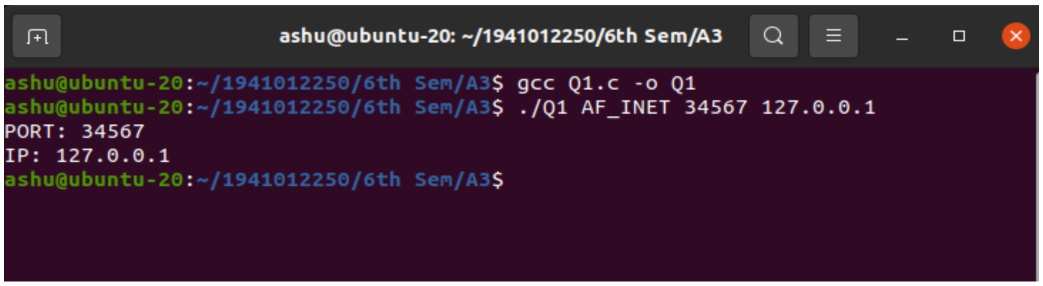
Minor Assignment 3

Q1. Write a program to create an IPV4 socket address structure and pack the structure with family `AF_INET`, port=34567 and IP address 127.0.0.1 respectively from the command line. Also, display the port and IP address.

Soln:-

```
#include <stdio.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
    struct sockaddr_in servaddr;
    servaddr.sin_family = atoi(argv[1]);
    servaddr.sin_port = htons(atoi(argv[2]));
    servaddr.sin_addr.s_addr = inet_addr(argv[3]);
    printf("PORT: %d\n", ntohs(servaddr.sin_port));
    printf("IP: %s\n", inet_ntoa(servaddr.sin_addr));
    return 0;
}
```

Output:-



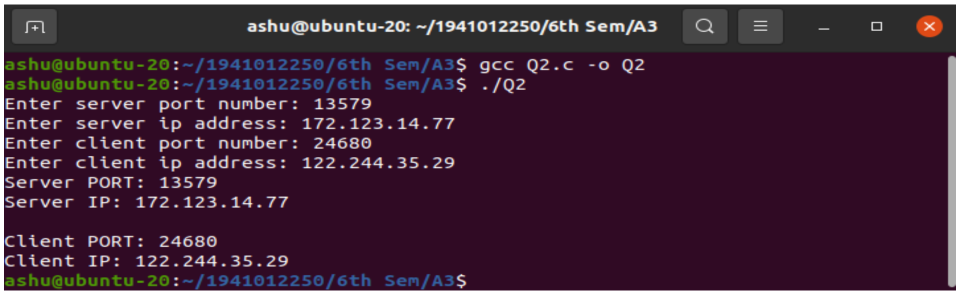
```
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ gcc Q1.c -o Q1
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q1 AF_INET 34567 127.0.0.1
PORT: 34567
IP: 127.0.0.1
ashu@ubuntu-20:~/1941012250/6th Sem/A3$
```

Q2. Write a program to declare two Internet socket address structure, namely `servaddr` and `cliaddr` respectively. Read the port and IP address for the structures you have declared from the keyboard and display the port and IP address.

Soln:-

```
#include <stdio.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdlib.h>
int main() {
    struct sockaddr_in servaddr;
    struct sockaddr_in cliaddr;
    char port_serv[16];
    char ip_serv[32];
    char port_cli[16];
    char ip_cli[32];
    printf("Enter server port number: ");
    scanf("%s", port_serv);
    printf("Enter server ip address: ");
    scanf("%s", ip_serv);
    servaddr.sin_family = AF_INET;
    servaddr.sin_port = htons(atoi(port_serv));
    servaddr.sin_addr.s_addr = inet_addr(ip_serv);

    printf("Enter client port number: ");
    scanf("%s", port_cli);
    printf("Enter client ip address: ");
    scanf("%s", ip_cli);
    cliaddr.sin_family = AF_INET;
    cliaddr.sin_port = htons(atoi(port_cli));
    cliaddr.sin_addr.s_addr = inet_addr(ip_cli);

    printf("Server PORT: %d\n", ntohs(servaddr.sin_port));
    printf("Server IP: %s\n\n", inet_ntoa(servaddr.sin_addr));
    printf("Client PORT: %d\n", ntohs(cliaddr.sin_port));
    printf("Client IP: %s\n", inet_ntoa(cliaddr.sin_addr));
    return 0;
}
```

Output:-



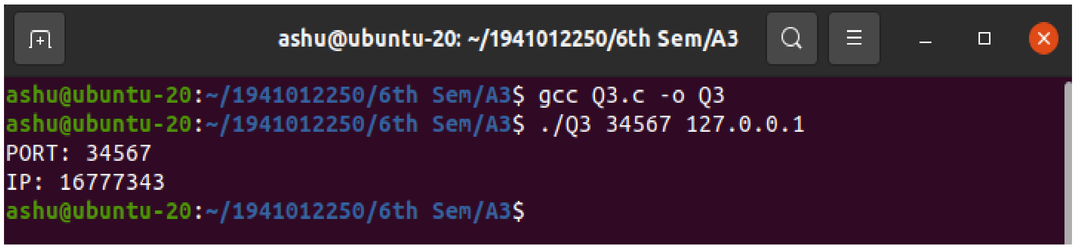
```
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ gcc Q2.c -o Q2
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q2
Enter server port number: 13579
Enter server ip address: 172.123.14.77
Enter client port number: 24680
Enter client ip address: 122.244.35.29
Server PORT: 13579
Server IP: 172.123.14.77

Client PORT: 24680
Client IP: 122.244.35.29
ashu@ubuntu-20:~/1941012250/6th Sem/A3$
```

Q3. Create a structure variable of the structure type **`struct sockaddr_in`** defined in the header<**`netinet/in.h`**>. Write a program to store the port and IP address in host byte order to net- work byte order from the command-line argument. Display the values of the structure variable, port and IP address, from network byte order to host byte order onto the monitor.

Soln:-

```
#include <stdio.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
    in_port_t port;
    in_addr_t ip;
    struct sockaddr_in sa;
    port = htons(atoi(argv[1]));
    ip = htonl(inet_addr(argv[2]));
    sa.sin_port = port;
    sa.sin_addr.s_addr = ip;
    printf("PORT: %hu\n", ntohs(sa.sin_port));
    printf("IP: %u\n", ntohl(sa.sin_addr.s_addr));
    return 0;
}
```

Output:-



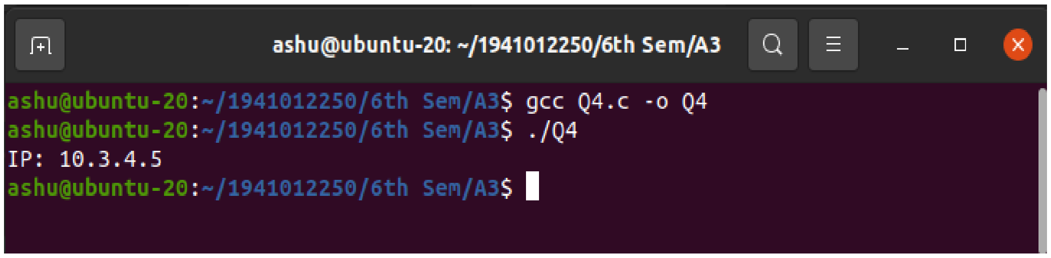
```
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ gcc Q3.c -o Q3
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q3 34567 127.0.0.1
PORT: 34567
IP: 16777343
ashu@ubuntu-20:~/1941012250/6th Sem/A3$
```

Q4. Assume that a line of code in your program is to read the IPV4 address in dotted decimal number and is stored in the network byte order as **`sa.sin addr.s addr=inet addr(‘‘10.3.4.5’’)`**, where **`sa`** is the structure variable of the type **`struct sockaddr in`**. You are required to rewritethe program to replace **`inet addr(...)`** with **`inet aton(...)`** to get the IP address and display that IP address using **`inet ntoa(..)`**.

