

## TEST DI PROGRAMMAZIONE - 18 Maggio 2016

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

Cognome e Nome: \_\_\_\_\_  
Matricola: \_\_\_\_\_

1. Data la seguente funzione:

```
let rec f p lst = match lst with
  [] -> []
  | x::xs -> if (p x) then x::(f p xs) else (f p xs);;
```

Qual'è il tipo della funzione ?

- A. 'a list -> 'a list -> 'a list    B. 'a list -> 'a list -> ('a -> bool)  
C. ('a -> bool)-> 'a list -> 'a list    D. 'a list -> 'a -> 'a list

1. \_\_\_\_\_

2. Data la seguente funzione g:

```
let g n = n mod 2 = 1;
dire che cosa restituisce.
```

- A. il valore di n mod 2    B. restituisce sempre false    C. true se n è un numero pari, false altrimenti  
D. true se n è un numero dispari, false altrimenti

2. \_\_\_\_\_

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

- A. f [1;2;4;5] [4]    B. f g [6;78;9;0;16]    C. f g ['a';'b';'z']  
D. f abs [-6;78;-9;0;16]    E. f 3 ['a';'b';'z']

3. \_\_\_\_\_

4. Dire quale delle seguenti espressioni ha come valore string list = [ "hello"; "world" ]:

- A. [ "hello" ] :: [ "world" ]    B. [ "hello" ] @ [ "world" ]    C. "hello" :: "world"  
D. "hello" @ "world"    E. "hello" @ [ "world" ]    F. [ "hello" ] @ "world"

4. \_\_\_\_\_

5. Sia tlista un puntatore ad una cella definita per una lista di interi. Data la seguente funzione:

```
int foo(tlista l) {
  if (l && l->next)
    return (l->info < l->next->info) ? 1+foo(l->next) : 0;
}
```

qual è il valore restituito dalla chiamata sulla lista 3-5-9-0-1-0?

- A. 2    B. 3    C. 5    D. nessuno dei valori precedentemente elencati

5. \_\_\_\_\_



6. Nel linguaggio C i parametri attuali sono sempre:

- A. Indirizzi di memoria B. Variabili C. Ambienti D. Espressioni

6. \_\_\_\_\_

7. Dato il vettore di stringhe `char* v1[10]` quale espressione tra le seguenti è equivalente a `v1[2][7]`?

- A. `*(v1+2+7)` B. `*(*(v1+2)+ 7)` C. `*(v1+2)+ 7` D. `*v1 + 7`

7. \_\_\_\_\_

8. Data la dichiarazione `char **p`, quali fra le seguenti affermazioni è errata?

- A. Il tipo di `*p` è `char*` e può comparire a sinistra di un assegnamento B. Il tipo di `*(p+7)` è `char*` e può comparire a sinistra di un assegnamento C. Il tipo di `**p +7` è `char` e può comparire a sinistra di un assegnamento D. Il tipo di `**(p +7)` è `char` e può comparire a sinistra di un assegnamento

8. \_\_\_\_\_

9. Sia `testa` una variabile di tipo `tlista` che punta ad una cella di una lista con almeno due elementi. Per quale ragione non possiamo dire che `testa+1` è l'indirizzo della cella successiva a quella puntata da `testa`?

- A. Perché `testa+1` è un'espressione non valida B. Perché `testa+1` pur essendo un'espressione valida non ha tipo `tlista` C. Perché le celle della lista nella memoria dinamica potrebbero essere allocate in modo non contiguo D. Perché `testa+1` incrementa il valore dell'informazione contenuta nella cella puntata da `testa`

9. \_\_\_\_\_

10. Data la seguente funzione:

```
void foo(int **vet) {  
    int i;  
    *vet = (int*) malloc(100*sizeof(int));  
    for (i=0; i<100; i++)  
        (*vet)[i]=0;  
}
```

e la chiamata

```
int main(){  
    int **vet;  
    foo(vet);  
}
```

Quale delle seguenti affermazioni è corretta?

- A. La chiamata causa un errore di tipo in compilazione B. La chiamata causa un errore di tipo in esecuzione C. Il codice è corretto e il chiamante potrà gestire il vettore allocato da `foo` D. Non ci sono errori di tipo ma la funzione `foo` causa del garbage in memoria

10. \_\_\_\_\_