VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CSE 1004

Network and Communication

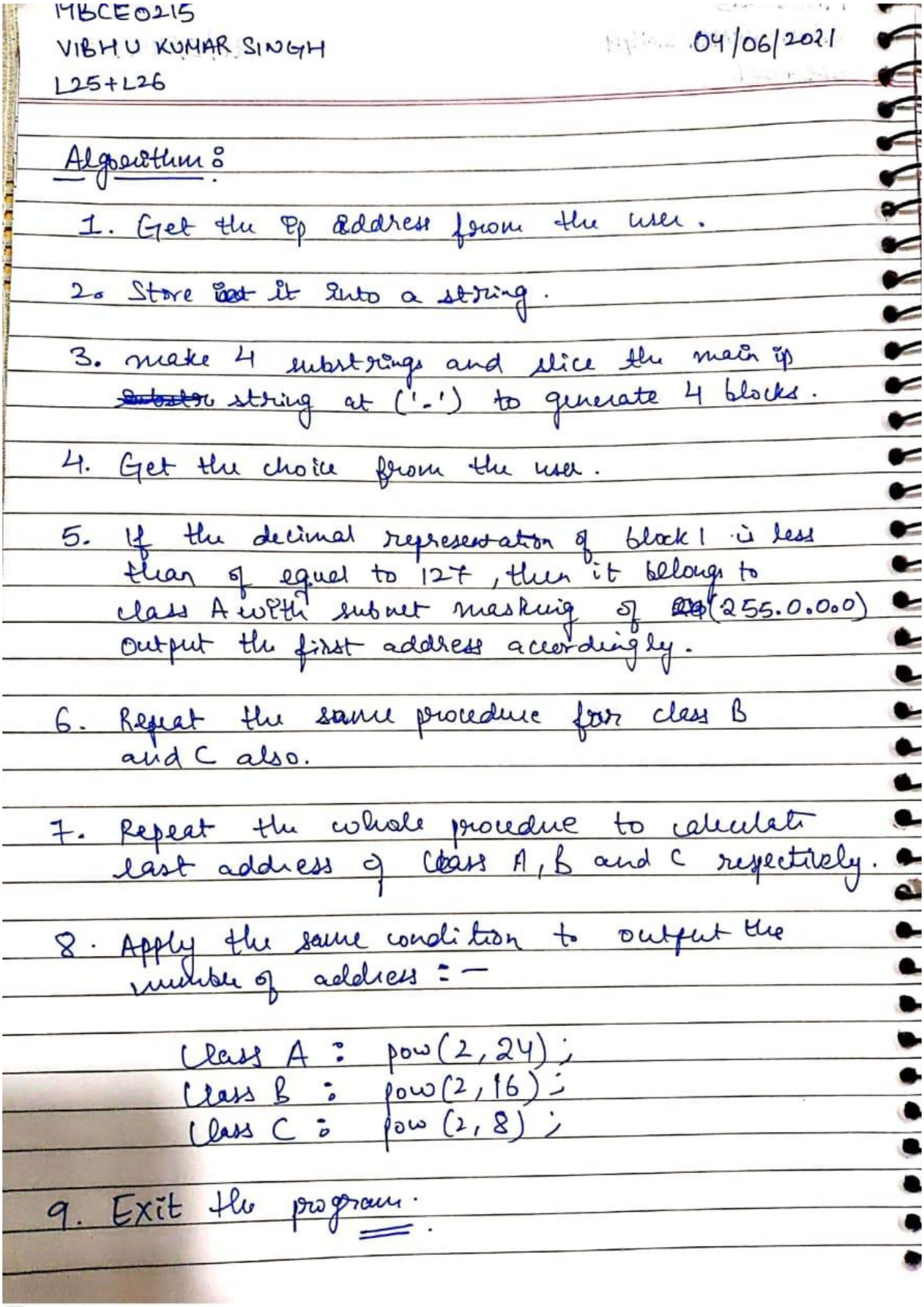
LAB FAT

NAME: Vibhu Kumar Singh
REG. NO: 19BCE0215
TEACHER: Santhi H.

Q2-a) Develop a menu driven code to find the first address, last address and number of addresses of Class A, B and C IP address.

Ans1-a)

Handwritten (Algorithm and Source Code):



17BCE0215

VIBHU KUMAR SINGH

04/06/2021

L25+L26

Algorithm :

1. Get the ip address from the user.

2. Store it into a string.

3. make 4 substrings and slice the main ip ~~substr~~ string at ('.') to generate 4 blocks.