Soln:-

```
#include <stdio.h>
#include <netinet/in.h>
#include <arpa/inet.h>
int main() {
    struct sockaddr_in sa;
    sa.sin_family = AF_INET;
    sa.sin_port = htons(34567);
    inet_aton("10.3.4.5", &sa.sin_addr);
    printf("IP: %s\n", inet_ntoa(sa.sin_addr));
    return 0;
}
```

Output:-



```
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ gcc Q4.c -o Q4
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q4
IP: 10.3.4.5
ashu@ubuntu-20:~/1941012250/6th Sem/A3$
```

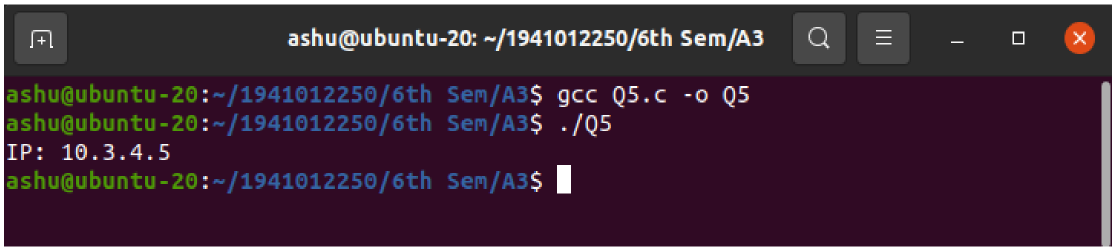
Q5. Assume that a line of code in your program is to read the IPV4 address in dotted decimal number and is stored in the network byte order as **`ca.sin_addr.s_addr=inet_addr(‘‘10.3.4.5’’)`**, where **`ca`** is the structure variable of the type **`struct sockaddr_in`**. You are required to rewritethe program to replace **`inet_addr(...)`** with **`inet_pton(...)`** to get the IP address and display that IP address using **`inet_ntop(..)`**.

Soln:-

```
#include <stdio.h>
#include <netinet/in.h>
#include <arpa/inet.h>
int main() {
    char *ip = "10.3.4.5";
    struct sockaddr_in ca;
    ca.sin_family = AF_INET;
    ca.sin_port = htons(34567);
    inet_pton(AF_INET, ip, &ca.sin_addr);
```

```
        char strptr[INET_ADDRSTRLEN];
        printf("IP: %s\n", inet_ntop(AF_INET, &ca.sin_addr, strptr,
    INET_ADDRSTRLEN));
        return 0;
    }
```

Output:-

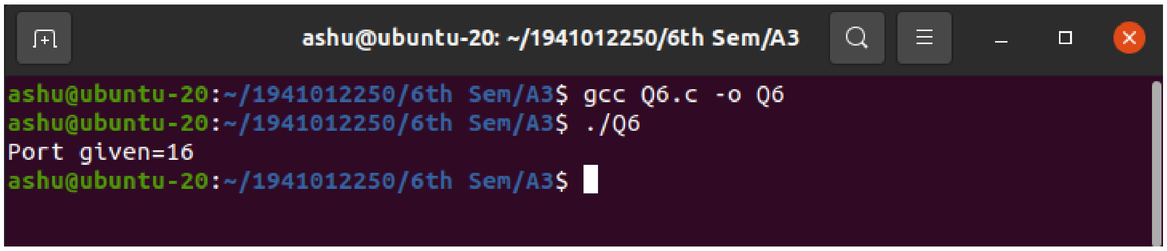


```
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ gcc Q5.c -o Q5
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q5
IP: 10.3.4.5
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ 
```

Q6. Find out the output of the given code snippet. Also state your answer for such output.

```
int main()
{
  struct  sockaddr_in  servaddr;
  servaddr.sin_family=AF_INET;
  servaddr.sin_port=16;
  printf("Port  given=%d\n",servaddr.sin_port);
  return 0;
}
```

Output:-

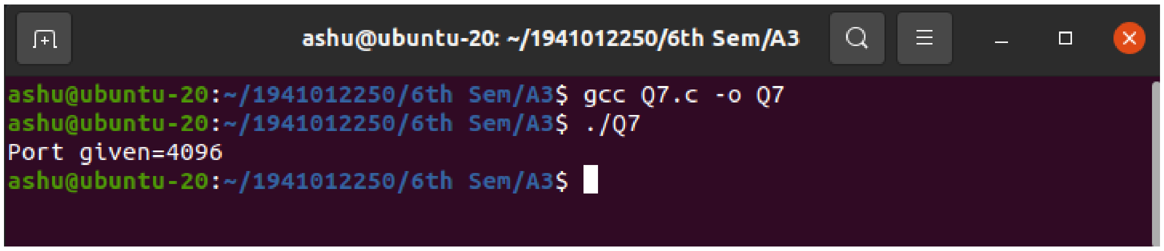


```
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ gcc Q6.c -o Q6
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q6
Port given=16
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ 
```

Q7. Find out the output of the given code snippet and justify the reason of getting such output (**Hint: lookinto Host byte order and Network byte order**).

```
int main()
{
  struct sockaddr_in servaddr;
  servaddr.sin_family=AF_INET;
  servaddr.sin_port=htons(16);
  printf("Port given=%d\n",servaddr.sin_port);
  return 0;
}
```

Output:-

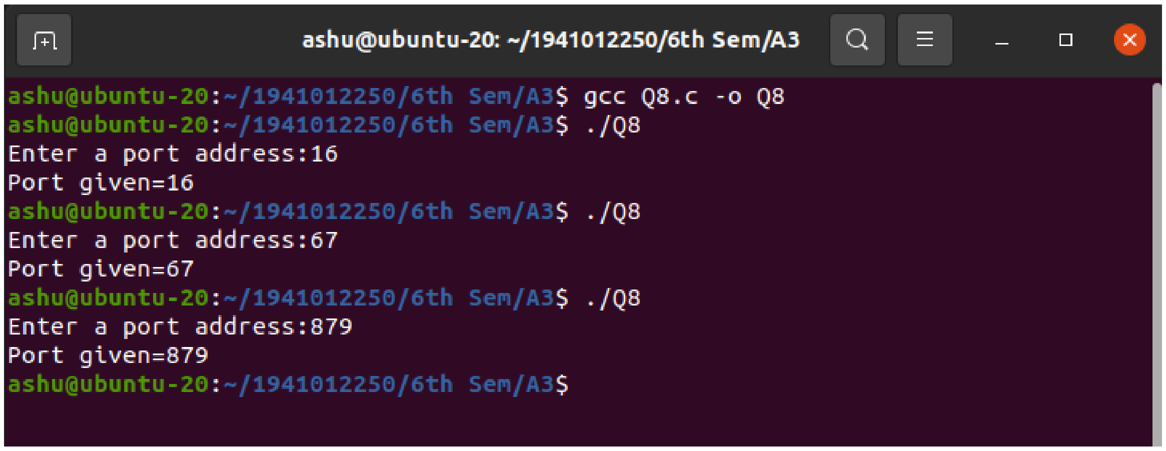


```
ashu@ubuntu-20: ~/1941012250/6th Sem/A3
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ gcc Q7.c -o Q7
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q7
Port given=4096
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ 
```

Q8. Fill out the missing parts of the following code snippet and Determine the output for the given portaddress as input: 16, 67, 879 respectively.

```
int main()
{
  __________ port; /* fill the desired data type */
  printf("Enter a port address:");
  scanf("%____",&port);
  struct sockaddr_in servaddr;
  servaddr.sin_family=AF_INET;
  servaddr.sin_port=htons(port);
  printf("Port given=%d\n",htons(servaddr.sin_port));
  return 0;
}
```

Output:-

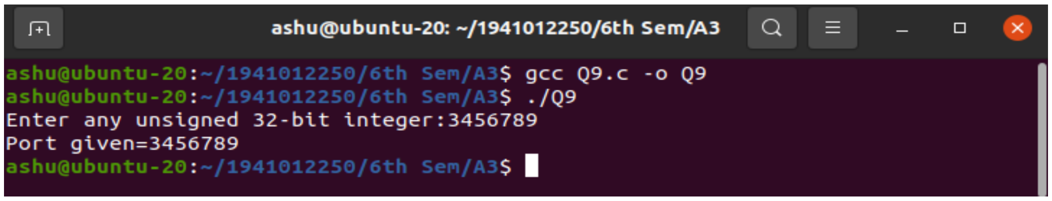


```
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ gcc Q8.c -o Q8
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q8
Enter a port address:16
Port given=16
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q8
Enter a port address:67
Port given=67
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q8
Enter a port address:879
Port given=879
ashu@ubuntu-20:~/1941012250/6th Sem/A3$
```

Q9. Find out the output of the code snippet and also state the reason for such output.

```
int main(){
  uint32_t ip;
  unsigned int x;
  printf("Enter any unsigned 32-bit integer:");
  scanf("%u",&x);
  ip=htonl(x);
  printf("Port given=%u\n",ntohl(ip));
  return 0;
}
```

