

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 = 1, z = 2;  
bool b = ++w == z++ || ++w == --z + w;  
cout << b << " " << w << " " << z << endl;
```

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

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

- ☐ a "" (stringa vuota)
☒ b "abc"
☐ c "abcd"
☐ 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 LIFO (*Last In First Out*)

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

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

7. Si consideri il seguente tipo struct

```
struct M {  
    int c1;  
    int c2;  
}
```

e la seguente dichiarazione: `M* m = new M;`. L'espressione corretta per accedere al campo `c2` della variabile di tipo `M` puntata da `m` è

- ☐ `a` `(*m + 1)[c2]`
- ☒ `b` `m->c2`
- ☐ `c` `m[c2]`
- ☐ `d` `m.c2`
- ☐ `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 caratteri `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 deallocata automaticamente.

☐ T ☒ F