4. Get the choice from the user.

5. If the decimal representation of block 1 is less than or equal to 127, then it belongs to class A with subnet masking of (255.0.0.0) output the first address accordingly.

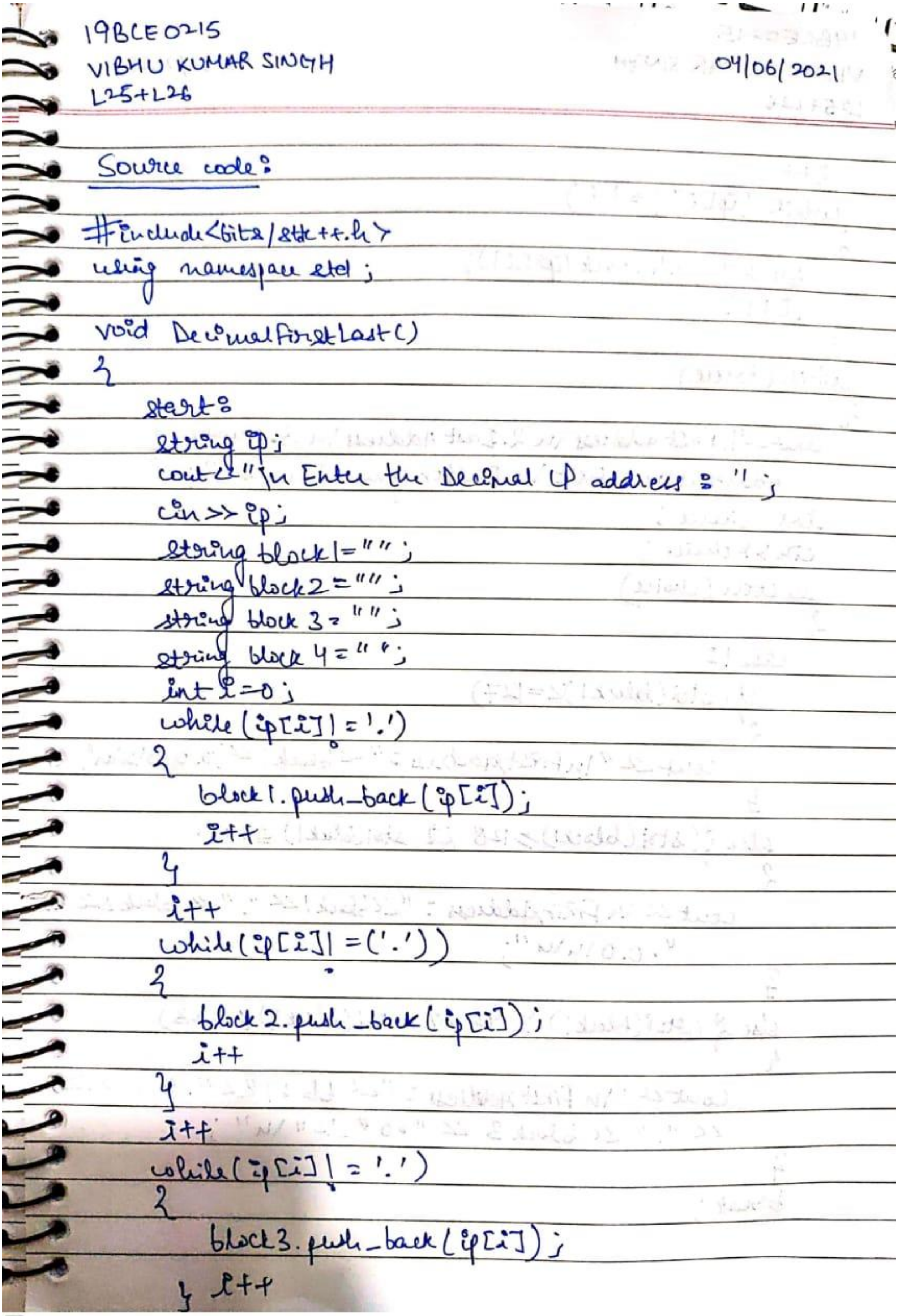
6. Repeat the same procedure for class B and C also.