Output:-



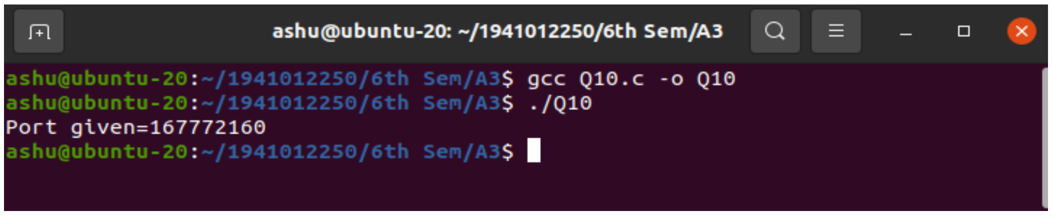
ashu@ubuntu-20: ~/1941012250/6th Sem/A3

```
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ gcc Q9.c -o Q9
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q9
Enter any unsigned 32-bit integer:3456789
Port given=3456789
ashu@ubuntu-20:~/1941012250/6th Sem/A3$
```

Q10. Find out the output of the code snippet and also trace the reason of getting such output.

```
int main()
{
  uint32_t ip;
  ip=htonl(10);
  printf("Port given=%u\n",ip);
  return 0;
}
```

Output:-



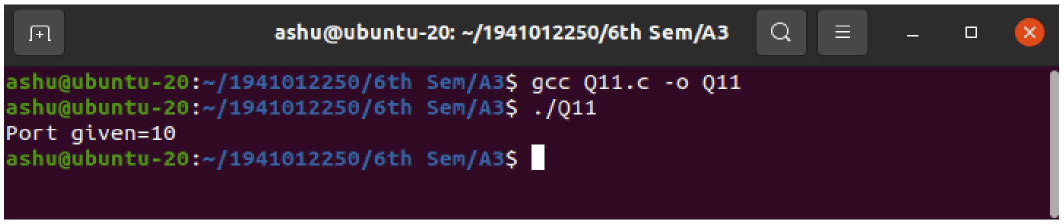
ashu@ubuntu-20: ~/1941012250/6th Sem/A3

```
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ gcc Q10.c -o Q10
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q10
Port given=167772160
ashu@ubuntu-20:~/1941012250/6th Sem/A3$
```

Q11. Find out the output of the code snippet and also trace the reason of getting such output.

```
int main()
{
  uint32_t ip;
  ip=10;
  printf("Port given=%u\n",ip);
  return 0;
}
```

Output:-

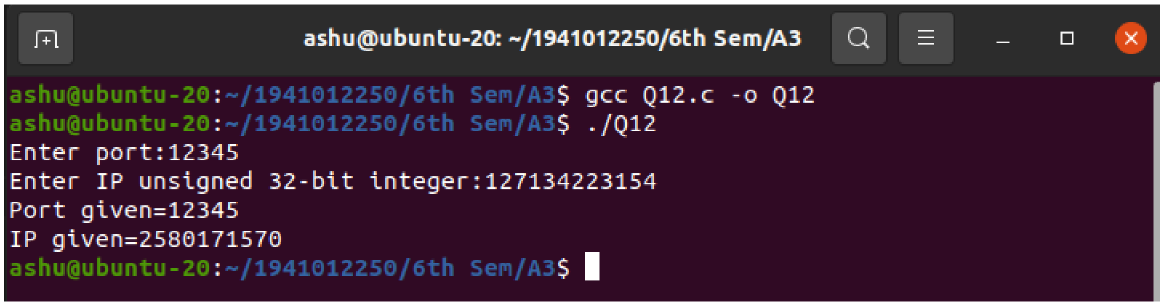


```
ashu@ubuntu-20: ~/1941012250/6th Sem/A3
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ gcc Q11.c -o Q11
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q11
Port given=10
ashu@ubuntu-20:~/1941012250/6th Sem/A3$
```

Q12. Justify the output of the given code snippet:

```
int main()
{
  in_addr_t ip;
  in_port_t  port;
  struct sockaddr_in sa;
  printf("Enter port:");
  scanf("%hu",&port);
  printf("Enter IP unsigned 32-bit integer:");
  scanf("%u",&ip);
  sa.sin_port=htons(port);
  sa.sin_addr.s_addr=htonl(ip);
  printf("Port given=%hu\n",ntohs(sa.sin_port));
  printf("IP given=%u\n",ntohl(sa.sin_addr.s_addr));
  return 0;
}
```

Output:-



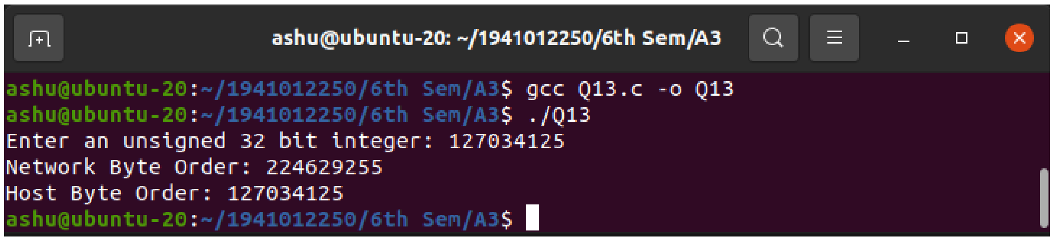
```
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ gcc Q12.c -o Q12
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q12
Enter port:12345
Enter IP unsigned 32-bit integer:127134223154
Port given=12345
IP given=2580171570
ashu@ubuntu-20:~/1941012250/6th Sem/A3$
```

Q13. Write a program to read an unsigned 32-bit integer *x* (*i.e datatype uint32 t*). Assign *x* to *y* in networkbyte order. Display the value of *y* in network byte order as well as in host byte order.

Soln:-

```
#include <stdio.h>
#include <netinet/in.h>
int main() {
    uint32_t x, y;
    printf("Enter an unsigned 32 bit integer: ");
    scanf("%u", &x);
    y = htonl(x);
    printf("Network Byte Order: %u\n", y);
    printf("Host Byte Order: %u\n", ntohl(y));
    return 0;
}
```

Output:-



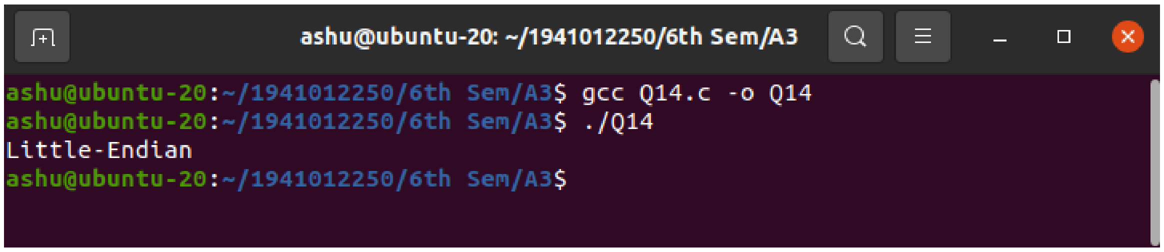
```
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ gcc Q13.c -o Q13
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q13
Enter an unsigned 32 bit integer: 127034125
Network Byte Order: 224629255
Host Byte Order: 127034125
ashu@ubuntu-20:~/1941012250/6th Sem/A3$
```

Q14. Develop a program to determine whether your working machine is in little-endian or in big-endianbyte order.

