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

int8_t is_ascii(char c) {
    return c <= 128;
}

void test_is_ascii(char c) {
    printf("is_ascii(%c %d) = %s\n", c, c, is_ascii(c) ? "true" : "false");
}
void tests_is_ascii() {
    test_is_ascii('a');
    test_is_ascii(0b10000000);
    test_is_ascii(0b01111111);
    for(int i = 0b00000000; i <= 0b11111111; i += 1) {
        test_is_ascii(i);
    }
}

int main() {
    tests_is_ascii();
}
```

```
$ gcc numbers.c -o numbers
$ ./numbers
is_ascii(a 97) = true
is_ascii(? -128) = true
is_ascii( 127) = true
is_ascii( 0) = true
is_ascii( 1) = true
is_ascii( 2) = true
... lots of output ...
is_ascii(z 122) = true
is_ascii({ 123) = true
is_ascii(| 124) = true
is_ascii(} 125) = true
is_ascii(~ 126) = true
is_ascii( 127) = true
is_ascii(? -128) = true
is_ascii(? -127) = true
... lots more output
is_ascii(? -1) = true
```

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

int32_t code_point2(char c1, char c2) {
    return (c1 & 0b00011111) * 64 + (c2 & 0b00111111);
}

int32_t code_point3(char c1, char c2, char c3) {
    return (c1 & 0b00001111) * 4096 + (c2 & 0b00111111) * 64 + (c3 & 0b00111111);
}

int main() {
    char joseph[] = "Joséph";
    printf("Code point: %d\n", code_point2(joseph[3], joseph[4]));



}
```