7. Repeat the whole procedure to calculate last address of class A, B and C respectively.

8. Apply the same condition to output the number of address :-

Class A : pow(2,24);
Class B : pow(2,16);
Class C : pow(2,8);

9. Exit the program.



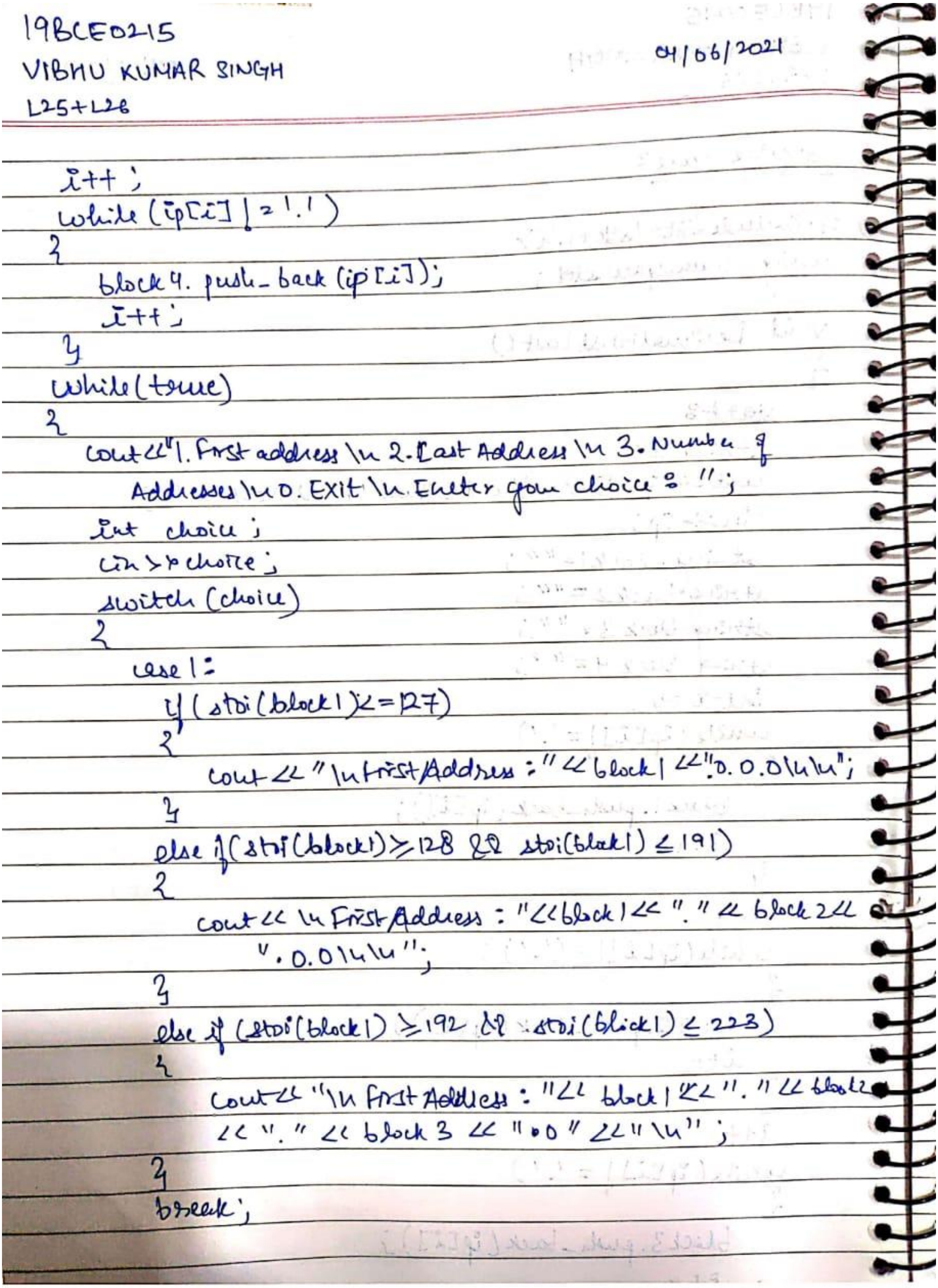
19BCE0215
VIBHU KUMAR SINGH
L25+L26

04/06/2021

Source code:

```
#include<bits/stdc++.h>
using namespace std;

void DecimalFirstLast()
{
    start:
    string ip;
    cout<<"\n Enter the Decimal IP address : ";
    cin>>ip;
    string block1="";
    string block2="";
    string block 3= "";
    string block 4=" ";
    int i=0;
    while(ip[i]!='.')
    {
        block1.push_back(ip[i]);
        i++
    }
    i++
    while(ip[i]!=('.'))
    {
        block2.push_back(ip[i]);
        i++
    }
    i++
    while(ip[i]!='.')
    {
        block3.push_back(ip[i]);
    } i++
```

