

```

//Classful
#include <stdio.h>
int main() {
    int o1, o2, o3, o4;
    printf("Enter an IP address (format: x.x.x.x): ");
    scanf("%d.%d.%d.%d", &o1, &o2, &o3, &o4);
    if (o1 >= 0 && o1 <= 127) {
        printf("Class A\n");
        printf("Network ID: %d.0.0.0\n", o1);
        printf("Host ID: 0.%d.%d.%d\n", o2, o3, o4);
    }
    else if (o1 >= 128 && o1 <= 191) {
        printf("Class B\n");
        printf("Network ID: %d.%d.0.0\n", o1, o2);
        printf("Host ID: 0.0.%d.%d\n", o3, o4);
    }
    else if (o1 >= 192 && o1 <= 223) {
        printf("Class C\n");
        printf("Network ID: %d.%d.%d.0\n", o1, o2, o3);
        printf("Host ID: 0.0.0.%d\n", o4);
    }
    else if (o1 >= 224 && o1 <= 239) {
        printf("Class D (Multicast)\n");
    }
    else if (o1 >= 240 && o1 <= 255) {
        printf("Class E (Experimental)\n");
    }
    else {
        printf("Invalid IP address\n");
    }
    return 0;
}

```

```

Enter an IP address (format: x.x.x.x): 192.168.1.5
Class C
Network ID: 192.168.1.0
Host ID: 0.0.0.5

```

=== Code Execution Successful ===

```

//Classless
#include <stdio.h>
int main() {
    int a,b,c,d,p;
    unsigned int ip, mask, net, bc, hosts;
    printf("Enter IP/prefix: ");
    scanf("%d.%d.%d.%d/%d", &a,&b,&c,&d,&p);
    ip = (a<<24) | (b<<16) | (c<<8) | d;
    mask = (p==0)?0:(0xFFFFFFFF << (32-p));
    net = ip & mask;
    bc = net | (~mask);
    hosts = (1u << (32-p)) - 2;
    printf("\nSubnet Mask: %u.%u.%u.%u\n",
        mask>>24, (mask>>16)&255, (mask>>8)&255, mask&255);
    printf("Network Address: %u.%u.%u.%u\n",
        net>>24, (net>>16)&255, (net>>8)&255, net&255);
    printf("Broadcast Address: %u.%u.%u.%u\n",
        bc>>24, (bc>>16)&255, (bc>>8)&255, bc&255);
    printf("Number of Hosts: %u\n", hosts);
}

```

Enter IP/prefix: 192.168.1.10/27

Subnet Mask: 255.255.255.224

Network Address: 192.168.1.0

Broadcast Address: 192.168.1.31

Number of Hosts: 30

=== Code Execution Successful ===

```

//Dijkstras
#include <stdio.h>
#define INF 9999
int main() {
    int n,i,j,s,c[20][20],d[20],v[20]={0};
    printf("Enter number of nodes: ");
    scanf("%d",&n);
    printf("Enter adjacency matrix (use 9999 for no path):\n");
    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
            scanf("%d",&c[i][j]);
    printf("Enter source node: ");
    scanf("%d",&s);
    for(i=0;i<n;i++) d[i]=c[s][i];
    d[s]=0;
    v[s]=1;
    for(i=1;i<n;i++){
        int u=-1,min=INF;
        for(j=0;j<n;j++)
            if(!v[j] && d[j]<min) min=d[j],u=j;
        v[u]=1;
        for(j=0;j<n;j++)
            if(!v[j] && d[u]+c[u][j] < d[j])
                d[j]=d[u]+c[u][j];
    }
    printf("\nShortest distances from node %d:\n", s);
    for(i=0;i<n;i++)
        printf("To node %d = %d\n", i, d[i]);
}

```

```

Enter number of nodes: 4
Enter adjacency matrix (use 9999 for no path):
0 5 9999 10
5 0 3 9999
9999 3 0 1
10 9999 1 0
Enter source node: 0

```

```

Shortest distances from node 0:
To node 0 = 0
To node 1 = 5
To node 2 = 8
To node 3 = 9

```

=== Code Execution Successful ===

// Subnets

```
#include <stdio.h>
int main() {
    char ip[20];
    int n, block, i;
    printf("Enter network IP (ex: 192.168.1.0): ");
    scanf("%s", ip);
    printf("Enter number of subnets: ");
    scanf("%d", &n);
    block = 256 / n; // size of each subnet
    printf("\n--- Subnets ---\n");
    int a, b, c, d;
    sscanf(ip, "%d.%d.%d.%d", &a, &b, &c, &d);
    for(i=0; i<n; i++) {
        printf("Subnet %d : %d.%d.%d.%d/24\n",
            i+1, a, b, c, i * block);
    }
    return 0;
}
```

Enter network IP (ex: 192.168.1.0): 192.168.10.0
Enter number of subnets: 4

--- Subnets ---

Subnet 1 : 192.168.10.0/24
Subnet 2 : 192.168.10.64/24
Subnet 3 : 192.168.10.128/24
Subnet 4 : 192.168.10.192/24

