



CORSO DI PROGRAMMAZIONE: SECONDA PROVA DI ACCERTAMENTO

COGNOME	NOME	MATRICOLA
----------------	-------------	------------------

1. Che cos'è una variabile riferimento (puntatore)?

RISPOSTA:

2. A quali espressioni si applica l'operatore unario di dereferenziazione *, e cosa restituisce?

RISPOSTA:

3. La seguente espressione di inizializzazione è corretta? Perché?

```
char *p = &((char*) p);
```

RISPOSTA:

4. Se la variabile `tab` è dichiarata come `char* arr[] = {"p", "a", "n", "i", "c", "o"};`

- L'espressione `arr[3]` è un designatore?
- qual è il suo tipo?
- qual è il suo valore?

RISPOSTE:

- -
 -
-

5. Come si può verificare se due variabili di tipo stringa rappresentano lo stesso valore?

RISPOSTA:

6. E' possibile usare l'aritmetica dei puntatori sulle liste? Perché?

RISPOSTA:

-
7. Dichiarare un tipo enumerazione `Colore` che rappresenta i 2 colori delle pedine degli scacchi (bianco e nero), ed un tipo enumerazione `Personaggio` che rappresenta i diversi tipi di pedina (re, regina, alfiere, cavallo, torre e pedone).

RISPOSTA:

8. Dichiarare un tipo record `Pedina`, che rappresenta una pedina degli scacchi, con due campi: `colore` e `personaggio`, di tipo `Colore` e `Personaggio`, rispettivamente.

RISPOSTA:

9. Dichiarare un tipo record `Casella` che rappresenta una casella della scacchiera, ed è un record di due campi: `colore` (il colore della casella, di tipo `Colore`) e `pezzo`, un riferimento ad una variabile di tipo `Pedina` (il riferimento sarà `NULL` se la casella non contiene nessuna pedina, altrimenti conterrà l'indirizzo di una variabile pedina).
Dichiarare infine il tipo `Scacchiera`, i cui elementi sono tabelle bidimensionali di dimensione 8 per 8 aventi come di tipo base il tipo `Casella`. Un valore di tipo `scacchiera` rappresenta una possibile configurazione della scacchiera.

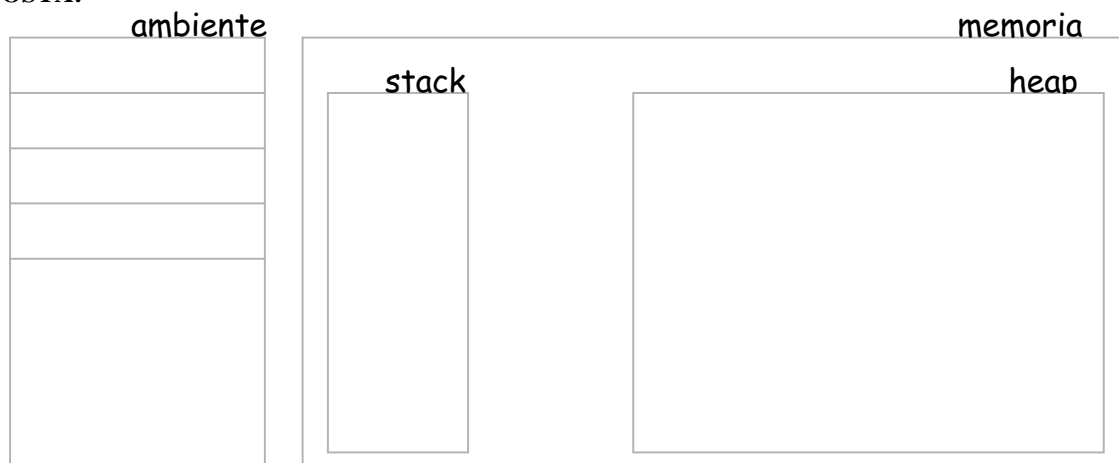
RISPOSTA:

10. Utilizzando i tipi definiti precedentemente, inizializzare una variabile `s` di tipo `Scacchiera` in modo che rappresenti una scacchiera in cui tutte le caselle sono nere e nella quale è presente solo il re bianco (in una cella qualsiasi).

RISPOSTA:

11. Utilizzando il modello ambiente-memoria, descrivere graficamente la situazione al termine dei comandi necessari a realizzare quanto richiesto dall'esercizio precedente.

RISPOSTA:



-
12. Utilizzando i tipi definiti negli esercizi precedenti, scrivere la definizione di una funzione che, prendendo come parametro attuale un valore `s` di tipo `Scacchiera` verifica se la torre nera è presente nella scacchiera `s`. La funzione restituisce 1 se la torre nera è presente, altrimenti restituisce 0.

RISPOSTA:

-
13. E' vero o no che ogni variabile (statica o dinamica) ha un indirizzo che la identifica univocamente?

RISPOSTA:

-
14. Cosa restituisce l'espressione `(double*) malloc(sizeof(double))` ?

RISPOSTA:

-
15. Si consideri `int *x`; Per quali valori di `x` l'espressione booleana `!((x==NULL) && (x==&y))` è falsa?

RISPOSTA:

-
16. Qual è il valore delle variabili `*v`, `w[0]` e `w[1]` dopo l'esecuzione del comando seguente?

```
int *v, w[2];
w[1] = 1;
w[0] = 0;
v = w;
*v++ = 2;
```

RISPOSTA:

17. In cosa possono differire la dichiarazione e la definizione di una stessa procedura?

RISPOSTA:

18. Si considerino le seguenti definizioni di costante di procedura:

```
char sempre_a(char x){ return 'a'; }
char sempre_b(char x){ return 'b'; }
```

Si definisca il tipo `Sempre_k` al quale appartengono sia `sempre_a` che `sempre_b`.

RISPOSTA:

19. Utilizzando il tipo `Sempre_k` dell'esercizio precedente, scrivere i comandi necessari per dichiarare due variabili di tipo `Sempre_k`, ed assegnare loro, rispettivamente, la costante di procedura `sempre_a` e `sempre_b` rispettivamente.

RISPOSTA:

20. Per quali valori del parametri attuale la seguente procedura termina correttamente, assumendo che `Lista` sia un tipo lista di interi con campi `val` e `next`?

```
int foo(Lista *lista){
    int i;
    if ((lista != NULL) && (lista->val != 0)) {
        i = lista->val;
        while (i != 0)
            i = foo(lista->next);
        return i;
    }
    else
        return 1;
}
```

RISPOSTA:
