EXPERIMENT 8

Implement Knowledge Representation Schemes

AIM:

To implement Knowledge representation schemes in expert medical diagnosis system.

ALGORITHM

1. Define Symptom-Checking Functions:

- Define four functions: `measles(a, b, c, d, e)`, `flu(a, b, c, d, e, f, g, h)`, `cold(c, j, k, d, h)`, and `chickenpox(a, h, g, b)`.

- Each function takes as input a set of symptoms as 'y' (yes) or 'n' (no) and checks for specific combinations of symptoms.

- If a combination of symptoms matches a known illness, the function returns the name of the illness (e.g., "measles," "flu"). If not, it returns `None`.

2. User Input and Symptom Gathering:

- Create a function called `run_diagnosis()` to interact with the user.

- Prompt the user to enter their name and a series of yes/no responses for various symptoms.

- Store the user's responses in variables (e.g., a, b, c, etc.), ensuring all responses are in lowercase.

3. Diagnosis Initialization:

- Initialize an empty list called `diagnosis` to store the possible diagnoses based on the symptoms.

4. Diagnosis for Each Illness:

- Use each symptom-checking function to determine if the user's symptoms match any known illnesses.

- If a function returns a diagnosis (i.e., not `None`), add the diagnosis to the `diagnosis` list.

5. Final Diagnosis:

- If there are any diagnoses in the `diagnosis` list, print a message to the user, including their name and the list of possible diagnoses.

- If no diagnoses were found (the list is empty), inform the user that no diagnosis could be made based on the given symptoms.

6. Execution:

- Call the `run_diagnosis()` function to execute the symptom-based diagnosis process.

4..PROGRAM

```
def measles(a, b, c, d, e):
    if a == 'y' and b == 'y' and c == 'y' and d == 'y' and e == 'y':
        return "measles"
    else:
        return None


def flu(a, b, c, d, e, f, g, h):
    if a == 'y' and b == 'y' and c == 'y' and d == 'y' and e == 'y' and f == 'y' and g == 'y' and h == 'y':
        return "flu"
    else:
        return None


def cold(c, j, k, d, h):
    if c == 'y' and j == 'y' and k == 'y' and d == 'y' and h == 'y':
        return "cold"
    else:
        return None


def chickenpox(a, h, g, b):
    if a == 'y' and h == 'y' and g == 'y' and b == 'y':
        return "chickenpox"
    else:
        return None
```

```
def run_diagnosis():
    name = input("Please enter your name: ")
    a = input("Do you have fever? (y/n): ").lower()
    b = input("Do you have rashes? (y/n): ").lower()
    c = input("Do you have headache? (y/n): ").lower()
    d = input("Do you have running nose? (y/n): ").lower()
    e = input("Do you have conjunctivitis? (y/n): ").lower()
    f = input("Do you have cough? (y/n): ").lower()
    g = input("Do you have ache? (y/n): ").lower()
    h = input("Do you have chills? (y/n): ").lower()
    i = input("Do you have swollen glands? (y/n): ").lower()
    j = input("Do you have sneezing? (y/n): ").lower()
    k = input("Do you have sore throat? (y/n): ").lower()

    diagnosis = []

    result = measles(a, f, e, d, b)
    if result:
        diagnosis.append(result)

    result = flu(a, c, g, e, h, k, f, d)
    if result:
        diagnosis.append(result)

    result = cold(c, j, k, d, h)
    if result:
        diagnosis.append(result)
```

```
    result = chickenpox(a, h, g, b)
    if result:
        diagnosis.append(result)


    if len(diagnosis) > 0:
        print(f"{name}, based on the symptoms provided, you may have: {', '.join(diagnosis)}.")
    else:
        print("No diagnosis could be made based on the given symptoms.")


run_diagnosis()
```

OUTPUT

```
Please enter your name:  Nikhil
Do you have fever? (y/n):  y
Do you have rashes? (y/n):  y
Do you have headache? (y/n):  n
Do you have running nose? (y/n):  y
Do you have conjunctivitis? (y/n):  n
Do you have cough? (y/n):  y
Do you have ache? (y/n):  y
Do you have chills? (y/n):  y
Do you have swollen glands? (y/n):  n
Do you have sneezing? (y/n):  y
Do you have sore throat? (y/n):  y
Nikhil, based on the symptoms provided, you may have: chickenpox.
```

RESULT:

Knowledge representation schemes in expert medical diagnosis system is implemented and the output is verified.