Soln:-

```
#include <stdio.h>
int main(int argc, char **argv) {
    union {
        short s;
        char c[sizeof(short)];
    } un;
    un.s = 0x0102;
    if (sizeof(short) == 2) {
        if (un.c[0] == 1 && un.c[1] == 2) {
            printf("Big-Endian\n");
        }
        else if (un.c[0] == 2 && un.c[1] == 1) {
            printf("Little-Endian\n");
        }
        else {
            printf("Unknown\n");
        }
    }
    else {
        printf("sizeof(short) = %lu\n", sizeof(short));
    }
    return 0;
}
```

Output:-

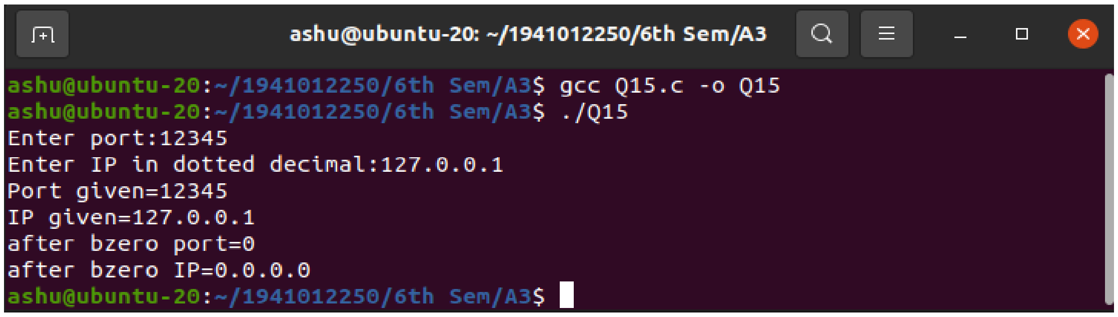


```
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ gcc Q14.c -o Q14
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q14
Little-Endian
ashu@ubuntu-20:~/1941012250/6th Sem/A3$
```

Q15. Find and Justify the output of the given code snippet:

```
int main()
{char ip[16];in_port_t port;
  struct sockaddr_in sa;
  bzero(&sa,sizeof(sa));
  printf("Enter port:");scanf("%hu",&port);
  printf("Enter IP in dotted decimal:");
  scanf("%s",ip);
  sa.sin_port=htons(port);
  sa.sin_addr.s_addr=inet_addr(ip); printf("Port
  given=%hu\n",ntohs(sa.sin_port));
  printf("IP given=%s\n",inet_ntoa(sa.sin_addr));
  bzero(&sa,sizeof(sa));
  printf("after bzero port=%hu\n",ntohs(sa.sin_port));
  printf("after bzero IP=%s\n",inet_ntoa(sa.sin_addr));
  bzero(&sa,sizeof(sa));


   return 0;
}
```

Output:-



```
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ gcc Q15.c -o Q15
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q15
Enter port:12345
Enter IP in dotted decimal:127.0.0.1
Port given=12345
IP given=127.0.0.1
after bzero port=0
after bzero IP=0.0.0.0
ashu@ubuntu-20:~/1941012250/6th Sem/A3$
```

Name: Ashutosh Palit Regd. Number: 1941012250

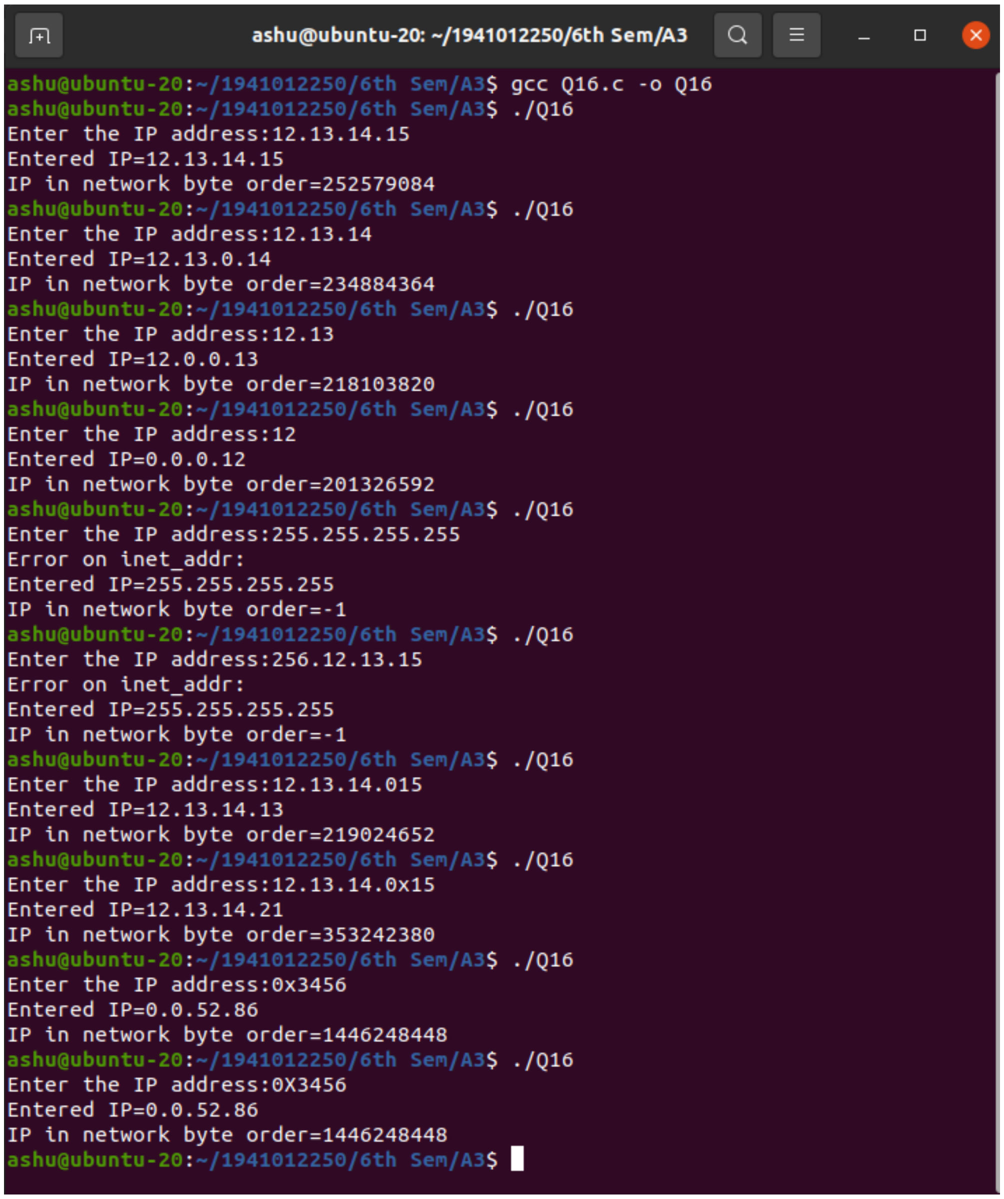
Q16. Consider the following code snippet to work with IP address conversion function **`inet_addr()`**:

```c
int main(){
  struct sockaddr_in serv;
  char ipaddr[16];
  printf("Enter the IP address:");
  scanf("%s",ipaddr);
  serv.sin_addr.s_addr=inet_addr(ipaddr);
  if(serv.sin_addr.s_addr==-1){
    printf("Error on inet_addr:\n");
  }
  printf("Entered IP=%s\n",inet_ntoa(serv.sin_addr));
  printf("IP in network byte order=%d\n",serv.sin_addr.s_addr);
  return 0;
}
```

Run the test cases and justify the program generated output

(a) Enter the IP address: 12.13.14.15
(b) Enter the IP address: 12.13.14
(c) Enter the IP address: 12.13
(d) Enter the IP address: 12
(e) Enter the IP address: 255.255.255.255
(f) Enter the IP address: 256.12.13.15
(g) Enter the IP address: 12.13.14.015
(h) Enter the IP address: 12.13.14.0x15
(i) Enter the IP address: 0x3456
(j) Enter the IP address: 0X3456

Output:-



```
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ gcc Q16.c -o Q16
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q16
Enter the IP address:12.13.14.15
Entered IP=12.13.14.15
IP in network byte order=252579084
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q16
Enter the IP address:12.13.14
Entered IP=12.13.0.14
IP in network byte order=234884364
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q16
Enter the IP address:12.13
Entered IP=12.0.0.13
IP in network byte order=218103820
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q16
Enter the IP address:12
Entered IP=0.0.0.12
IP in network byte order=201326592
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q16
Enter the IP address:255.255.255.255
Error on inet_addr:
Entered IP=255.255.255.255
IP in network byte order=-1
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q16
Enter the IP address:256.12.13.15
Error on inet_addr:
Entered IP=255.255.255.255
IP in network byte order=-1
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q16
Enter the IP address:12.13.14.015
Entered IP=12.13.14.13
IP in network byte order=219024652
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q16
Enter the IP address:12.13.14.0x15
Entered IP=12.13.14.21
IP in network byte order=353242380
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q16
Enter the IP address:0x3456
Entered IP=0.0.52.86
IP in network byte order=1446248448
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q16
Enter the IP address:0X3456
Entered IP=0.0.52.86
IP in network byte order=1446248448
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ 
```

Q17. Rewrite the question-14 code to replace **`inet_addr()`** function with **`inet_aton()`** and verify the above test cases in your modified program.

