

Università Ca' Foscari
Dipartimento di informatica

Programmazione part-time

Esame Ocaml

Nome: _____

Matricola: _____

Samuel Rota Bulò , a.a. 2009/2010

Teoria. (10 punti) Rispondere ai seguenti quesiti utilizzando eventualmente gli appositi spazi bianchi.
Tempo totale a disposizione: 20 min.

(2^{pts}) 1. Dato il seguente codice:

```
let f x = fun y -> x * y * 2;;
```

quale delle seguenti affermazioni è corretta? (*Barrare la risposta esatta*)

- (a) `f 2. 3. = 12.` (b) `f` è di tipo `int -> int` (c) `(f 2)` è una funzione che
quadruplica il suo argomento
- (d) l'espressione è errata (e) `f 2 = 4`

2 pts

(3^{pts}) 2. Scrivere una funzione che somma i primi due elementi di una lista di interi. Se la lista ha un solo elemento deve ritornare quell'elemento e se la lista è vuota deve ritornare 0.

Per esempio:

- con `[1;2;3]` otteniamo 3
- con `[1]` otteniamo 1
- con `[]` otteniamo 0

3 pts

(2^{pts}) 3. Qual è il tipo della funzione `somma_primi_due` definita nell'esercizio precedente?

```
val somma_primi_due : _____
```

2 pts

(3^{pts}) 4. Date le seguenti variabili:

```
let f = function _ -> fun x -> x + x ;;  
let h x = if x>0 then true else 0;;  
let g = fun f -> f 2 ;;
```

dire quale di queste affermazioni è vera (**V**) o falsa (**F**). (*0.5 punti per risposta corretta*).

- (a) _____ `f 2 = 4.`
- (b) _____ il tipo di `f` è `('a -> int)-> int.`
- (c) _____ `g (*)` è una funzione che raddoppia il suo argomento.
- (d) _____ `h 0 = 0.`
- (e) _____ `g 2` produce un errore di tipo.
- (f) _____ il tipo di `g` è `(int -> 'a)-> 'a.`

3 pts

Pratica. (23 punti) Nello svolgimento del seguente esame, il candidato crei una cartella con il proprio cognome, numero di matricola e la lettera L (e.g. Rossi887766L) inserendo all'interno i file corrispondenti agli esercizi che si intendono consegnare (e.g. Esercizio1.ml). Le prime righe del file devono essere dei commenti che specifichino il vostro nome e cognome e l'esercizio a cui si riferiscono.

Tempo a disposizione: 1h e 45 min.

- (6^{pts}) **1.** Scrivere una funzione che date due stringhe verifichi se una delle due è prefisso dell'altra. Per esempio, con "ocaml" e "esame" ritorna falso. Con "gatto" e "gattopardo" ritorna vero. Con "leone" e "leopardo" ritorna falso. Con "pescecane" e "pesce" ritorna vero. È consentito utilizzare le seguenti funzioni:

- `val length : string -> int`
`String.length s` ritorna la lunghezza della data stringa.
- `val get : string -> int -> char`
`String.get s n` ritorna l'*n*-esimo carattere nella stringa *s*.

6 pts

- (17^{pts}) **2.** Battaglia navale. In un campo da gioco 10×10 , cioè in cui ogni casell è individuata da una coppia di coordinate (x, y) con $1 \leq x, y \leq 10$, sono piazzate delle navi alle quali è associato un identificatore. Lo stato del campo di gioco è descritto da una lista di celle. Ciascuna cella contiene l'identificatore della nave e le sue coordinate. Notiamo che una nave può essere composta da più celle e quindi nella lista lo stesso identificatore può comparire più volte. Lo stato del campo di gioco deve essere gestita con una lista di triplette, ciascuna rappresentante una cella. Ogni tripletta memorizza le coordinate della cella nei suoi primi due elementi e la stringa identificatore della nave nel terzo.

- (a) (1 pt) Definire una variabile globale e inizializzarla con una possibile stato del campo di gioco.
- (b) (3 pts) Scrivere una funzione cerca nave che dato uno stato del campo di gioco e un identificatore nave verifichi se la nave è ancora presente sul campo (ritorna vero se c'è, falso altrimenti).
- (c) (3 pts) Scrivere una funzione che dato uno stato del campo di gioco rimuova eventuali celle posizionate fuori dal campo di gioco.
- (d) (4 pts) Scrivere una funzione che dato un campo di gioco conti il numero di navi (non celle !) presenti sul campo.
- (e) (1 pt) Definire un nuovo tipo `esito_colpo` per gestire 3 stati: Acqua, Colpito, e Affondato. Gli stati Colpito e Affondato devono poter contenere l'identificatore della nave colpita/affondata.
- (f) (5 pts) Scrivere una funzione spara che gestisca il colpo del giocatore. La funzione prende in ingresso lo stato del campo di gioco e la coordinata a cui si spara. La funzione ritorna in una coppia il nuovo stato del campo di gioco e l'esito del colpo. Distinguiamo 3 possibili scenari caratterizzanti il valore di ritorno della funzione:
 - Il colpo è andato a vuoto. Il nuovo stato del campo di gioco sarà quindi uguale a quello in ingresso e l'esito del colpo sarà Acqua.
 - È stata colpita l'ultima cella appartenente ad una certa nave *x* sul campo. Il nuovo stato del campo di gioco sarà quello precedente meno la cella colpita e l'esito del colpo sarà Affondato *x*.
 - È stata colpita la cella appartenente ad una certa nave *x*, ma la nave è ancora presente sul campo. Il nuovo stato del campo di gioco sarà quello precedente meno la cella colpita e l'esito del colpo sarà Colpito *x*.

17 pts