=== Code Execution Successful ===

```
// DVR
#include <stdio.h>
#define INF 9999
int main() {
    int n,i,j,k,c[20][20];
    scanf("%d",&n);
    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
            scanf("%d",&c[i][j]);
    for(k=0;k<n;k++)
        for(i=0;i<n;i++)
            for(j=0;j<n;j++)
                if(c[i][j] > c[i][k] + c[k][j])
                    c[i][j] = c[i][k] + c[k][j];
    for(i=0;i<n;i++){
        printf("\nRouter %d:\n", i);
        for(j=0;j<n;j++)
            printf("To %d : Cost = %d\n", j, c[i][j]);
    }
}
```

```
3
0 2 7
2 0 1
7 1 0
```

```
Router 0:
To 0 : Cost = 0
To 1 : Cost = 2
To 2 : Cost = 3
```

```
Router 1:
To 0 : Cost = 2
To 1 : Cost = 0
To 2 : Cost = 1
```

```
Router 2:
To 0 : Cost = 3
To 1 : Cost = 1
To 2 : Cost = 0
```

=== Code Execution Successful ===

```
//LS
#include <stdio.h>
#define INF 9999
int main() {
    int n,i,j,k,c[20][20];
    scanf("%d",&n);
    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
            scanf("%d",&c[i][j]);
    for(k=0;k<n;k++)
        for(i=0;i<n;i++)
            for(j=0;j<n;j++)
                if(c[i][j] > c[i][k] + c[k][j])
                    c[i][j] = c[i][k] + c[k][j];
    for(i=0;i<n;i++){
        printf("\nRouter %d:\n", i);
        for(j=0;j<n;j++)
            printf("To %d : Cost = %d\n", j, c[i][j]);
    }
}
4
0 3 9999 7
3 0 2 9999
9999 2 0 1
7 9999 1 0
```

Router 0:
 To 0 : Cost = 0
 To 1 : Cost = 3
 To 2 : Cost = 5
 To 3 : Cost = 6

Router 1:
 To 0 : Cost = 3
 To 1 : Cost = 0
 To 2 : Cost = 2
 To 3 : Cost = 3

Router 2:
 To 0 : Cost = 5
 To 1 : Cost = 2
 To 2 : Cost = 0
 To 3 : Cost = 1

Router 3:
 To 0 : Cost = 6
 To 1 : Cost = 3
 To 2 : Cost = 1
 To 3 : Cost = 0

=== Code Execution Successful ===

// Leaky Bucket

```
#include <stdio.h>
int main() {
    int cap, rate, n, pkt, bucket = 0, i;
    printf("Enter bucket capacity: ");
    scanf("%d", &cap);
    printf("Enter output rate: ");
    scanf("%d", &rate);
    printf("Enter number of packets: ");
    scanf("%d", &n);
    printf("Enter packet sizes:\n");
    for(i = 1; i <= n; i++) {
        scanf("%d", &pkt);
        printf("\nPacket %d of size %d arrived.\n", i, pkt);
        if(bucket + pkt > cap)
            printf("Bucket overflow! Packet dropped.\n");
        else {
            bucket += pkt;
            printf("Bucket size now = %d\n", bucket);
            printf("Sent out %d\n", bucket < rate ? bucket : rate);
            bucket = bucket < rate ? 0 : bucket - rate;
            printf("Bucket left = %d\n", bucket);
        }
    }
}
```

Enter bucket capacity: 10
Enter output rate: 4
Enter number of packets: 5
Enter packet sizes:
3 7 5 2 6

Packet 1 of size 3 arrived.
Bucket size now = 3
Sent out 3
Bucket left = 0

Packet 2 of size 7 arrived.
Bucket size now = 7
Sent out 4
Bucket left = 3

Packet 3 of size 5 arrived.
Bucket size now = 8
Sent out 4
Bucket left = 4

Packet 4 of size 2 arrived.
Bucket size now = 6
Sent out 4
Bucket left = 2

Packet 5 of size 6 arrived.
Bucket size now = 8
Sent out 4
Bucket left = 4

=== Code Execution Successful ===

// **ED**

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

```
int main() {  
    char t[100];  
    int k, c, i;  
    printf("Enter text: ");  
    scanf("%[^\n]", t);  
    printf("Enter key: ");  
    scanf("%d", &k);  
    printf("1. Encrypt 2. Decrypt: ");  
    scanf("%d", &c);  
    for(i = 0; t[i] != '\0'; i++) {  
        if(c == 1) t[i] = t[i] + k; // encryption  
        else      t[i] = t[i] - k; // decryption  
    }  
    printf("Output: %s", t);  
}
```

Enter text: khor
Enter key: 3
1. Encrypt 2. Decrypt: 2
Output: hello

=== Code Execution Successful ===