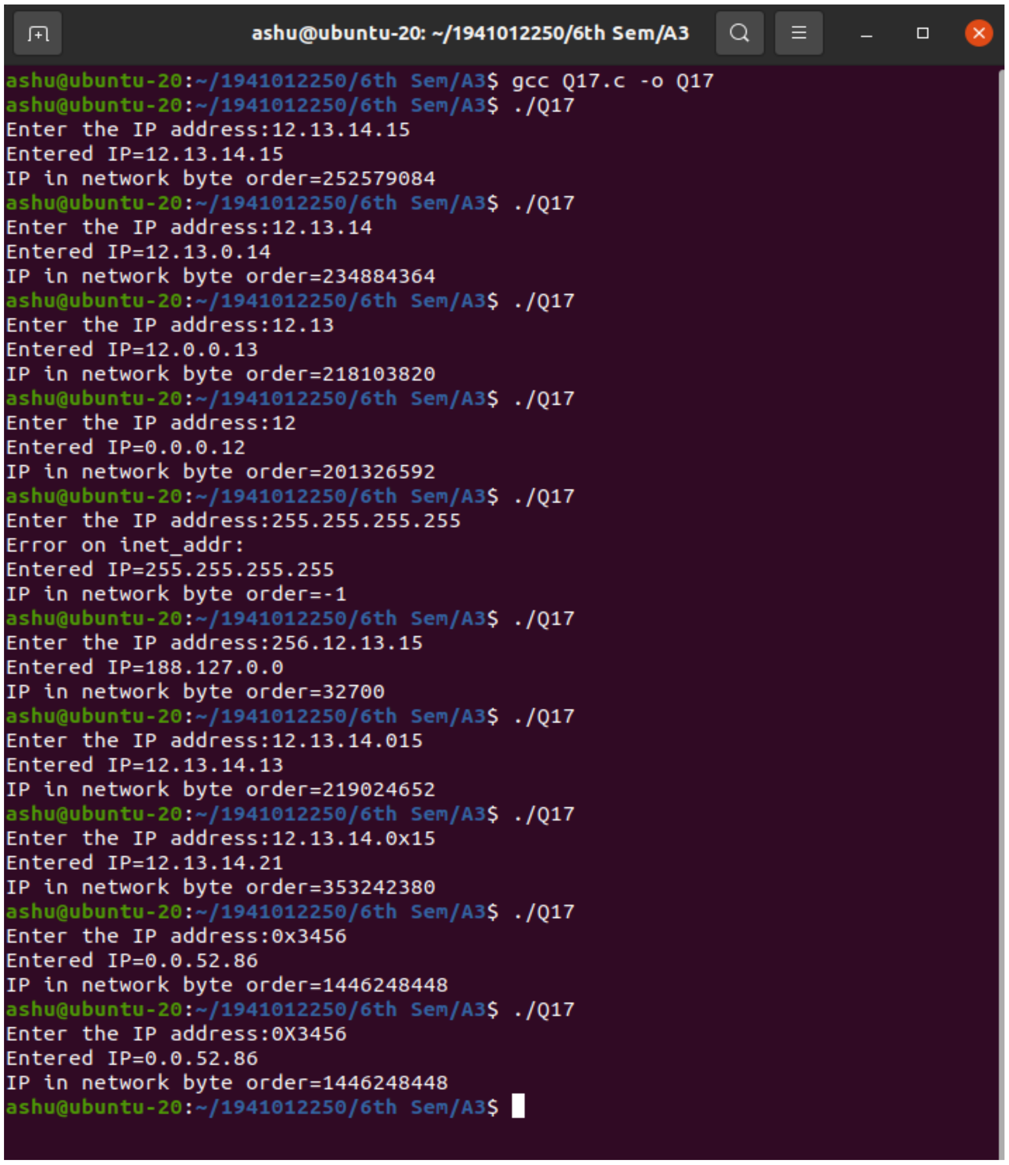
Soln:-

```
#include <stdio.h>
#include <netinet/in.h>
#include <arpa/inet.h>
int main(){
    struct sockaddr_in serv;
    char ipaddr[16];
    printf("Enter the IP address:");
    scanf("%s",ipaddr);
    inet_aton(ipaddr, &serv.sin_addr);
    if(serv.sin_addr.s_addr==-1){
        printf("Error on inet_addr:\n");
    }
    printf("Entered IP=%s\n",inet_ntoa(serv.sin_addr));
    printf("IP in network byte order=%d\n",serv.sin_addr.s_addr);
    return 0;
}
```

Run the test cases and justify the program generated output

(a) Enter the IP address: 12.13.14.15
(b) Enter the IP address: 12.13.14
(c) Enter the IP address: 12.13
(d) Enter the IP address: 12
(e) Enter the IP address: 255.255.255.255
(f) Enter the IP address: 256.12.13.15
(g) Enter the IP address: 12.13.14.015
(h) Enter the IP address: 12.13.14.0x15
(i) Enter the IP address: 0x3456
(j) Enter the IP address: 0X3456

Output:-



```
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ gcc Q17.c -o Q17
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q17
Enter the IP address:12.13.14.15
Entered IP=12.13.14.15
IP in network byte order=252579084
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q17
Enter the IP address:12.13.14
Entered IP=12.13.0.14
IP in network byte order=234884364
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q17
Enter the IP address:12.13
Entered IP=12.0.0.13
IP in network byte order=218103820
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q17
Enter the IP address:12
Entered IP=0.0.0.12
IP in network byte order=201326592
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q17
Enter the IP address:255.255.255.255
Error on inet_addr:
Entered IP=255.255.255.255
IP in network byte order=-1
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q17
Enter the IP address:256.12.13.15
Entered IP=188.127.0.0
IP in network byte order=32700
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q17
Enter the IP address:12.13.14.015
Entered IP=12.13.14.13
IP in network byte order=219024652
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q17
Enter the IP address:12.13.14.0x15
Entered IP=12.13.14.21
IP in network byte order=353242380
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q17
Enter the IP address:0x3456
Entered IP=0.0.52.86
IP in network byte order=1446248448
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q17
Enter the IP address:0X3456
Entered IP=0.0.52.86
IP in network byte order=1446248448
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ 
```

Q18. Rewrite the question-14 code to replace **`inet addr()`** function with **`inet pton()`** and verify the above test cases in your modified program.

