

```

import math
def ip(attr, a, n):
    dataset = [[0 for i in range(n)] for i in range(a+1)]

    for i in range(0,a):
        attr.append(input("Enter attribute {} : ".format(i+1)))
    attr.append(input("Enter target attribute : "))

    for i in range(0,a+1):
        curr_att = attr[i]
        for j in range(0,n):
            dataset[i][j] = input("Enter the value {} of {} : ".format(j,curr_att))
    return dataset

```

```

def tab(yes, no, noi, val, attr, dataset, sel, n):

```

```

    for i in range(0,n):
        x = dataset[sel][i]
        y = dataset[-1][i]
        if x not in val:
            val.append(x)
            yes.append(0)
            no.append(0)
            noi.append(0)
        pos = val.index(x)
        noi[pos] += 1
        if y == 'Y':
            yes[pos] += 1
        elif y == 'N':
            no[pos] += 1

```

```

def compute(attr, dataset, sel, n):

```

```

    yes = []
    no = []
    noi = []
    val = []
    gini = 0
    tab(yes, no, noi, val, attr, dataset, sel, n)
    for i in range(0,len(val)):
        y = yes[i]/noi[i]
        z = no[i]/noi[i]
        x = 1 - math.pow(y,2) - math.pow(z,2)
        gini += noi[i]/n*x
    return(gini)

```

```

def next(attr, gini_l, dataset, a, n):

```

```

    for i in range(0,a):

```

```

    gini_l.append(compute(attr, dataset, i, n))
print(gini_l)
m=1
na=0
for i in range(0,a):
    if gini_l[i] < m:
        m = gini_l[i]
        na = i
return(na)

```

```

def update(attr, dataset, na, v, a, n):
    ndataset = [[0 for i in range(n)] for i in range(a)]
    nattr = []
    for i in range(0,a):
        if i != na:
            nattr.append(attr[i])
    k = 0
    ngini_l = []
    for j in range(0,n):
        if dataset[na][j] == v:
            y = 0
            for i in range(0,a+1):
                if i != na:
                    ndataset[y][k] = dataset[i][j]
                y += 1
            k += 1
    nna = next(nattr, ngini_l, ndataset, a-1, k)
    return(nattr[nna])

```

```

def alist(attr, dataset, na, a, n, tree):
    yes = []
    no = []
    noi = []
    val = []
    tree.append(attr[na])
    tab(yes, no, noi, val, attr, dataset, na, n)
    for i in range(0,len(val)):
        tree.append(val[i])
        if yes[i] == 0:
            tree.append("No")
        elif no[i] == 0:
            tree.append("Yes")
        else:
            v = val[i]
            nattr = update(attr, dataset, na, v, a, n)
            tree.append(nattr)

```

```
print(tree)
```

```
if __name__ == "__main__":  
    attr = []  
    gini_l = []  
    tree = []  
    a = int(input("Enter the number of attributes : "))  
    n = int(input("Enter the size of dataset : "))  
    dataset = ip(attr, a, n)  
    na = next(attr, gini_l, dataset, a, n)  
    alist(attr, dataset, na, a, n, tree)
```

"""

O/P:

```
> python cart.py
```

Enter the number of attributes : 4

Enter the size of dataset : 8

Enter attribute 1 : HAIR

Enter attribute 2 : HEIGHT

Enter attribute 3 : WEIGHT

Enter attribute 4 : LOC

Enter target attribute : CLASS

Enter the value 0 of HAIR : B

Enter the value 1 of HAIR : B

Enter the value 2 of HAIR : BR

Enter the value 3 of HAIR : B

Enter the value 4 of HAIR : R

Enter the value 5 of HAIR : BR

Enter the value 6 of HAIR : BR

Enter the value 7 of HAIR : B

Enter the value 0 of HEIGHT : A

Enter the value 1 of HEIGHT : T

Enter the value 2 of HEIGHT : S

Enter the value 3 of HEIGHT : S

Enter the value 4 of HEIGHT : A

Enter the value 5 of HEIGHT : T

Enter the value 6 of HEIGHT : A

Enter the value 7 of HEIGHT : S

Enter the value 0 of WEIGHT : L

Enter the value 1 of WEIGHT : A

Enter the value 2 of WEIGHT : A

Enter the value 3 of WEIGHT : A

Enter the value 4 of WEIGHT : H

Enter the value 5 of WEIGHT : H

Enter the value 6 of WEIGHT : H

Enter the value 7 of WEIGHT : L

Enter the value 0 of LOC : N

Enter the value 1 of LOC : Y

Enter the value 2 of LOC : Y

Enter the value 3 of LOC : N

Enter the value 4 of LOC : N

Enter the value 5 of LOC : N

Enter the value 6 of LOC : N

Enter the value 7 of LOC : Y

Enter the value 0 of CLASS : Y

Enter the value 1 of CLASS : N

Enter the value 2 of CLASS : N

Enter the value 3 of CLASS : Y

Enter the value 4 of CLASS : Y

Enter the value 5 of CLASS : N

Enter the value 6 of CLASS : N

Enter the value 7 of CLASS : N

[0.25, 0.33333333333333337, 0.45833333333333326, 0.3]

[0.25, 0.5, 0.0]

['HAIR', 'B', 'LOC']

['HAIR', 'B', 'LOC', 'BR', 'No']

['HAIR', 'B', 'LOC', 'BR', 'No', 'R', 'Yes']

\*\*\*\*