Problem Definition

There is a region on the earth which is said that Youth Elixir lie there. However, going to that region is not safe because there is an active volcano and toxic faults there. So, finding the Elixir has some costs. We model the region by an n*n matrix. In some cells of the matrix (corresponding to a specific parts of the region), there may be a toxic fault. The position of faults are exactly identified. However, the amount of poison spreading from reach fault is a stochastic (random) variable having a certain distribution. Starting from an arbitrary location (cell), our goal is to reach to the cell containing the youth elixir such that we receive the least possible amount of poison (The sum of poisons spread from the cells in the path is minimum). Because the amount of poisons spreading from the cells are random, we cannot identify the best path with certainty. Instead, by employing the methods of reinforcement learning, we find a set of different paths such that the path(s) with the minimum amount of poison in the set has greater probability of being chosen.

There are two different questions we are going to address here:

Question 1)

a. Having an arbitrary policy π, we estimate the value function V^{π} using the first-visit MC method.
b. We find the optimal action-value function Q^{*} and the optimal $\epsilon - soft$ policy $\pi^{*}(s, a)$ using the method of $\epsilon - soft\ on - policy\ Mont\ Carlo.$

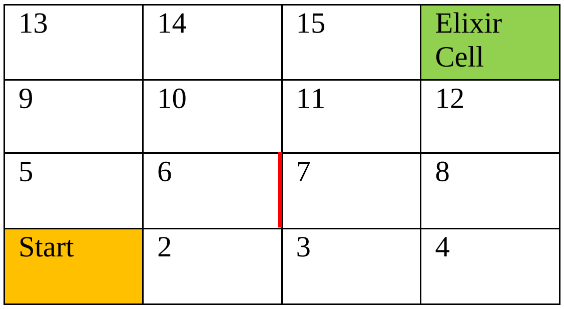
Question 2)

a. Having an arbitrary policy π, we estimate the value function V^{π} using the TD(0) method.
b. We find the optimal policy using the actor-critic method. The optimal policy $\pi^{*}(s, a)$ is produced using the Gibbs Softmax method.

The stop condition for the first part of this question is the number of episodes produced and for the second part is that the optimal path have the probability of (being chosen) more than δ for an arbitrary δ.

Input

The below picture is an arbitrary visualized model from the problem.



13	14	15	Elixir Cell
9	10	11	12
5	6	7	8
Start	2	3	4

We start from the first cell and 16th cell contains the elixir youth which is our destination. There is a fault between the 6th and the 7th cell. This fault spreads two different amount of poisons 10 and 4 with the probability of 0.8 and 0.2 respectively.

The Input file is according to the following table. We suppose the amount of poison from each fault takes at most five different values each having a different probability. So, for each fault we get the number of different poison values (n_i) and in the following n_i lines, a value and a probability comes.

Number of Questions		
Question number(e.g Q11)	Number of iterations	Number of episodes
Rows' number	Column's number	Number of faults
Start Cell	Elixir Cell	
location of first fault	n_1 (between 1-5, number of poisons)	Fault's information
first prob	first value	
second prob	second value	
…	…	
$n_1 th$ prob	$n_1 th$ value	
…	…	

Method Description

Question 1)

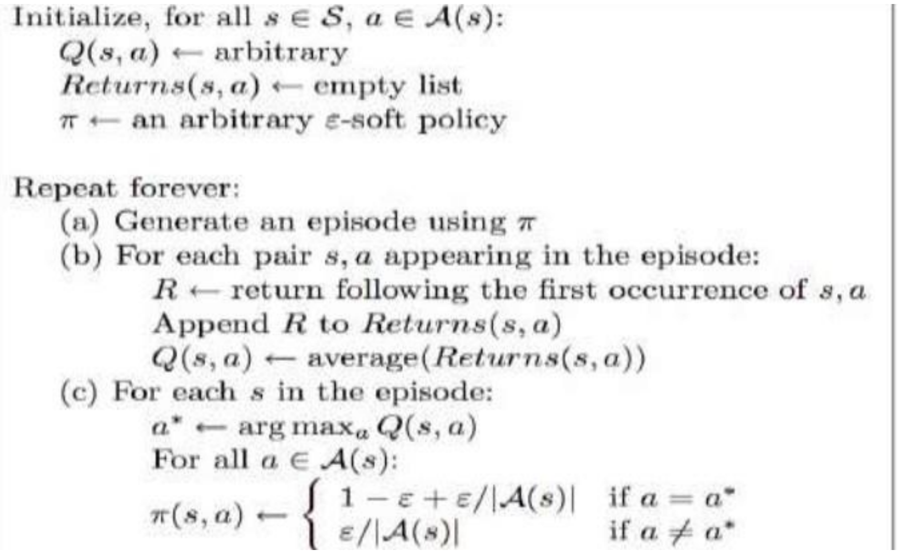
a. We are going to estimate the value function according to the following Bellman equation:

$$V^{\pi}(s) = \sum_{a} \pi(s,a) \sum_{s'} P_{ss'}^{a} [R_{ss'}^{a} + \gamma V^{\pi}(s')] \qquad equation\ (1)$$

Where $\pi(s,a)$ is the policy to be evaluated, $P_{ss'}^{a}$ is the probability of going to state s' from state s by action a which is supposed to be deterministic here and $R_{ss'}^{a}$ is the reward that the agent gets by going to state s' from state s by action a (a random negative variable for fault's cell and a certain positive variable for the elixir cell).

