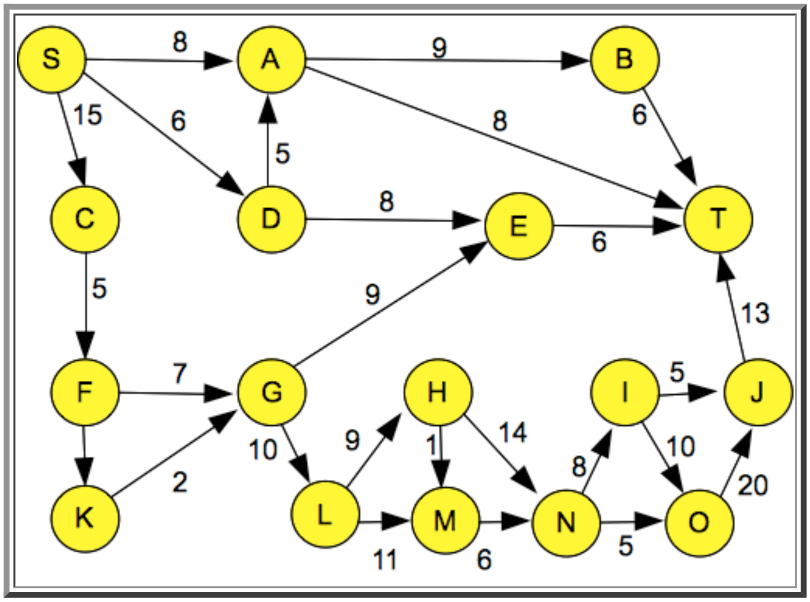
Question 5

Behold the following graph with source **S** and sink **T**.



Part A: What is the maximum flow of this graph?

Part B: What edges compose a minimum cut of this graph?

Part C: Using the Edmonds-Karp algorithm, what are the augmenting paths to find the maximum flow? If there are multiple alternatives, just show a legal one.

Part D: Draw the final flow graph that results when the Edmonds-Karp algorithm is used to find the maximum flow. If there can be more than one final flow graph, just draw one.

I have included an answer sheet for you to hand in your final answer, and some work sheets for your intermediate calculations. Just hand in the final answer. I don't want to see your work.

Question 6

A number is "righteous" if it fits the following definition:

- 1, 2, 3 and 4 are righteous numbers.
- Suppose i is a d digit righteous number whose last digit is l. Let k equal $l+3$. If $k < 10$, then $(i*10 + k)$ is a righteous number.
- Suppose i is a d digit righteous number whose last digit is l. Let m equal $l-2$. If $m \geq 0$, then $(i*10 + m)$ is a righteous number.

So, 20, 142, and 4758 are all righteous numbers. 5, 41 and 470 are not.

Write a program **righteous.cpp** that takes one command line argument *n* and prints out the number of righteous numbers that have exactly *n* digits.

The running time of this should be $O(n)$. Use dynamic programming, either with memoization or without recursion. I did both, and I think that doing it without recursion is easier. Hint: how can you use **cache[d][l]**, where **d** is the number of digits and **l** is the last digit?

You may assume that *n* is less than or equal to 40. The following table computes **righteous** for **i** between 1 and 10:

n	1	2	3	4	5	6	7	8	9	10
righteous *n*	4	7	11	18	28	46	71	112	178	278