int main()$
12 {$
13     //declaring variables$
14     string s,p;$
15     +++$
16     //inserting inputs for string and pattern.$
17     cout << "please enter the string : ";$
18     cin >> s;$
19     +++$
20     cout << "enter the pattern : ";$
21     cin >> p;$
22     +++$
23     // validating the user inputs by callin checkchars and checkpattn functions.+++$
24     if(checkchars(s) == 1 || checkpattn(p) == 1 || p[0] == '*' || s.length() < 0 || s.length() > 20 || p.length() < 0 || p.length() > 30)$
25     {$
26         ++++++$
27         cout << "invalid input";$
28         ++++++$
29         return -1;$
30     }$
31     +++$
32     +++$
33     else$
34     {$
02 fa.cpp 1.1 Top
Menu vi Q2_fa.cpp - Parrot T...
```

```
parrot.2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
vi Q2_fa.cpp - Parrot Terminal
File Edit View Search Terminal Help
37     return -1;$
38 }$
39 +++$
40 else$
41 {$
42     //calling the checker function to check wheather the number is match for the pattern$
43     if(checker(s,p) == 0)$
44     {$
45         cout << "False \n";$
46     }$
47     ++++++$
48     else$
49     if(checker(s,p) == 1)$
50     {$
51         cout << "True \n";$
52     }$
53     ++++++$
54     ++++++$
55     ++++++$
56 }$
57 $
58     return 0;$
59 }$
60 $
61 $
62 //check the pattern matches the strings$
63 bool checker(string s, string p)$
64 {$
65     if((p.size() > 1) && p[1] == '*')$
66     {$
67         //check recursive function is true+$
68         if(checker(s,p.substr(2)))$
69         {$
70             ++++++$
02 fa.cpp 40,8 32%
```



```
112 ++++$
113 }$
114 $
115 $
116 //check all the character are in lowercase in the strings$
117 bool checkchars(string s)$
118 {$
119     for(int i = 0 ; i < s.length() ; i++)$
120     {$
121         if(s[i] < 97 || s[i] > 122)$
122         {$
123             return true;$
124         }$
125     }$
126 }$
127 $
128 $
129 //check all the character are in lowercase in pattern$
130 bool checkpattn(string p)$
131 {$
132     for (int i = 0 ; i < p.length() ; i++)$
133     {$
134         if (isupper(p[i]) || isdigit(p[i]))$
135         {$
136             return true;$
137         }$
138     }$
139 }$
140 ++++$
141 }$
142 $
143 $
144 $
~
Q2_fa.cpp 115,0-1 Bot
```

Output of Q2

```
[root@malintha-virtualbox]~/home/malintha/Desktop
#g++ Q2_fa.cpp -o fa2
Q2_fa.cpp: In function 'bool checkchars(std::string)':
Q2_fa.cpp:127:1: warning: control reaches end of non-void function [-Wreturn-type]
127 | }
    | ^
Q2_fa.cpp: In function 'bool checkpattn(std::string)':
Q2_fa.cpp:141:1: warning: control reaches end of non-void function [-Wreturn-type]
141 | }
    | ^
[root@malintha-virtualbox]~/home/malintha/Desktop
#./fa2
please enter the string : tiyy
enter the pattern : t..*
True
[root@malintha-virtualbox]~/home/malintha/Desktop
#
```