```
/*
* Author: Linh Phan
* Course: CS1
* Assignment: Assignment 3
* Date: 4/6/2025 Spring 2025
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

//trie struct
typedef struct trienode {
      struct trienode *children[26];
      int flag;
      int frequency;
      int maxFrequency;
      char maxFreqWord[100];
} trienode;

// Function prototypes
trienode *createTrieNode();
void insertWord(trienode *root, char *string, int frequency);
char* write(trienode* root, char str[]);

//creating node
trienode *createNode() {
      trienode *node = NULL;
      node = (trienode *)malloc(sizeof(trienode));
      if (node != NULL) {
            node->flag = 0;
            for (int x = 0; x < 26; ++x) {
                  node->children[x] = NULL;
            }
      }
      return node;
}

//for first function or option to add a word
void insertWord(trienode *root, char *string, int freq) {
    //variables
      int length = strlen(string);
      trienode *temp = root;
    //adding word provided by string and then checking freq to assign it to
maxFreqWord
      for (int i = 0; i < length; i++) {
            int index = string[i] - 'a';
            if (temp->children[index] == NULL) {
                  temp->children[index] = createNode();
            }
            temp = temp->children[index];
            //assigning it to maxFrequencyWord
            if(freq > temp->maxFrequency) {
                  temp->maxFrequency = freq;
                  strcpy(temp->maxFreqWord, string);
            }

      }
    //setting flag to 1 to signify the end of word
      temp->flag = 1;
}

//for second function or option to print a word
char* write(trienode* root, char* string) {
    //setting variables
      int length = strlen(string);
      trienode *temp = root;
      char strauto[100];
      int count = 0;

    //checking the prefix and getting temp to that node
      for (int i = 0; i < length; i++) {
            strauto[count] = string[i];
            count++;
            int index = string[i] - 'a';
            if (temp->children[index] == NULL) {
                  return ("unknown word\n");
            }
            temp = temp->children[index];
      }

    return (temp->maxFreqWord);
}

int main()
{
    //setting variables
      trienode *root = createNode();
      int count = 0;
      int option;
      char string[100];
      char prefix[100];
      int frequency;

    //checking how many commands
      scanf("%d", &count);

      //looping to have the same instructions count
      for(int i = 0; i < count; i++) {
          //checking if it's inserting words or printing maxFreqWord
            if (scanf("%d", &option)) {
                  if (option == 1) {
                        scanf("%s %d", string, &frequency);
                        insertWord(root, string, frequency);
                  }
                  else if (option == 2) {
                        scanf("%s", prefix);
                        printf("%s\n", write(root, prefix));
                  }
            }
      }
      return 0;
}
```