Scanned with CamScanner

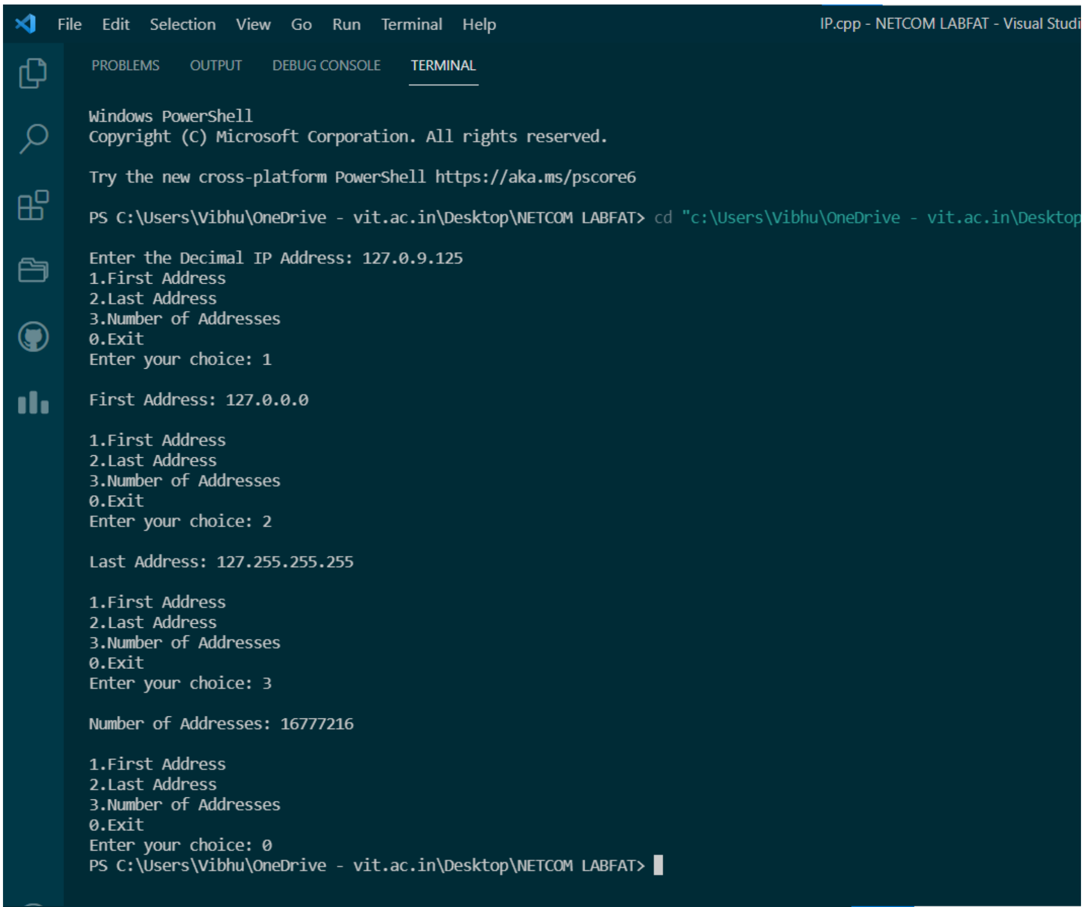


19BCE0215
VIBHU KUMAR SINGH
L25+L26

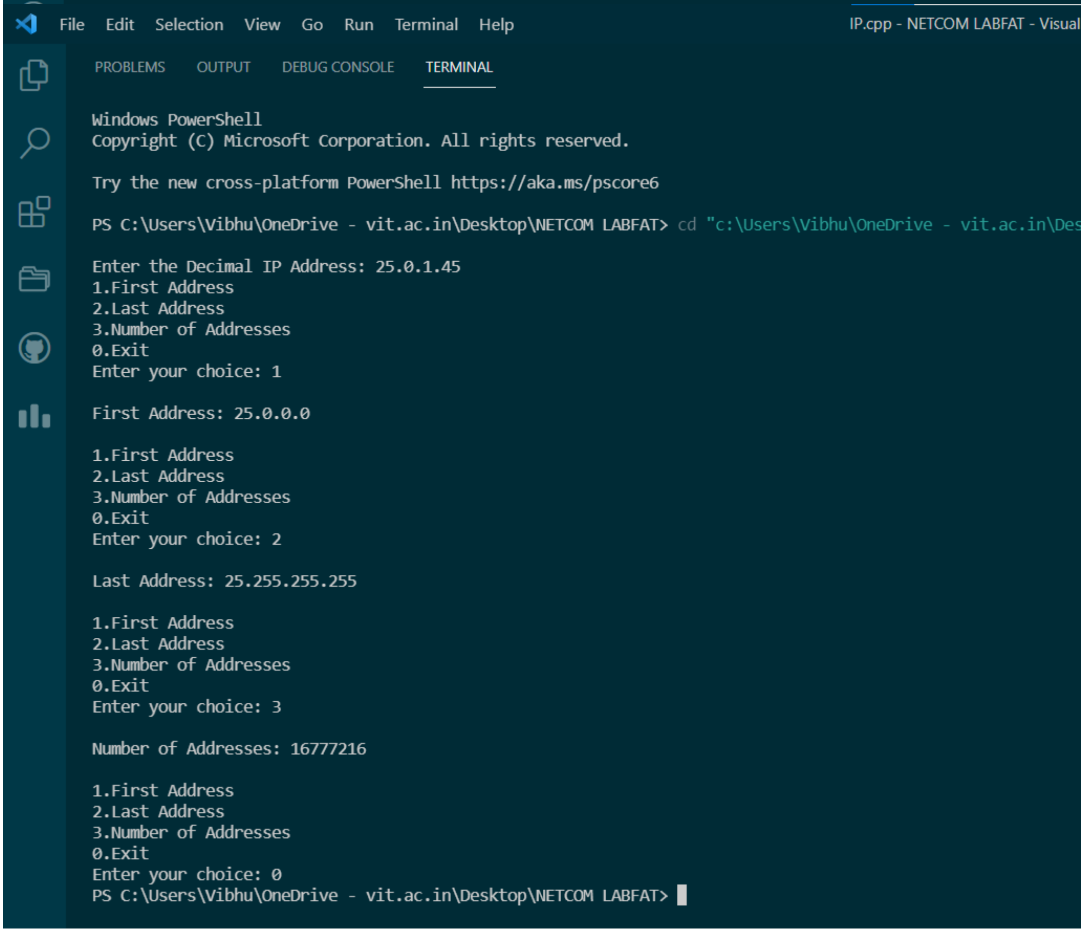
04/06/2021

```
i++;
while(ip[i]!='.')
{
    block4.push_back(ip[i]);
    i++;
}
while(true)
{
   cout<<"1. First address \n 2.Last Address \n 3.Number of
      Addresses\n0. EXIT \n. Enter your choice: ";
   int choice;
   cin>>choice;
   switch(choice)
   {
      case 1:
        if(stoi(block1)<=127)
        {
           cout<<"\nfrist Address :"<<block1<<".0.0.0\n\n";
        }
        else if(stoi(block1)≥128 && stoi(block1)≤191)
        {
           cout<< \n Frist Address :"<<block1<<"."<<block2<<
              ".0.0\n\n";
        }
        else if(stoi(block1)≥192 && stoi(block1)≤223)
        {
           cout<<"\n first Address :"<<block1<<"."<<block2
           <<"."<<block3<<".0"<<"\n";
        }
        break;
        case 2:
            // similiar conditions
            // cout .255 instead of .0

        case 3:
            // similar conditions
            // cout << (long long int) pow (2, 24);.. ✓

        case 0:
                exit(0);
                break;
        default :
                cout<<"invalid input" ;
                break;
    }
}

// Driver code :

  int main
  {
     DecimalfirstLast();
      return 0;
  }
```

OUTPUT:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Vibhu\OneDrive - vit.ac.in\Desktop\NETCOM LABFAT> cd "c:\Users\Vibhu\OneDrive - vit.ac.in\Desktop

Enter the Decimal IP Address: 127.0.9.125
1.First Address
2.Last Address
3.Number of Addresses
0.Exit
Enter your choice: 1

