```c
#include <string.h>
#include <stdio.h>
#include <stdint.h>

int8_t is_ascii(char s[]) {
  for(int i = 0; s[i] != 0; i += 1) {
    if(s[i] < 0 || s[i] > 127) {
      return 0;
    }
  }
  return 1;
}

// Assuming c is the start byte of a utf8 code point,
// return the number of bytes for that code point
uint8_t width_from_start_byte(char c) {
  if((c & 0b10000000) == 0) { return 1; }
  if((c & 0b11100000) == 0b11000000) { return 2; }
  if((c & 0b11110000) == 0b11100000) { return 3; }
  if((c & 0b11111000) == 0b11110000) { return 4; }
  return 0;
}

// Count the number of _code points_ in the string
int32_t utf8_len(char s[]) {









}

int main() {
  char input[100];
  fgets(input, 100, stdin);
  printf("%s\n", input);
  printf("Is ASCII: %d\n", is_ascii(input));
  printf("Width of first byte: %d\n", width_from_start_byte(input[0]));


}
```

```
$ gcc utf8analyzer.c -o utf8analyzer
$ ./utf8analyzer
éclaire
éclaire

Is ASCII: 0
Width of first byte: 2
```