

TEST DI PROGRAMMAZIONE - 05 giugno 2013

È necessario rispondere correttamente ad almeno 7 domande su 10 affinché venga valutata la seconda parte di esercizi.

Nome	Cognome
Matricola	Email

1. Data la seguente funzione:

```
let rec f x y = match x with
  [] -> false
  | z::zs -> (y=z) or f zs y;;
```

Qual è il tipo della funzione ?

- A. int list -> int -> bool B. 'a list -> 'a -> bool
C. int list -> int -> int list D. 'a list -> 'a -> 'a list

1. _____

2. Data la funzione *f* dell'esercizio 1, dire quali delle seguenti chiamate di funzione è corretta.

- A. *f*([1;2;4;5], [4]) B. *f*(3, [1;3;5]) C. *f*([5], 1.0) D. *f*(['a';'b';'z'], 'd')
E. *f*('a', [1]) F. *f*([1.0;3.4], [2.5])

2. _____

3. Data la funzione *f* dell'esercizio 1, dire che cosa restituisce.

- A. true se *y* è un elemento della lista *x*, false altrimenti B. true se *x* è un elemento della lista *y*, false altrimenti
C. false se *x* è la lista vuota, true altrimenti D. true se la lista è formata da elementi tutti uguali tra loro, false altrimenti

3. _____

4. Dire quale delle seguenti espressioni ha come valore float list = [2.0;4.8;7.9;5.1;9.6]:

- A. [2.0;4.8;7.9]::[5.1;9.6] B. [2.0;4.8;7.9]@[5.1;9.6] C. [2.0]::[4.8;7.9;5.1;9.6]
D. 2.0@[4.8;7.9;5.1;9.6]

4. _____

5. Data la seguente funzione:

```
int foo(int a, int b) {
  if (a == b)
    return a;
  else
    return (a>b) ? foo(a-b, b) : foo(a, b-a);
}
```

quanto vale l'espressione *foo*(18, 60)?

- A. 6 B. 60 C. 18 D. La ricorsione non termina

5. _____

ESAME DI PROGRAMMAZIONE
(Programmazione Funzionale)

Cognome:

Nome:

Matricola:

Data: 05/06/2013

Svolgere i seguenti quesiti nel foglio protocollo. Consegnare: presente testo, bella e brutta copia. Tempo previsto: 60 minuti.

1. **Esercizio 1**

Definire una funzione ricorsiva `count_odd_digits: int -> int` che conta il numero di cifre dispari che compaiono nella rappresentazione decimale di un numero intero. Per esempio:
`(count_odd_digits 12896) = 2` e `(count_odd_digits 1279) = 3`.

2. **Esercizio 2**

Definire le funzioni `take` e `drop` che prendono come argomenti un numero naturale `n` e una lista `l` in modo tale che: `take n l` restituisca la lista contenente i primi `n` elementi della lista `l` e `drop n l` restituisca la lista contenente gli ultimi elementi della lista `l` esclusi i primi `n`. Inoltre si scriva il tipo delle funzioni `take` e `drop`.

3. **Esercizio 3**

Definire una funzione ricorsiva `rimuovi-duplicati: 'a list -> 'a list` che data una lista di elementi `l` restituisca la lista ottenuta da `l` sostituendo ogni sequenza di elementi uguali contigui in `l` con una sola occorrenza di tali elementi.

Per esempio:

`(rimuovi-duplicati [1;1;2;2;2;3;4])=[1;2;3;4]`.

ESAME DI PROGRAMMAZIONE 05/06/2013
(Programmazione Imperativa)

Cognome:

Nome:

Matricola:

Email:

Svolgere i seguenti quesiti nel foglio di protocollo. Consegnare: presente testo, bella e brutta copia. Tempo previsto: 90 minuti. Qualora lo si ritenesse necessario, è possibile scomporre l'esercizio in sottoproblemi risolti da corrispettive funzioni. In tal caso commentare lo scopo di tali funzioni.

1. **Esercizio 1 [13pt]**

Scrivere una funzione **ricorsiva** che, data una stringa contenente delle parole separate da uno o più caratteri di spaziatura ' ' restituisca una lista di stringhe in cui ciascuna cella memorizza una parola del testo in input.

- Dichiarare il tipo lista di stringa e descrivere brevemente la strategia di memorizzazione delle stringhe nelle celle.
- Definire la funzione ricorsiva.
- Scrivere un main di prova assumendo siano date le funzioni di inizializzazione delle strutture dati.

Non è possibile utilizzare alcuna funzione di libreria.

2. **Esercizio 2 [10pt]**

Scrivere una funzione che dato un array ordinato (in senso lato) di interi e la sua dimensione in input, allochi dinamicamente in memoria un array di interi in cui compare una sola volta ogni elemento dell'array in input. La funzione restituisce: un puntatore all'array allocato e la sua dimensione. L'array allocato deve avere la dimensione minima necessaria. Scrivere un main di prova assumendo siano date le funzione di inizializzazione delle strutture dati.

Esempio: Se l'array in input ha dimensione 10 e i suoi elementi sono:

5, 5, 9, 9, 9, 10, 14, 15, 15, 20

Allora l'array allocato dinamicamente sarà:

5, 9, 10, 14, 15, 20

e la sua dimensione sarà pari a 6.

3. **Esercizio 3 [10pt]**

Scrivere una funzione che decida se due liste di stringhe sono uguali. Per liste uguali intendiamo che le stringhe contenute in celle corrispondenti sono identiche carattere per carattere (anche se allocate in celle di memoria diverse). È possibile utilizzare le funzioni della libreria `string.h` qualora lo si ritenesse necessario.

6. Data la seguente funzione:

```
int* mia(int a) {  
    return &a;  
}
```

e la seguente chiamata:

```
int k1=8;  
int *pt;  
pt = mia(k1);
```

Quale delle seguenti è corretta?

- A. pt punta alla variabile k1 B. pt punta ad una locazione non associata ad una variabile
C. Il codice non compila a causa di un errore di tipo D. Il valore di pt è 8

6. _____

7. Date le dichiarazioni `float *a; float *b;` la scrittura `b-a` denota:

- A. Un'espressione di tipo `float*` ma non una variabile B. Un'espressione di tipo `float` ma non una variabile
C. Un'espressione di tipo `int` ma non una variabile D. Una variabile di tipo `float*`

7. _____

8. Il linguaggio C supporta il passaggio di parametri:

- A. per copia B. per indirizzo C. sia per copia che per indirizzo D. ne' per copia ne' per indirizzo

8. _____

9. Data la dichiarazione `double* miovevett[100];`, come puoi interpretare questa struttura dati?

- A. Un vettore di 100 celle `double` B. Un vettore di 100 puntatori a celle `double` C. Un puntatore ad un vettore di `double` D. Un puntatore ad un vettore di puntatori a `double`

9. _____

10. Date le seguenti dichiarazioni `float a[100]; int b=20;`, la scrittura `*(a+b)` è equivalente a:

- A. Non è corretta B. `&a[0]+b` C. `*a[b]` D. `a[b]`

10. _____