

FONDAMENTI DI PROGRAMMAZIONE A*Tempo a disposizione: 20 minuti*

Nome Cognome Matricola

Per accedere alla prova di programmazione è necessario rispondere correttamente ad almeno il 70% delle domande

1. Si consideri il seguente frammento di codice.

```
int a = 52;
int* b = &a;
```

Si indichi la risposta corretta

- a b contiene l'indirizzo di memoria della variabile a
- b il frammento di codice ritorna un errore a tempo di compilazione
- c b è un riferimento alla variabile a
- d il frammento di codice compila ma ritorna un errore a tempo di esecuzione
- e nessuna delle precedenti

2. Supponendo che le espressioni vengano valutate da sinistra verso destra, cosa stampa a monitor il seguente frammento di codice?

```
int w = 3, z = 2;
bool b = (--w == z) || (z++ != 2);
cout << b << " " << w << " " << z << endl;
```

- a 0 2 2
- b 0 2 3
- c 1 3 3
- d 1 2 2
- e nessuna delle precedenti

3. Si consideri la seguente dichiarazione: `char s[] = {'a', '\0', 'c', '\0', 'd'};`. La variabile s corrisponde alla stringa C-style

- a "" (stringa vuota)
- b "a"
- c "ac"
- d non è una stringa C-style
- e nessuna delle precedenti

4. Cosa stampa il seguente programma?

```
for (int i = 4; i > 0; i--) {
    if (i > 3)
        continue;
    cout << i;
}
```

- a 2 1
- b 3 2 1
- c 4 3 2 1
- d 4 3 2 1 0
- e nessuna delle precedenti

5. Gli array prevedono un accesso di tipo

- a LIFO (*Last In First Out*)
- b FIFO (*First In First Out*)
- c diretto
- d sequenziale
- e nessuna delle precedenti

6. Sia p un puntatore a interi. L'espressione `(*p)++`

- a ritorna un errore a tempo di compilazione
- b incrementa il valore della variabile puntata da p
- c incrementa l'indirizzo di memoria contenuto in p
- d compila ma ritorna un errore a tempo di esecuzione
- e nessuna delle precedenti

7. Si consideri il seguente tipo struct

```
struct Point {  
    int x;  
    int y;  
}
```

e la seguente dichiarazione: `Point p = {1, 2};`. L'espressione corretta per accedere al campo x della variabile p è

- a `(*p).x`
- b `p->x`
- c `p[x]`
- d `p.x`
- e nessuna delle precedenti

8. Si indichi quale operatore viene utilizzato per referenziare una variabile

- a `.`
- b `*`
- c `&`
- d `->`
- e nessuna delle precedenti

9. Dato un puntatore a interi ptr, è possibile eseguire l'operazione `*ptr + 5`.

T F

10. In C++, se una zona di memoria allocata nello heap non è puntata da nessun puntatore, è considerata *garbage* e viene deallocated automaticamente.

T F