

1. Да се напише рекурзивна функција со која ќе преврти одреден број.

```
#include <stdio.h>

int function(int x) {
    static int y = 0;
    if (x == 0) return y;
    y = y * 10 + x % 10;
    return function(x/10);
}

int main()
{
    int x;
    printf("Vnesi broj:\n");
    scanf("%d", &x);
    printf("%d", function(x));
    return 0;
}
```

2. Да се напише рекурзивна функција која ќе дозволи внесување на непознат број на цели броеви. Рекурзивната функција треба да најде колку парни броеви се внесени. Рекурзијата прекинува кога ќе се внесе било што различно од број.

Пример: 12, 5, 16, 2, 5, а -> се печати 3

```
#include <stdio.h>

int x;

int fr() {
    if (scanf("%d", &x) == 0) return 0;
    return (x % 2 == 0) ? (1 + fr()) : fr();
}

int main()
{
    printf("%d", fr());
    return 0;
}
```

3. Да се напише функција која прима 3 аргументи. Функцијата треба да врати нов број кој е формиран со замена на цифрата која се наоѓа на дадена позиција во бројот со цифра која се пренесува како аргумент (позицијата се брои од десно кон лево, почнувајќи од еден). Да се напише главна програма во која се тестира работата на функцијата.

Дополнително, задачата решете ја и со рекурзивна функција.

Пример. За f(2376, 3, 5) -> 2576

```

#include <stdio.h>

int fr(int x, int o, int y) {
    if (x == 0) return 0;
    return fr(x/10, o-1, y)*10 + ((o == 1) ? y : x % 10);
}

int f(int x, int o, int y) {
    int xr = 0;
    int stepen = 1;
    while (x > 0) {
        if (o == 1)
            xr = xr + y * stepen;
        else
            xr = xr + (x % 10) * stepen;
        stepen *= 10;
        o--;
        x /= 10;
    }

    return xr;
}

int main()
{
    printf("%d", f(123456, 4, 9));
    return 0;
}

```

4. Да се напише програма во која од тастатура се внесуваат почеток и крај на интервал на цели броеви. Програмата треба да ги испечати сите броеви од интервалот за кои ќе важи следното: превртениот еквивалент на бројот да биде прост број. На крај да се испечати колку такви броеви постојат.

Пример:

71 – превртен еквивалент – 17 – прост број

23 – превртен еквивалент – 32 – не е прост број // нема да го печати

```

#include<stdio.h>

int eprost(int x) {
    int j;
    for(j = 2; j <= x/2; j++) {
        if (x % j == 0) return 0;
    }
    return 1;
}

int prev(int x) {
    int xr = 0;
    while(x > 0) {
        xr = xr * 10 + x % 10;
        x /= 10;
    }
    return xr;
}

int main() {
    int i, begin, end;
    int br = 0;

    scanf("%d %d", &begin, &end);

    for(i = begin; i <= end; i++) {
        if(eprost(prev(i))) {
            printf("Broj %d.\n", i);
            br++;
        }
    }

    printf("Najdeni se takvi %d broevi.", br);
    return 0;
}

```