First Address: 127.0.0.0

1.First Address
2.Last Address
3.Number of Addresses
0.Exit
Enter your choice: 2

Last Address: 127.255.255.255

1.First Address
2.Last Address
3.Number of Addresses
0.Exit
Enter your choice: 3

Number of Addresses: 16777216

1.First Address
2.Last Address
3.Number of Addresses
0.Exit
Enter your choice: 0
PS C:\Users\Vibhu\OneDrive - vit.ac.in\Desktop\NETCOM LABFAT>
```

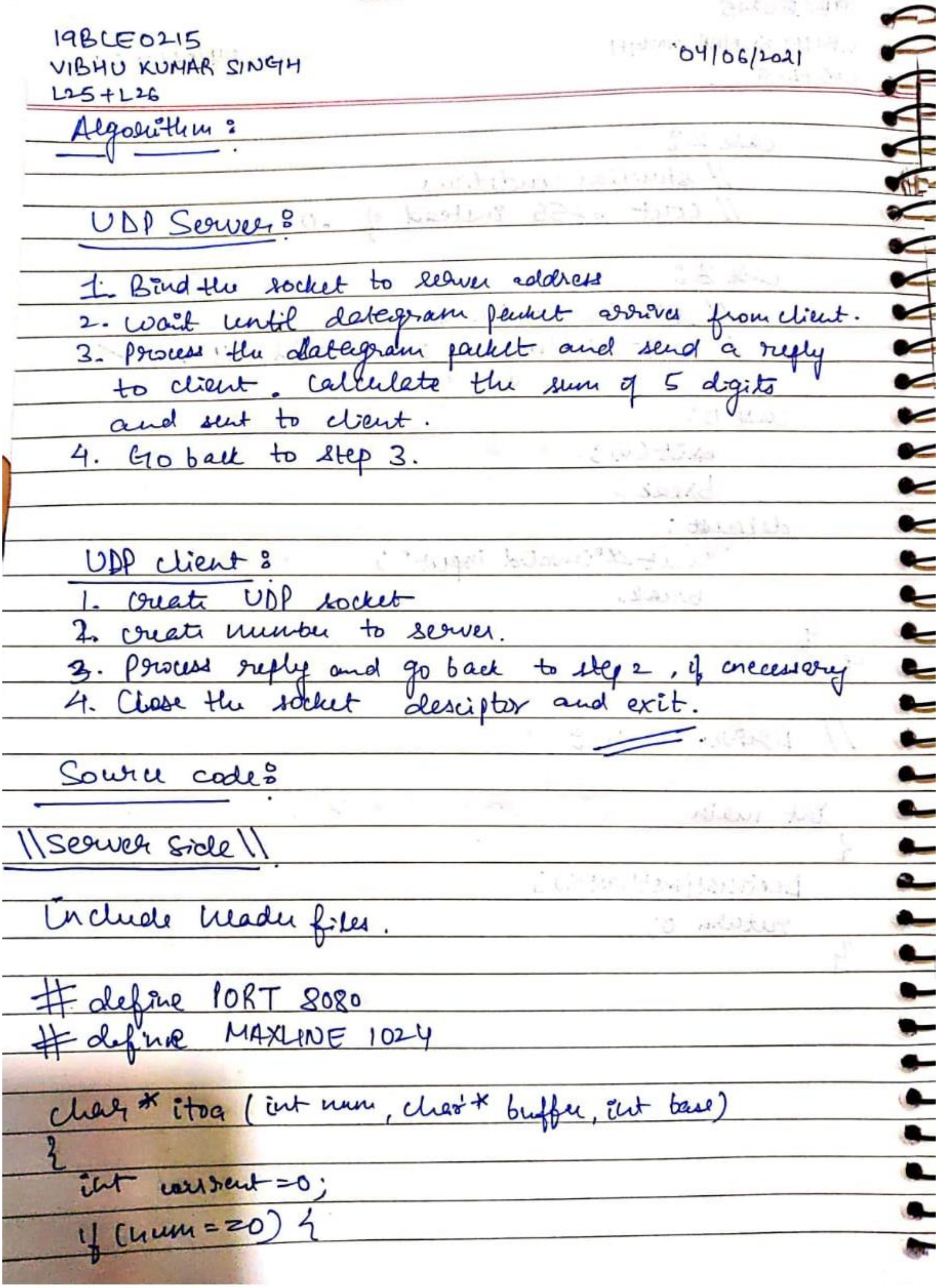


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Vibhu\OneDrive - vit.ac.in\Desktop\NETCOM LABFAT> cd "c:\Users\Vibhu\OneDrive - vit.ac.in\Des

Enter the Decimal IP Address: 25.0.1.45
1.First Address
2.Last Address
3.Number of Addresses
0.Exit
Enter your choice: 1

First Address: 25.0.0.0

1.First Address
2.Last Address
3.Number of Addresses
0.Exit
Enter your choice: 2

Last Address: 25.255.255.255

1.First Address
2.Last Address
3.Number of Addresses
0.Exit
Enter your choice: 3

Number of Addresses: 16777216

1.First Address
2.Last Address
3.Number of Addresses
0.Exit
Enter your choice: 0
PS C:\Users\Vibhu\OneDrive - vit.ac.in\Desktop\NETCOM LABFAT>
```

