Esecizio 1
(a) Si completino le seguenti definizioni Caml

```
type coppia = {uno:int;due:int};;
type terna = {c:coppia;...};;
type tuple = C of coppia | T of ...;;
let rec fun_C f (x:...) =
    match x with
      C y -> f (y.uno) (y.due) | ... -> f(fun_C ...) (y.tre)
    ;;
```

che introducono il tipo concreto tuple come unione dei tipi coppia e terna
ed un'operazione fun_C che, applicata ad una funzione f binaria, ed ad una
tupla x, restituisce il valore ottenuto applicando f ai componenti della tupla.
(b) In accordo alle definizioni date in (a), si completino le definizioni sotto
in modo tale che c sia una coppia di interi k1 e k2, e t sia una terna di interi
k1, k2, k3, per tre arbitrari k1, k2, k3,

```
let k1 = ...
let k2 = ...
let k3 = ...
let c = C(...)
let t = T(...)
```

(c) e r1 sia completato con un'espressione che applica fun_C alla funzione somma
(di interi) e al valore c; e r2 sia completato con un'espressione che applica fun_C
alla funzione minimo (di interi) e valore t;

```
let r1 = ...
let r2 = ...
```

(d) Si Riscrivano i tipi concreti coppia, terna, tuple in tre tipi astratti CoppiaOfT,
TternaOfT, TupleOfT, polimorfi che esportino le operazioni necessarie per definire
la funzione fun_C.
(e) Si Riscriva la funzione fun_C in una funzione fun_cOfT che si applica, ora ad una
polimorfa f ed ad un valore TupleOfT.
(f) Si mostri l'espressione che applica fun_cOfT alla somma e al corrispondente TupleOfT,
cc, di c.
(f) Si mostri l'espressione che applica fun_cOfT al prodotto e al corrispondente TupleOfT,
tt, di t.