

```

print("Nim game!!\nWe are having 12 tokens")          #nim(2)
def g(t):
    global T
    print("How many tokens would you like to take? ", end="")
    t = int(input())
    if t < 1 or t > 3: print("Number must be between 1 and 3.\n"); g(T); return
    T = T - t
    print('You take', t, 'tokens.\n', T, 'tokens remaining.\n')
def c(t):
    global T
    t = T % 4
    T = T - t
    print('Computer takes', t, 'tokens.\n', T, 'tokens remaining.\n')
T = 12
while T > 0:
    g(T)
    if T <= 0: break
    c(T)
print("Computer wins!")

```

```

import heapq
a*(3)

def h(n, g): return abs(n[0] - g[0]) + abs(n[1] - g[1])

def a(graph, start, goal):
    queue, visited = [(0, start)], set();
    while queue:
        cost, node = heapq.heappop(queue)
        if node == goal:
            return cost
        if node in visited:
            continue
        visited.add(node)
        for neighbor in graph[node]:
            heapq.heappush(queue, (cost + graph[node][neighbor] + h(neighbor,
goal), neighbor))
    return -1

graph = {(0, 0): {(0, 1): 1, (1, 0): 1}, (0, 1):
        {(0, 0): 1, (1, 1): 1}, (1, 0):
        {(0, 0): 1, (1, 1): 1}, (1, 1):
        {(0, 1): 1, (1, 0): 1}}

start = (0, 0)
goal = (1, 1)
print("Shortest path cost:", a(graph, start, goal))

```

M, m = 1000, -1000

(alpha beta (4))

```
def minimax(d, n, p, v, a, b):
```

```
    if d == 3: return v[n]
```

```
    b = m if p else M
```

```
    for i in range(2):
```

```
        val = minimax(d + 1, n * 2 + i, not p, v, a, b)
```

```
        b = max(b, val) if p else min(b, val)
```

```
        a = max(a, b) if p else a
```

```
        if b <= a: break
```

```
    return b
```

```
if __name__ == "__main__":
```

```
    v = [10, 9, 14, 18, 5, 4, 50, 3]
```

```
    print("The optimal value is:", minimax(0, 0, True, v, m, M))
```

```

def fuzzy_union(s1, s2):                                fuzzy(5)
    u={}
    for e in s1:u[e]=max(s1[e],s2.get(e,0))
    for e in s2:u[e]=s2[e]if e not in u else u[e]
    return u
def fuzzy_intersection(s1, s2):
    i={}
    for e in s1:i[e]=min(s1[e],s2[e])if e in s2 else 0
    return i
def display_fuzzy_set(s):
    print("{",end="");[print(f"{e}: {s[e]}",end=", ")for e in s];print("}")
s1={'a':.8,'b':.6,'c':.4,'d':.2,'e':.1}
s2={'a':.7,'b':.5,'c':.3,'f':.9,'g':.4}
print("Fuzzy set 1:");display_fuzzy_set(s1)
print("\nFuzzy set 2:");display_fuzzy_set(s2)
print("\nUnion of the fuzzy sets:");display_fuzzy_set(fuzzy_union(s1,s2))
print("\nIntersection of the fuzzy
sets:");display_fuzzy_set(fuzzy_intersection(s1,s2))

```

```

def ask_question(question):
    return input(question + "\nUser: ")

def main():
    print("WELCOME TO CareAssist!")
    print("I'll ask you a few questions to assist you.")
    name = ask_question("1. What is your full name?")
    age = ask_question("2. How old are you?")
    gender = ask_question("3. What is your gender?")
    symptoms = ask_question("4. What symptoms are you experiencing?")
    duration = ask_question("5. How long have you been experiencing these symptoms (In Days)?")
    medical_history = ask_question("6. Do you have any existing medical conditions?")
    allergies = ask_question("7. Are you allergic to any medications?")
    insurance = ask_question("8. Do you have health insurance?")
    contact_info = ask_question("9. What is the best way to contact you?")
    additional_info = ask_question("10. Is there anything else you'd like to share?")
    if additional_info.lower() == "yes":
        print("\nPlease Contact: 9986075076 For Further Queries")
    print("\nThank you", name, "for providing the information. We'll get back to you shortly.")
if __name__ == "__main__":
    main()

```