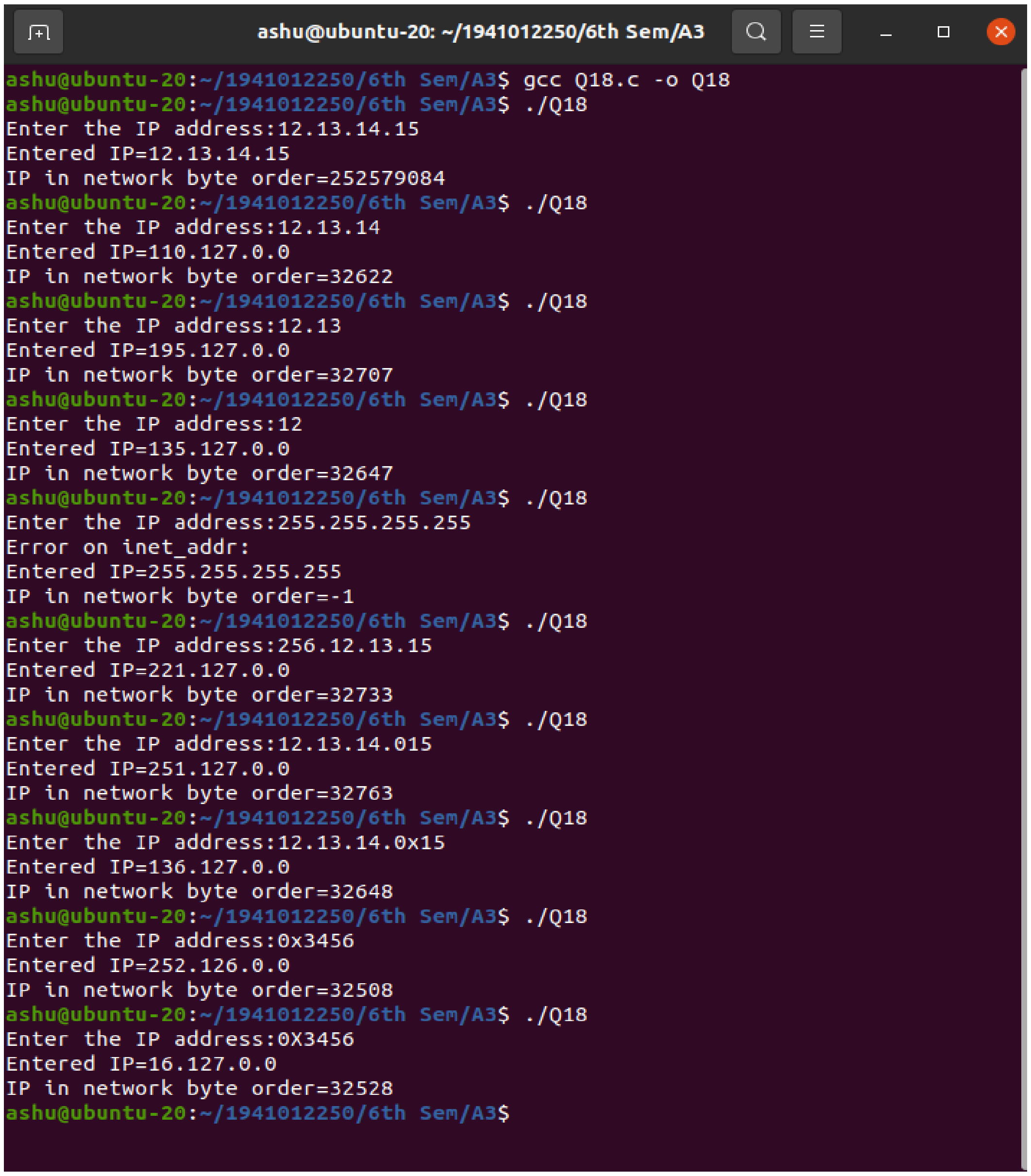
Soln:-

```
#include <stdio.h>
#include <netinet/in.h>
#include <arpa/inet.h>
int main(){
    struct sockaddr_in serv;
    char ipaddr[16];
    printf("Enter the IP address:");
    scanf("%s",ipaddr);
    inet_pton(AF_INET, ipaddr, &serv.sin_addr);
    if(serv.sin_addr.s_addr==-1){
        printf("Error on inet_addr:\n");
    }
    printf("Entered IP=%s\n",inet_ntoa(serv.sin_addr));
    printf("IP in network byte order=%d\n",serv.sin_addr.s_addr);
    return 0;
}
```

Run the test cases and justify the program generated output

(a) Enter the IP address: 12.13.14.15
(b) Enter the IP address: 12.13.14
(c) Enter the IP address: 12.13
(d) Enter the IP address: 12
(e) Enter the IP address: 255.255.255.255
(f) Enter the IP address: 256.12.13.15
(g) Enter the IP address: 12.13.14.015
(h) Enter the IP address: 12.13.14.0x15
(i) Enter the IP address: 0x3456
(j) Enter the IP address: 0X3456

Output:-



```
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ gcc Q18.c -o Q18
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q18
Enter the IP address:12.13.14.15
Entered IP=12.13.14.15
IP in network byte order=252579084
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q18
Enter the IP address:12.13.14
Entered IP=110.127.0.0
IP in network byte order=32622
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q18
Enter the IP address:12.13
Entered IP=195.127.0.0
IP in network byte order=32707
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q18
Enter the IP address:12
Entered IP=135.127.0.0
IP in network byte order=32647
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q18
Enter the IP address:255.255.255.255
Error on inet_addr:
Entered IP=255.255.255.255
IP in network byte order=-1
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q18
Enter the IP address:256.12.13.15
Entered IP=221.127.0.0
IP in network byte order=32733
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q18
Enter the IP address:12.13.14.015
Entered IP=251.127.0.0
IP in network byte order=32763
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q18
Enter the IP address:12.13.14.0x15
Entered IP=136.127.0.0
IP in network byte order=32648
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q18
Enter the IP address:0x3456
Entered IP=252.126.0.0
IP in network byte order=32508
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q18
Enter the IP address:0X3456
Entered IP=16.127.0.0
IP in network byte order=32528
ashu@ubuntu-20:~/1941012250/6th Sem/A3$
```

Q19. Rewrite the question-14 code to replace **`inet addr()`** function with **`inet pton()`** and verify the above test cases in your modified program.

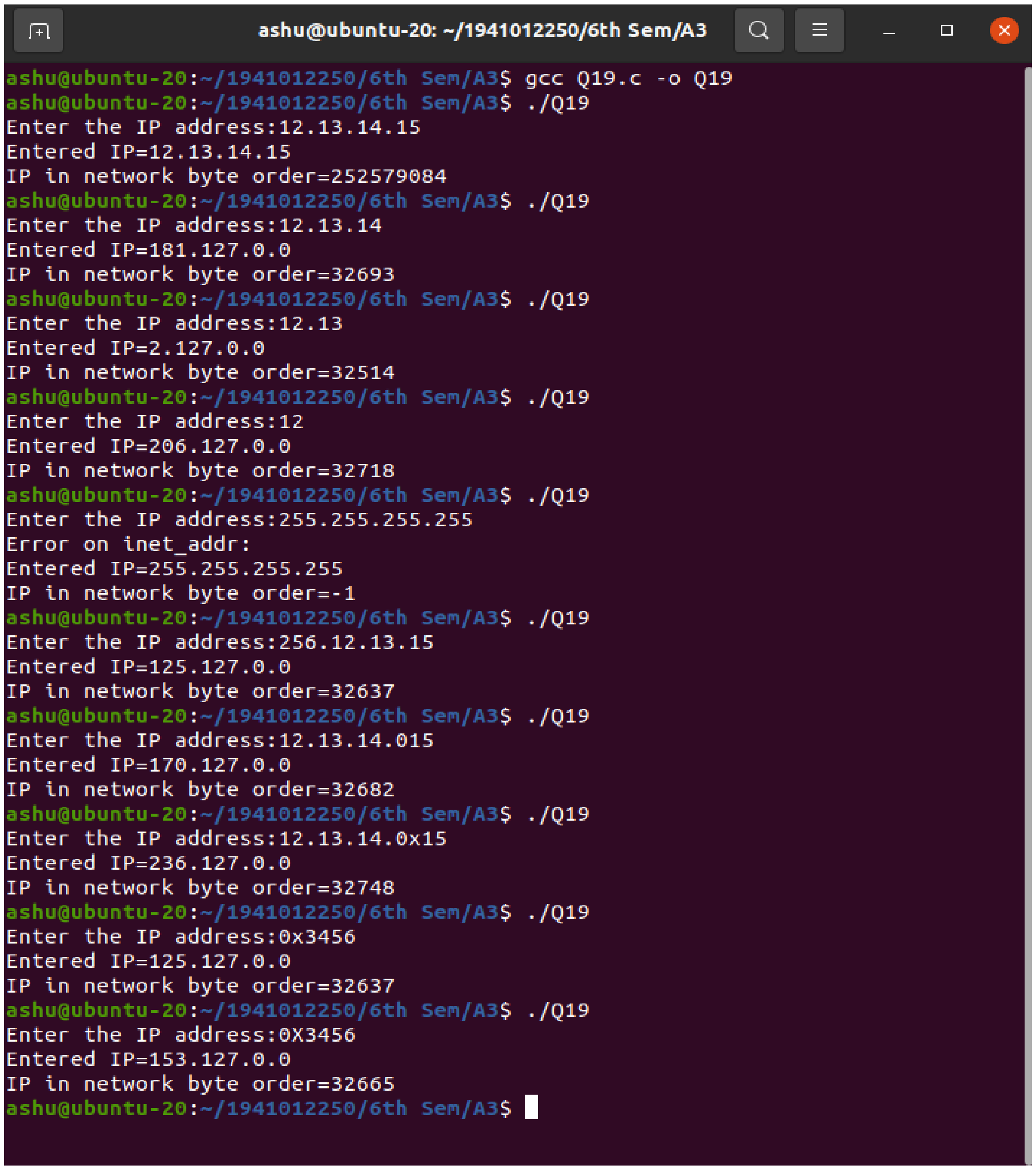
<u>Soln</u>:-

```
#include <stdio.h>
#include <netinet/in.h>
#include <arpa/inet.h>
int main(){
    struct sockaddr_in serv;
    char ipaddr[16];
    printf("Enter the IP address:");
    scanf("%s",ipaddr);
    inet_pton(AF_INET, ipaddr, &serv.sin_addr);
    if(serv.sin_addr.s_addr==-1){
        printf("Error on inet_addr:\n");
    }
    printf("Entered IP=%s\n",inet_ntoa(serv.sin_addr));
    printf("IP in network byte order=%d\n",serv.sin_addr.s_addr);
    return 0;
}
```

