CS 3353 Spring 2023

Program 4 – Who Wants to Win Wads of Water? (WWWWW)

Due: ***5/6 (Sat) 9:00pm*** (No late pass, and no late submission will be considered)

At the height of the popularity of "Who wants to be a millionaire", various versions of the show was produced in other countries and were also immensely popular. In fact, it is so popular that the planet "Alpha Centuri" decided to produce their own version of the show, and call it "Who Wants to Win Wads of Water" (In Alpha Centuri, "Wads of Water" is a slang for "Lots of Money").

While they are happy with the general structure of the game, they also want to experiment with various aspect of the game (number of questions, where is the safety line for the contestants, how does the lifeline works etc.) And they want to write a program to run some tests and calculate the expected amount of money that a contestant can win (so they can have enough money to prepare).

Base case (100 points)

For the base case, the rules are as follows.

- The game consists of j questions (where j is a parameter to be provided) (You can assume $j > 3$).
- The contestant is to answer the questions one at a time.
- At any moment, the contestant can quit.
- If the contestant quits after answering k questions correctly, he/she will win w_k dollars ($1 \le k \le j$). All the values of $w_1, w_2, \ldots w_j$ are parameters to be provided, with the stipulation that the dollar amount will strictly increases as the number of correct answer increases.
- If a contestant answers a question wrong, the game ends immediately.
    - There will be two numbers $1 \le s < t < j$. If the contestant answers at least t questions correctly, he/she will receive w_t dollars. Otherwise, if the constant answer at least s questions correctly, he/she will receive w_s dollars. Otherwise, he/she will receive 0 dollars.
- There are no lifelines available.
- For question k, we will assume the contestant has a probability of p_k of answering that question correctly. ($0.25 \le p_k < 1$)

I will have a program that reads in the parameters from a file. The file will be formatted as follows.

- The first line has 3 numbers, which are j, s, t respectively
- The second line has j numbers, which corresponds to $w_1, w_2, \ldots w_j$ respectively. (Notice they can be floating point numbers)
- The third line has j numbers which corresponds to $p_1, p_2, \ldots p_j$ respectively. (You can assume each number is within the range).

To make things easily for indexing, we set $w_0 = p_0 = 0$

Task

You are to implement the following function:

pair< vector<float>, vector<int> > WWWWW(vector<float> w, vector<float> p, int s, int t)

(You should infer the value of j based on the length of the array)

This program will return a pair of vectors

- The first vector contains the expected amount of money won. The i-th entry stores the expected amount of money won after the contestant answer i questions correctly.
- The second vector stores 0 or 1 The (i-1)-th entry stores whether the contestant should quit when he face question i (recall there is no question 0).

You must use the dynamic programming algorithm we discussed in class.

Bonus 1 (15 points)

Now suppose the following lifeline is available:

- Get easier: Once the lifeline is used, the probability of answering the question correctly becomes 0.5 + p/2.0 (where p is the original probability). However, the new probability cannot be more than 0.999
- Once you decide to use the lifeline, you MUST answer the question.

Now implement the following function:

pair< vector < vector<float> > ,vector < vector<int> > > WWWWW_1(vector<float> w, vector<float> p, int s, int t)

where the return elements is now two vectors of vectors (think of it two 2-D arrays)

- Assume the first *vector < vector<float> >* be named a.
  - a[0][i] corresponds to the expected amount of money won after answering the i-th questions correct, but with no lifeline available.
  - a[1][i] corresponds to the expected amount of money won after answering the i-th questions correct, but with the lifeline still available.
- Assume we name second *vector <vector <int> >* b.
  - b[0][i-1] corresponding to the decision (0 = quit, 1 = answer) whether the contestant should quit (0) or answer question i (1), when no lifeline is available
  - b[1][i-1] corresponds to the decision whether the contest should quit (0) continue without using the lifeline on question i (1) and continue but use the lifeline on question i (2), assuming the lifeline is available

Bonus 2 (25 points)

Now suppose you have a second lifeline available:

- Get through: if you use the lifeline, the game will provide you with the right answer to the current question (i.e. p = 1 for that question). However, but v * 10 dollars will be subtracted from your final winnings (where v is the number of question that you answer correctly immediately before you apply to lifeline).
- You cannot use both lifelines on the same question.

Now implement the following function:

pair< vector < vector<float> > ,vector < vector<int> > > WWWWW_2(vector<float> w, vector<float> p, int s, int t)

where the return elements is now two vectors of vectors (think of it two 2-D arrays)

- Assume the first *vector < vector<float> >* be named a.
  - a[0][i] corresponds to the expected amount of money won after answering the i-th questions correct, but with no lifeline available.
  - a[1][i] corresponds to the expected amount of money won after answering the i-th questions correct, but with the “get easier” lifeline (only) still available.
  - a[2][i] corresponds to the expected amount of money won after answering the i-th questions correct, but with the “get through” lifeline (only) still available.
  - a[3][i] corresponds to the expected amount of money won after answering the i-th questions correct, but with both lifelines still available.

- Assume we name second *vector <vector <int> >* b.
  - b[0][i-1] corresponding to the decision (0 = quit, 1 = answer) whether the contestant should quit (0) or answer question i (1), when no lifeline is available
  - b[1][i-1] corresponds to the decision whether the contest should quit (0) continue without using the lifeline on question i (1) and continue but use the lifeline on question i (2), assuming the “get easier” lifeline (only) is available
  - b[2][i-1] corresponds to the decision whether the contest should quit (0) continue without using the lifeline on question i (1) and continue but use the lifeline on question i (3), assuming the “get through” lifeline (only) is available
  - b[3][i-1] corresponds to the decision whether the contest should quit (0) continue without using the lifeline on question i (1) and continue but use the “get easier”(2) / “get through” lifeline on question i (3), assuming both lifelines is available

You will be given a program “hw4prog.cpp” which contains the code to run the program. You must make sure that your function compiles correctly with it.

You are to hand in a single cpp file that contains the function(s).