Q2-b) Develop a UDP client-server program to find the sum of any five digit number. The client program send any five digit integer number to the client and the server program reads and compute the sum of five digit received number and display the same at client side.

Ans2-b)

Handwritten (Algorithm and Source Code):



19BCE0215
VIBHU KUMAR SINGH
L25+L26

04/06/2021

Algorithm :

UDP Server :

1. Bind the socket to server address
2. Wait until datagram packet arrives from client.
3. Process the datagram packet and send a reply to client. Calculate the sum of 5 digits and sent to client.
4. Go back to step 3.

UDP client :

1. Creat UDP socket
2. creat number to server.
3. Process reply and go back to step 2, if cnecessary
4. Close the socket desciptor and exit.

Source code :

\\Server Side\\

Include header files.

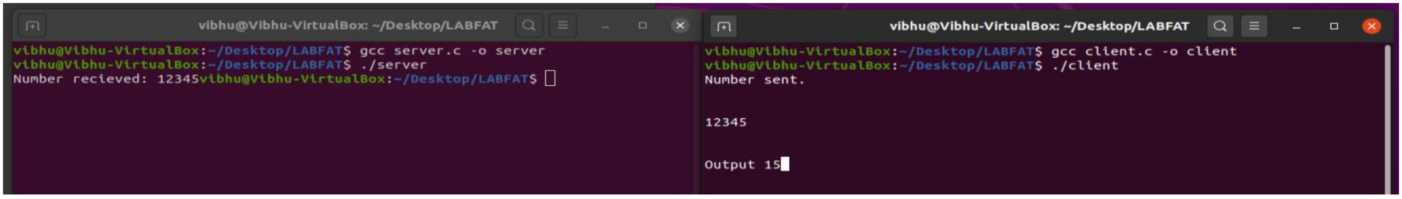
```
#define PORT 8080
#define MAXLINE 1024

char* itoa (int num, char* buffer, int base)
{
  int current=0;
  if (num==0) {
    buffer[current ++] = "0";
    buffer[current ] = "\0";
    return buffer;
  }
  int num_digits = 0;
  if (num <0){
    if (base == 10) {
     num_digits ++
     buffer [current] = '-' ;
     current ++ ;
     num *= -1 ;
    }
    else
     return NUll;
   num_digits += (int)floor( log(num)/ log(base)) + 1;
   while (current < num_digits)
   {
      int base_val = (int) pow(base, num_digits -1 - current);
      int num_val = num / base_val;
      char value = num_val - '0';
      buffer[current] = value ;
      current ++
      num -= base_val * num_val;
   }
   buffer [current] = "\0";
   return buffer;
  }
```

\\ Client side \\

include header files

```
#define port 8080
#define MAXLINE 1024

int main(){
    int sockfd;
    char buffer[MAXLINE];
    char* hello = "Hello from client";
    struct sockaddr_in servaddr;

    if(sockfd = socket(AF_INET, SOCK_DGRAM, 0))<0){
        perror("socket creation failed");
        exit(EXIT_FAILURE);
    }

    memset(&servaddr, 0, sizeof(servaddr));

    servaddr.sin_family = AF_NET;
      "     .sin_port = htons(PORT);
      "     .sin_addr.s_addr = INADDR_ANY;

    int n, len;

    buffer[n] = "\0";
    close(sockfd);
    return 0;
}
```

Screenshots:



