Code:

```
#include <stdio.h>
#include <limits.h>

// Function to find the maximum of two numbers
int max(int a, int b) {
  return (a > b) ? a : b;
}

// Function to find the minimum of two numbers
int min(int a, int b) {
  return (a < b) ? a : b;
}

// Minimax function
int minimax(int depth, int nodeIndex, int isMax, int scores[], int h) {
  // If we reach the leaf node
  if (depth == h)
    return scores[nodeIndex];

  // If current move is of maximizer
  if (isMax)
    return max(
      minimax(depth + 1, nodeIndex * 2, 0, scores, h),
      minimax(depth + 1, nodeIndex * 2 + 1, 0, scores, h)
    );
  else // Minimizer's move
    return min(
      minimax(depth + 1, nodeIndex * 2, 1, scores, h),
      minimax(depth + 1, nodeIndex * 2 + 1, 1, scores, h)
    );
```

```
}

// Function to calculate log base 2
int log2int(int n) {
   int r = 0;
   while (n > 1) {
      n /= 2;
      r++;
   }
   return r;
}

int main() {
   // Example leaf nodes (scores)
   int scores[] = {3, 5, 2, 9};
   int n = sizeof(scores)/sizeof(scores[0]);
   int h = log2int(n);

   int result = minimax(0, 0, 1, scores, h);
   printf("The optimal value is : %d\n", result);
   return 0;
}
```

OUTPUT:

The optimal value is : 5