Run the test cases and justify the program generated output

(a) Enter the IP address: 12.13.14.15
(b) Enter the IP address: 12.13.14
(c) Enter the IP address: 12.13
(d) Enter the IP address: 12
(e) Enter the IP address: 255.255.255.255
(f) Enter the IP address: 256.12.13.15
(g) Enter the IP address: 12.13.14.015
(h) Enter the IP address: 12.13.14.0x15
(i) Enter the IP address: 0x3456
(j) Enter the IP address: 0X3456

Output:-


ashu@ubuntu-20: ~/1941012250/6th Sem/A3

```
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ gcc Q19.c -o Q19
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q19
Enter the IP address:12.13.14.15
Entered IP=12.13.14.15
IP in network byte order=252579084
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q19
Enter the IP address:12.13.14
Entered IP=181.127.0.0
IP in network byte order=32693
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q19
Enter the IP address:12.13
Entered IP=2.127.0.0
IP in network byte order=32514
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q19
Enter the IP address:12
Entered IP=206.127.0.0
IP in network byte order=32718
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q19
Enter the IP address:255.255.255.255
Error on inet_addr:
Entered IP=255.255.255.255
IP in network byte order=-1
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q19
Enter the IP address:256.12.13.15
Entered IP=125.127.0.0
IP in network byte order=32637
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q19
Enter the IP address:12.13.14.015
Entered IP=170.127.0.0
IP in network byte order=32682
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q19
Enter the IP address:12.13.14.0x15
Entered IP=236.127.0.0
IP in network byte order=32748
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q19
Enter the IP address:0x3456
Entered IP=125.127.0.0
IP in network byte order=32637
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q19
Enter the IP address:0X3456
Entered IP=153.127.0.0
IP in network byte order=32665
ashu@ubuntu-20:~/1941012250/6th Sem/A3$
```

Q20. Rewrite the question-14 code to replace **`inet addr()`** function with **`inet pton()`** and **`inet ntoa()`** to **`inet ntop()`**. Also, verify the above test cases in your modified program.

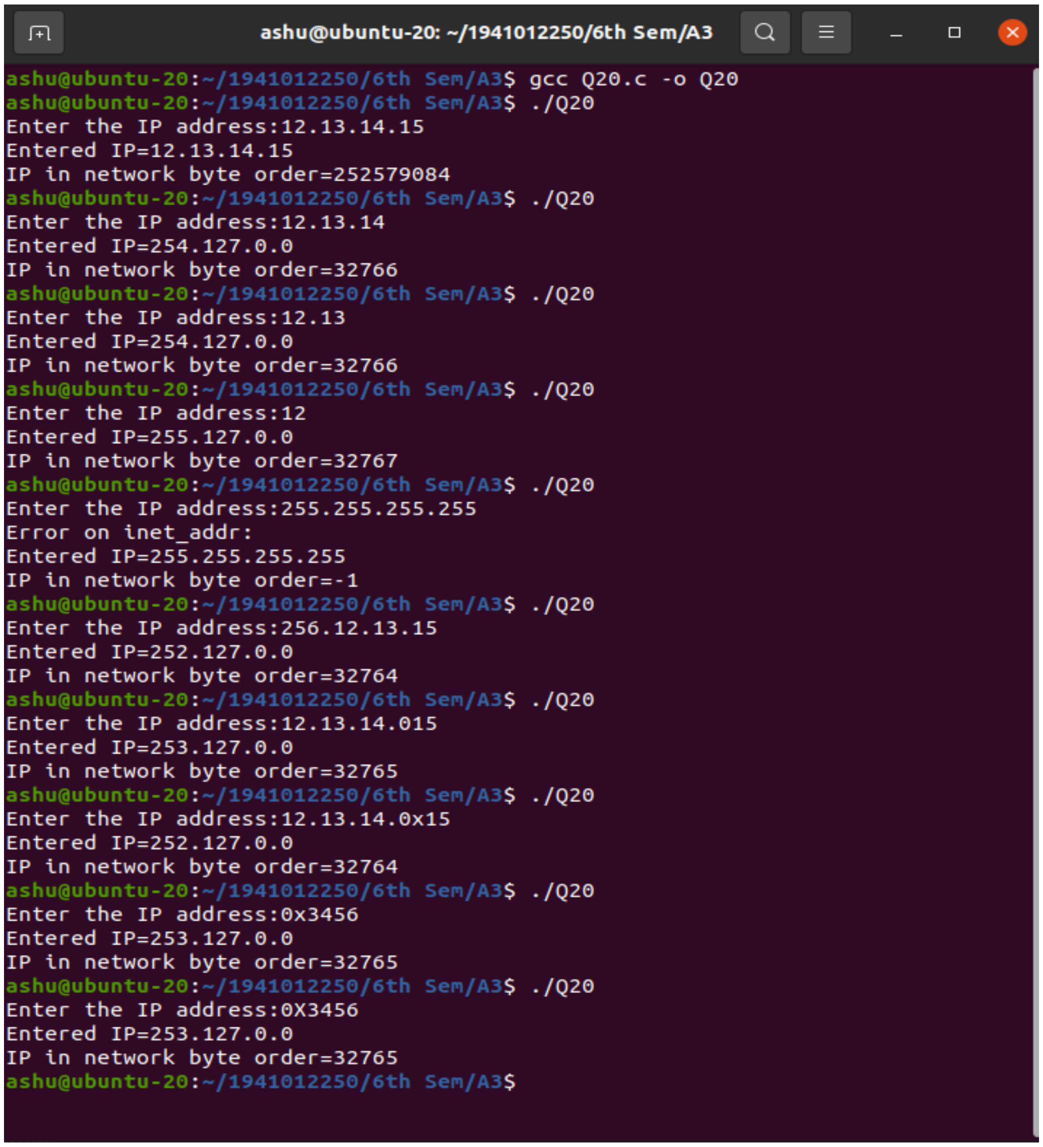
Soln:-

```
#include <stdio.h>
#include <netinet/in.h>
#include <arpa/inet.h>
int main(){
    struct sockaddr_in serv;
    char ipaddr[16];
    printf("Enter the IP address:");
    scanf("%s",ipaddr);
    inet_pton(AF_INET, ipaddr, &serv.sin_addr);
    if(serv.sin_addr.s_addr==-1){
        printf("Error on inet_addr:\n");
    }
    char strptr[INET_ADDRSTRLEN];
    printf("Entered IP=%s\n",inet_ntop(AF_INET, &serv.sin_addr, strptr,
INET_ADDRSTRLEN));
    printf("IP in network byte order=%d\n",serv.sin_addr.s_addr);
    return 0;
}
```

Run the test cases and justify the program generated output

(a) Enter the IP address: 12.13.14.15
(b) Enter the IP address: 12.13.14
(c) Enter the IP address: 12.13
(d) Enter the IP address: 12
(e) Enter the IP address: 255.255.255.255
(f) Enter the IP address: 256.12.13.15
(g) Enter the IP address: 12.13.14.015
(h) Enter the IP address: 12.13.14.0x15
(i) Enter the IP address: 0x3456
(j) Enter the IP address: 0X3456