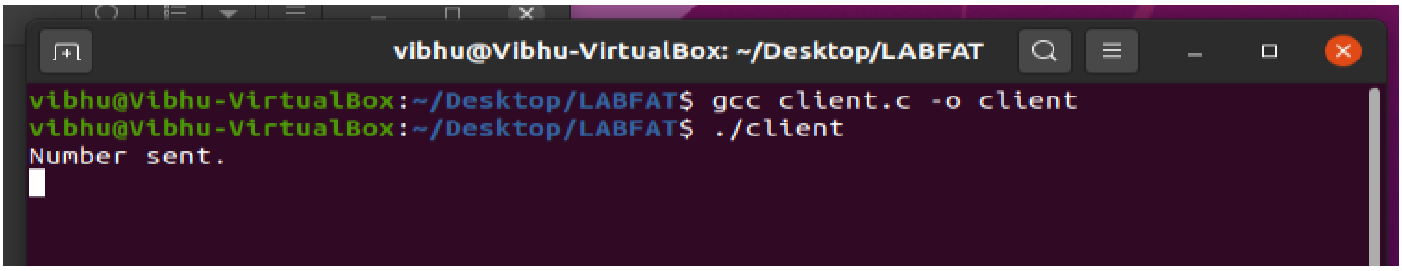
vibhu@Vibhu-VirtualBox: ~/Desktop/LABFAT

```
vibhu@Vibhu-VirtualBox:~/Desktop/LABFAT$ gcc server.c -o server
vibhu@Vibhu-VirtualBox:~/Desktop/LABFAT$ ./server
Number recieved: 12345vibhu@Vibhu-VirtualBox:~/Desktop/LABFAT$ 
```

vibhu@Vibhu-VirtualBox: ~/Desktop/LABFAT

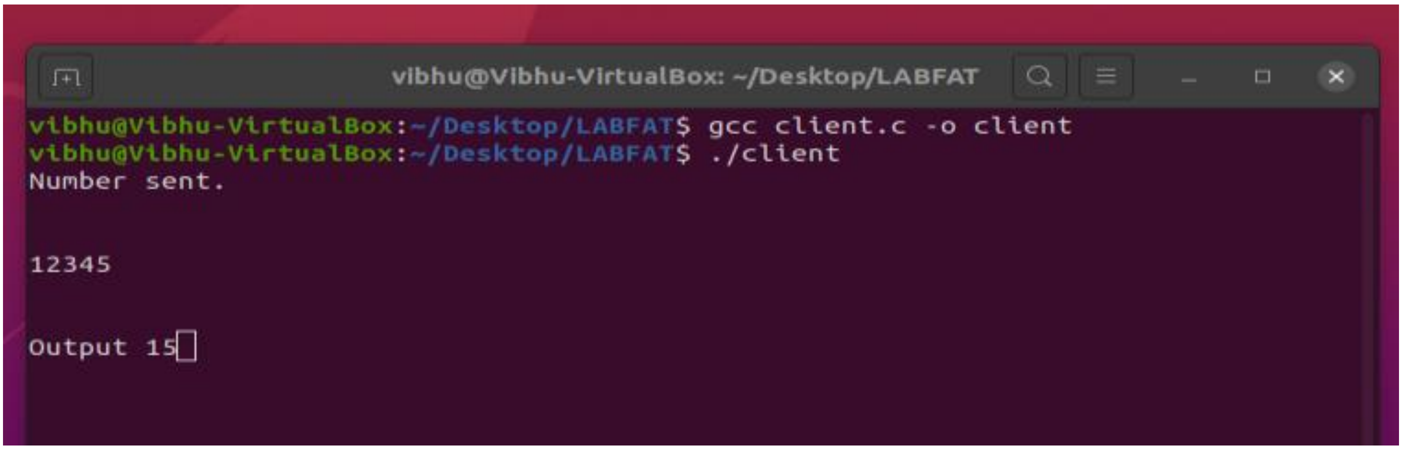
```
vibhu@Vibhu-VirtualBox:~/Desktop/LABFAT$ gcc client.c -o client
vibhu@Vibhu-VirtualBox:~/Desktop/LABFAT$ ./client
Number sent.


12345


Output 15
```

Client Side:



```
vibhu@Vibhu-VirtualBox:~/Desktop/LABFAT$ gcc client.c -o client
vibhu@Vibhu-VirtualBox:~/Desktop/LABFAT$ ./client
Number sent.
```

Server Side:



```
vibhu@Vibhu-VirtualBox:~/Desktop/LABFAT$ gcc client.c -o client
vibhu@Vibhu-VirtualBox:~/Desktop/LABFAT$ ./client
Number sent.


12345


Output 15
```