Problem 3.2.

ALEXANDRU
MIHAI
GLONȚARU

a) • recursive function

```
recval f: int -> int = (x: int) -> F(x),
    val main: int = f(1)

F(x) := if (x<=0) {1}
        else { x * f(x-1)}
```

• while loop

```
val f: int -> int = (x: int) -> F(x),
    val main: int = f(1)

F(x) := {
    var result = 1,
    while (x>0) {
        result = result * x,
        x = x-1 },
    result
}
```

b) • f(1) ⇝ ((x:int) ⇝ F(x)) • 1 ⇝ F(1) ⇝ 1 • fact(1-1) ⇝ 1 • fact(0) ⇝ 1 • F(0) ⇝ 1 • 1 ⇝ 1

• f(1) ⇝ ((x:int) ⇝ F(x)) • 1 ⇝ F(1)

F(1) ⇝ result = 1

while (1>0) ⇝ { result = 1 • 1, x = 1-0} ⇝ ..

.. ⇝ wh(0>0) ⇝ result ⇝ 1

ALEXANDRU
MIHAI
GLOMȚARU

Problem 3.3:

$\Gamma ::= decl^*$

$decl ::= type \quad a = A$

$decl ::= type \quad b = B$

→ productions to the grammar:

$decl ::= type \quad u = A + B$

→ typing rules:

$$\frac{\Gamma \vdash A : type \quad \Gamma \vdash B : type}{\Gamma \vdash A + B : type}$$

→ evaluation rules:

$$\frac{\Gamma \vdash inj_1(a:A) \to u : A+B; \quad \Gamma \vdash inj_2(b:B) \to v : A+B; \quad \Gamma \vdash f(a:A) \to c : C; \quad \Gamma \vdash g(b:B) \to c : C}{cases(u, f, g) \to f(a); \qquad cases(v, f, g) \to g(a)}$$