Output:-



ashu@ubuntu-20: ~/1941012250/6th Sem/A3

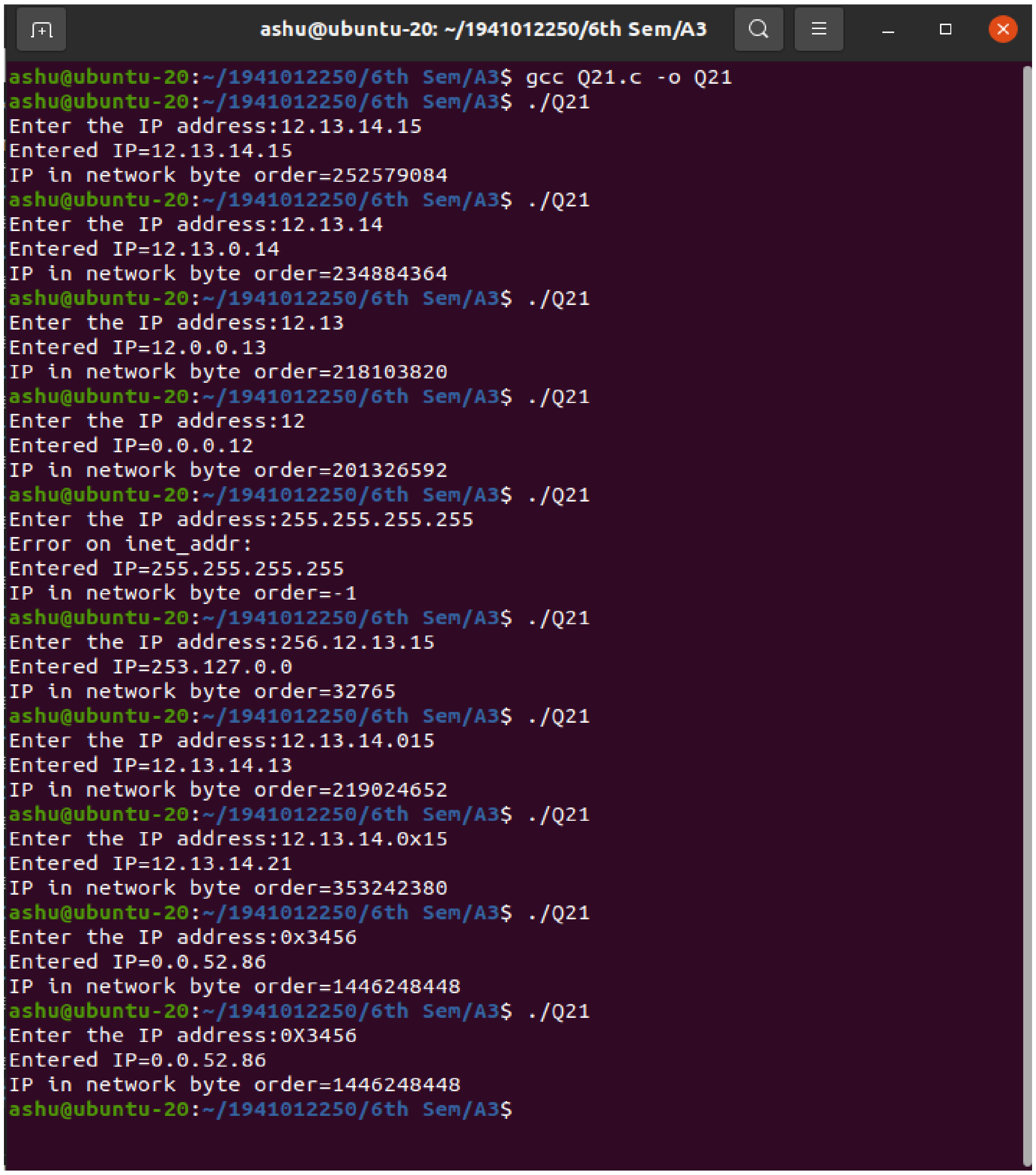
```
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ gcc Q20.c -o Q20
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q20
Enter the IP address:12.13.14.15
Entered IP=12.13.14.15
IP in network byte order=252579084
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q20
Enter the IP address:12.13.14
Entered IP=254.127.0.0
IP in network byte order=32766
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q20
Enter the IP address:12.13
Entered IP=254.127.0.0
IP in network byte order=32766
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q20
Enter the IP address:12
Entered IP=255.127.0.0
IP in network byte order=32767
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q20
Enter the IP address:255.255.255.255
Error on inet_addr:
Entered IP=255.255.255.255
IP in network byte order=-1
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q20
Enter the IP address:256.12.13.15
Entered IP=252.127.0.0
IP in network byte order=32764
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q20
Enter the IP address:12.13.14.015
Entered IP=253.127.0.0
IP in network byte order=32765
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q20
Enter the IP address:12.13.14.0x15
Entered IP=252.127.0.0
IP in network byte order=32764
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q20
Enter the IP address:0x3456
Entered IP=253.127.0.0
IP in network byte order=32765
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q20
Enter the IP address:0X3456
Entered IP=253.127.0.0
IP in network byte order=32765
ashu@ubuntu-20:~/1941012250/6th Sem/A3$
```

Q21. You know that **inet pton** is much stricter with IPV4 address and requires exactly four numbers sep- arated by three decimal points, with each number between 0 to 255. To make only specific to IPV4 and not stricter address format, write a program to develop an alternate version of **inet pton** func- tion named as **inet pton as aton** to support only IPV4 family and *will work like in*et **aton**. The function prototype is given as **int inet pton as aton(int family, const char *strptr, void *addrptr)**;. Your designed function must return 1 on success, 0 on failure. Also pass all the test cases of question-14.

<u>**Soln**</u>:-

```
#include <stdio.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <errno.h>
#include <string.h>
int inet_pton_as_aton(int, const char *, void *);
int main() {
    struct sockaddr_in serv;
    char ipaddr[16];
    printf("Enter the IP address:");
    scanf("%s",ipaddr);
    inet_pton_as_aton(AF_INET, ipaddr, &serv.sin_addr);
    if(serv.sin_addr.s_addr==-1){
        printf("Error on inet_addr:\n");
    }
    char strptr[INET_ADDRSTRLEN];
    printf("Entered IP=%s\n",inet_ntop(AF_INET, &serv.sin_addr, strptr,
INET_ADDRSTRLEN));
    printf("IP in network byte order=%d\n",serv.sin_addr.s_addr);
    return 0;
}

int inet_pton_as_aton(int family, const char *strptr, void *addrptr) {
    if (family == AF_INET) {
        struct in_addr ipv4;
        if (inet_aton(strptr, &ipv4)) {
            memcpy(addrptr, &ipv4, sizeof(struct in_addr));
            return 1;
        }
        return 0;
    }
    errno = EAFNOSUPPORT;
    return -1;
}
```

Output:-



ashu@ubuntu-20: ~/1941012250/6th Sem/A3

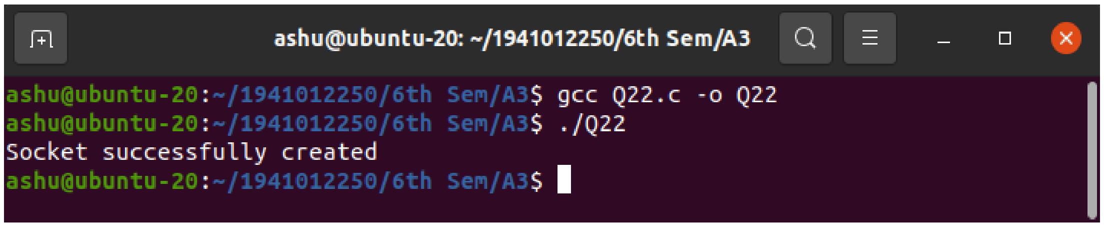
```
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ gcc Q21.c -o Q21
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q21
Enter the IP address:12.13.14.15
Entered IP=12.13.14.15
IP in network byte order=252579084
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q21
Enter the IP address:12.13.14
Entered IP=12.13.0.14
IP in network byte order=234884364
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q21
Enter the IP address:12.13
Entered IP=12.0.0.13
IP in network byte order=218103820
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q21
Enter the IP address:12
Entered IP=0.0.0.12
IP in network byte order=201326592
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q21
Enter the IP address:255.255.255.255
Error on inet_addr:
Entered IP=255.255.255.255
IP in network byte order=-1
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q21
Enter the IP address:256.12.13.15
Entered IP=253.127.0.0
IP in network byte order=32765
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q21
Enter the IP address:12.13.14.015
Entered IP=12.13.14.13
IP in network byte order=219024652
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q21
Enter the IP address:12.13.14.0x15
Entered IP=12.13.14.21
IP in network byte order=353242380
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q21
Enter the IP address:0x3456
Entered IP=0.0.52.86
IP in network byte order=1446248448
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q21
Enter the IP address:0X3456
Entered IP=0.0.52.86
IP in network byte order=1446248448
ashu@ubuntu-20:~/1941012250/6th Sem/A3$
```

Q22. Write a program to create a socket (i.e. end-point of a connection) and display the whether end-pointis successfully created or not.

Soln:-

```
#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
int main() {
    int sockfd = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
    sockfd ? printf("Socket successfully created\n") : printf("Socket
creation failed\n");
    return 0;
}
```

Output:-



```
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ gcc Q22.c -o Q22
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ ./Q22
Socket successfully created
ashu@ubuntu-20:~/1941012250/6th Sem/A3$ 
```