The main code relating to this part is in the file named "first_vist_mc.cpp". Necessary functions are in the file "common1.cpp" which is common between the first part of question 2. The main function generates n episodes and for each episode, it estimates the value of each state using the given policy. The value of each state then becomes the average of rewards seen in all the generated episodes and iterations according to the shown Bellman equation.

b. According to the following algorithm, we find the optimal policy. For implementing this algorithm, we divided it into three different phases (functions): generate episode, policy evaluation and policy improvement. These functions are in the file "esoft_mc.cpp" and the necessary functions are in the file "common2.cpp".

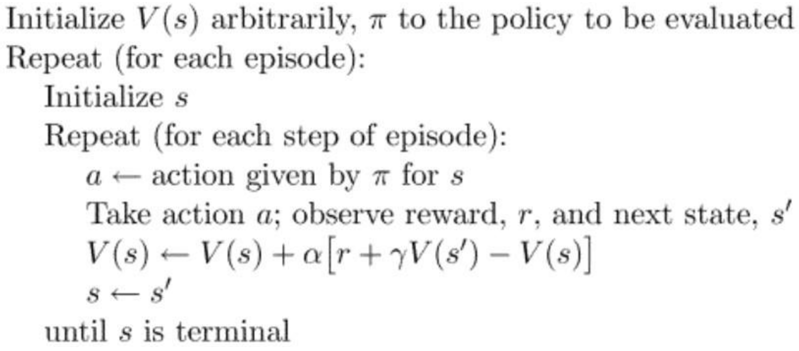


```
Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
    $Q(s, a) \leftarrow$ arbitrary
    $Returns(s, a) \leftarrow$ empty list
    $\pi \leftarrow$ an arbitrary $\varepsilon$-soft policy

Repeat forever:
    (a) Generate an episode using $\pi$
    (b) For each pair $s, a$ appearing in the episode:
            $R \leftarrow$ return following the first occurrence of $s, a$
            Append $R$ to $Returns(s, a)$
            $Q(s, a) \leftarrow$ average($Returns(s, a)$)
    (c) For each $s$ in the episode:
            $a^* \leftarrow \arg\max_a Q(s, a)$
            For all $a \in \mathcal{A}(s)$:
            $\pi(s, a) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(s)| & \text{if } a = a^* \\ \varepsilon/|\mathcal{A}(s)| & \text{if } a \neq a^* \end{cases}$
```

Question 2)

a. According to the following algorithm named TD(0), we evaluate an arbitrary given policy function. The code for this part lies in a file with the same name TD(0).



```
Initialize $V(s)$ arbitrarily, $\pi$ to the policy to be evaluated
Repeat (for each episode):
   Initialize $s$
   Repeat (for each step of episode):
      $a \leftarrow$ action given by $\pi$ for $s$
      Take action $a$; observe reward, $r$, and next state, $s'$
      $V(s) \leftarrow V(s) + \alpha\big[r + \gamma V(s') - V(s)\big]$
      $s \leftarrow s'$
   until $s$ is terminal
```

b. According to the famous actor-critic method, we try to find the optimal policy. The main functions for this part are in the file with the same name. There are three different functions (phases) in this file. The main function is the actor-critic function which generates one episode and gives it to the function actor_critic_one_episode to run one iteration of the algorithm. Then finds the optimal path and this procedure continues until the probability of optimal path reaches to a certain amount. The function actor_critic_one_episode updates the amount of preferences and the value of states according to the TD(0) method and equation 2. Then it calls the policy_update function. This function updates the probabilities of state-action according to equation 3.

$$p(s_t, a_t) \leftarrow p(s_t, a_t) + \beta\delta_t \qquad \textit{equation } (2)$$

$$\textit{where } \delta_t \textit{ is the amount of reward}$$

$$\pi_t(s,a) = \Pr\{a_t = a \mid s_t = s\} = \frac{e^{p(s,a)}}{\sum_b e^{p(s,a)}} \qquad equation\ (3)$$

Output

The output is identified according to the question. It can be a value function, an action-value function or the optimal policy found.