Department of Computer Science & Engineering
Faculty of Engineering & Technology (ITER)

MAJOR ASSIGNMENT-0
UNIX Network Programming (CSE 4042)

Problem Statement

Computation of all valid IPV4 addresses

In this assignment, you are required to compute all the valid combination of IPV4 addesses. A decimal string is a string consisting of digits between 0 and 9. Internet protocol (IP) version-4 addresses can be written as four decimal strings separated by periods, e.g., 192.168.1.201. A careless programmer mangles a string representing an IP address in such a way that all the periods vanish.

Synopsis:

Write a program that determines where to add periods to a decimal string so that the resulting string is a valid IP address. There may be more than one valid IP address corresponding to a string, in which case you should print all possibilities. Additionally your program must validate that neither group must not exceed 255.

For example:

If the mangled string is “19216811” then two corresponding IP addresses are 192.168.1.1 and 192.216.81.1.There are also seven other possible IP addresses for this string.

Solution hint:

There are three periods in a valid IP address, so you can enumerate all possible placements of these periods, and check whether all four corresponding substrings are between 0 and 255. You can reduce the number of placements considered by spacing the periods 1 to 3 characters apart. You can also prune by stopping assoon as a substring is not valid.

For example, if the string is “19216811”, you could put the first period after “1”, “19”, and “192”. If the first part is “1”, the second part could be “9”, “92”, and “921”. Of these, “921” is illegal so you do not continue with it.

Name: Ashutosh Palit Regd. Number: 1941012250

Soln:-

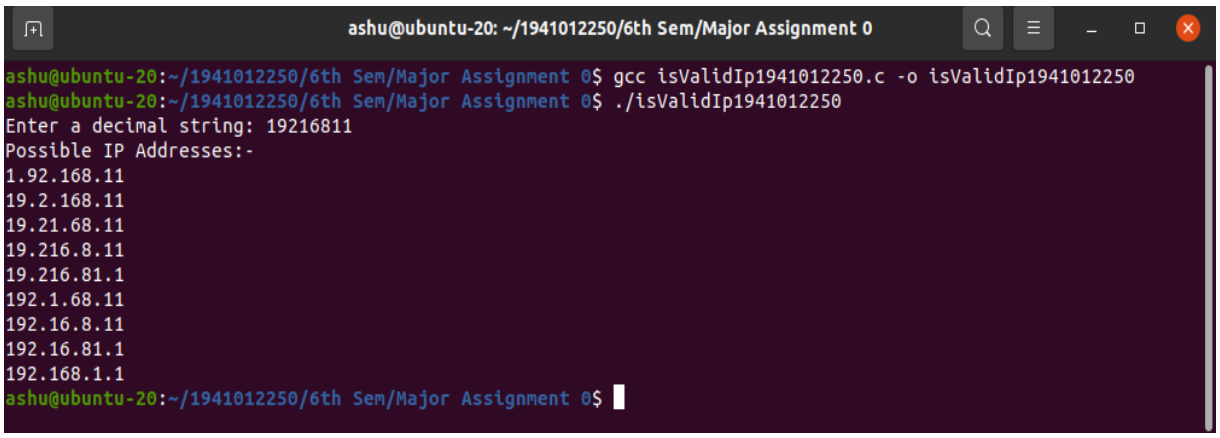
```
#include <stdio.h>
#include <string.h>
#include <stdbool.h>
#include <stdlib.h>
#include <ctype.h>

bool isValidDecimalString(char []);
bool isValidOctet(char []);
char substring(char [], char *, int, int);
char printValidIPAddresses(char *);

int main()
{
    char dec_str[20] = "";
    printf("Enter a decimal string: ");
    scanf("%[^\n]s", dec_str);
    while (true)
    {
        if (isValidDecimalString(dec_str))
        {
            break;
        }
        else
        {
            printf("Invalid Decimal string!!!\n");
            char dec_str[20] = "";
            printf("Please enter a decimal string again: ");
            scanf("%s", dec_str);
        }
    }
    printf("Possible IP Addresses:-\n");
    printValidIPAddresses(dec_str);
    return 0;
}

char substring(char dest[], char *src, int start, int end)
{
    int j = 0;
    for (int i = start; i < end && (*(src + i) != '\0'); i++)
    {
        dest[j++] = *(src + i);
    }
    dest[j] = '\0';
}
```

```
char printValidIPAddresses(char *str)
{
    int resultIndex = 0;
    for (int i = 1; i < 4 && i < strlen(str); ++i)
    {
        char firstOctet[10] = "";
        substring(firstOctet, str, 0, i);
        if (isValidOctet(firstOctet))
        {
            for (int j = 1; (i + j) < strlen(str) && j < 4; ++j)
            {
                char secondOctet[10] = "";
                substring(secondOctet, str, i, i + j);
                if (isValidOctet(secondOctet))
                {
                    for (int k = 1; (i + j + k) < strlen(str) &&
k < 4; ++k)
                    {
                        char thirdOctet[10] = "";
                        substring(thirdOctet, str, i + j, i + j +
k);
                        char fourthOctet[10] = "";
                        substring(fourthOctet, str, i + j + k,
strlen(str));
                        if (isValidOctet(thirdOctet) &&
isValidOctet(fourthOctet))
                        {
                            char IP[20] = "";
                            strcat(IP, firstOctet);
                            strcat(IP, ".");
                            strcat(IP, secondOctet);
                            strcat(IP, ".");
                            strcat(IP, thirdOctet);
                            strcat(IP, ".");
                            strcat(IP, fourthOctet);
                            puts(IP);
                        }
                    }
                }
            }
        }
    }
}
```

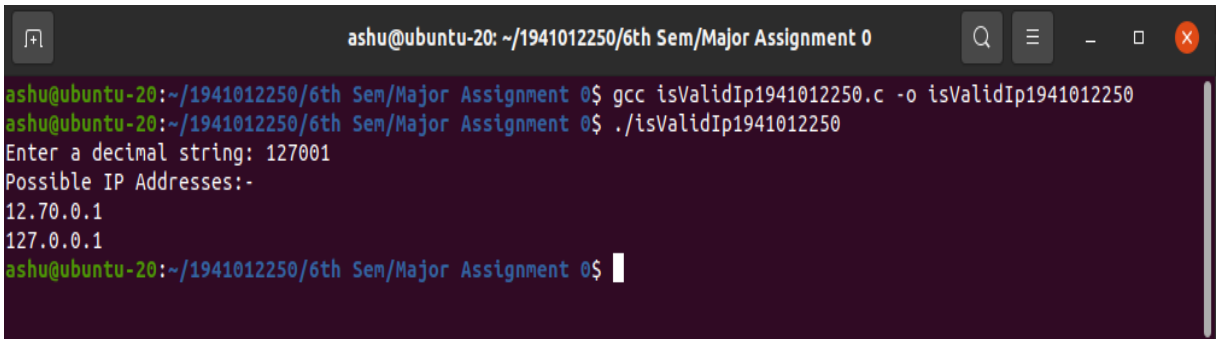
```
bool isValidOctet(char str[])
{
    if (strlen(str) > 3 || strlen(str) == 0)
    {
        return false;
    }
    if (str[0] == '0' && strlen(str) > 1)
    {
        return false;
    }
    int val = atoi(str);
    return (val >= 0) && (val <= 255);
}

bool isValidDecimalString(char str[])
{
    int len = strlen(str);
    for (int i = 0; i < len; i++)
    {
        if (isdigit(str[i]) == 0)
        {
            return false;
        }
    }
    return true;
}
```

Output:-

(i) For decimal string: 19216811



(ii) For decimal string: 127001



(ii) For decimal string: 17213425