

Solution for problem 8:

This one is pretty straight forward, write a forward recursion, keep dividing upto \sqrt{n} and if none succeeds, the number is a prime, else not.

```
[.c]
#include <stdio.h>
#include <math.h>

int isprime(int i, int rt, int n)
{
    if(n < 2) return 0;
    if(i > rt) return 1;
    if(n%i==0) return 0;
    return isprime(i+1, rt, n);
}

int main()
{
    int n;
    while(scanf("%d", &n)==1)
    {
        if(isprime(2, (int)sqrt(n), n)==1) printf("prime\n");
        else printf("not prime\n");
    }
    return 0;
}

[.c]
```

Well this code is not a good one, and actually we don't test primes in such a weird way, but you know, exam always sucks...