

Take a IPv4 address as input. Write a C program to check in which class does it belong. Also print special comment for network ID and broadcast ID.

Code

```
#include<stdio.h>

#include<string.h>

#include<stdlib.h>

void class_check(int *arr)

{    if(arr[0]>=0 && arr[0]<=127)

        {    printf("Class of Ip Address : A\n");

                printf("Network id= %d.0.0.0 \n",arr[0]);

                printf("Broadcast id= %d.255.255.255 \n",arr[0]);

        }

    else if(arr[0]>=128 && arr[0]<=191)

        {    printf("Class of Ip Address : B\n");

                printf("Network id= %d.%d.0.0 \n",arr[0],arr[1]);

                printf("Broadcast id= %d.%d.255.255 \n",arr[0],arr[1]);

        }

    else if(arr[0]>=192 && arr[0]<=223)

        {    printf("Class of Ip Address : C\n");

                printf("Network id= %d.%d.%d.0 \n",arr[0],arr[1],arr[2]);

                printf("Broadcast id= %d.%d.%d.255 \n",arr[0],arr[1],arr[2]);

        }

    else if(arr[0] >= 224 && arr[0] <= 239)

        {    printf("Class of Ip Address : D\n"); }

    else{

                printf("Class of Ip Address : E\n"); }}
```

```

int main()

{   char ipadd[50];

    char ch;

    gets(ipadd);

    int flag= 0;

//~~~~~

        // Check quadrate

//~~~~~

        int dot=0;

        int size = strlen(ipadd);

        for(int i=0;i<size;i++)

        {   if('.')==ipadd[i] dot++; }

        if(dot !=3)

        {   ch='N';

            //printf("Invalid Ip Address\n");

            flag=1;   }

//~~~~~

// check numbers or not

//~~~~~

        for (int i=0;i<size-1;i++)

        {   int num = ipadd[i];

            if((num<48 || num > 57) && num!=46)// 46 for dot (.)

            {   //printf("%d ",num);

                ch='N'; //printf("Invalid Ip Address\n");

                flag =1;

                break;   }   }

```

```

//~~~~~

// Check number range

//~~~~~

// 23.60.54.90

    // Extract the first token

    char * token = strtok(ipadd, "."); // token = 23

    int quadrate[4];

    // loop through the string to extract all other tokens

    for(int i=0; token != NULL;i++)

    {
        quadrate[i] = atoi(token); // 266

        //printf("%d ",quadrate[i]);

        if(quadrate[i]<0 || quadrate[i]>255)

        {
            ch='N';

            flag =1;

            break; }

        token = strtok(NULL, ".");

        // printf("%d\n",atoi(token));

// If valid Ipv4, send Y

    if(flag==0)

    {

        //printf("Reply '%c' sent to the client>>\n",ch);

        // Check Class

        class_check(quadrate);

    }

    else printf("Invalid Ip Address\n");

    printf("\n");}    return 0;}

```

```
intel@intel-HP-Notebook: ~/Documents/network_lab/exp_7
intel@intel-HP-Notebook:~/Documents/network_lab/exp_7$ gcc check.c -o check.o &&
./check.o
check.c: In function 'main':
check.c:39:2: warning: implicit declaration of function 'gets'; did you mean 'fgets'? [-Wimplicit-function-declaration]
   39 |     gets(ipadd);
      |     ^~~~~
      |     fgets
/usr/bin/ld: /tmp/ccXg5aTW.o: in function `main':
check.c:(.text+0x1b9): warning: the `gets' function is dangerous and should not
be used.
10.10.10.10
Class of Ip Address : A
Network id= 10.0.0.0
Broadcast id= 10.255.255.255

intel@intel-HP-Notebook:~/Documents/network_lab/exp_7$
```