

FONDAMENTI DI PROGRAMMAZIONE A*Tempo a disposizione: 30 minuti*

Nome Cognome Matricola

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

1. Per quali valori di `x`, variabile di tipo `int`, l'espressione `x >= 0 && x <= 10` risulta **vera**?

- ☐ *a* qualsiasi valore intero
☐ *b* esclusivamente i valori compresi fra 1 (incluso) e 9 (incluso)
☐ *c* esclusivamente i valori compresi fra 0 (incluso) e 10 (incluso)
☐ *d* esclusivamente i valori minori strettamente di 10

2. Cosa stampa il seguente programma?

```
for (int k = 0; k <= 2; k++) {  
    if (k == 1) {  
        continue;  
    }  
    cout << k;  
}
```

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

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

```
int x = 1, y = 1;  
bool b = (++x == ++y) || (++x != y--);  
cout << b << " " << x << " " << y << endl;
```

- ☐ *a* 0 4 1
☐ *b* 1 1 1
☐ *c* 1 2 2
☐ *d* 0 3 1

4. In C++, il passaggio di parametri per riferimento è implementato esclusivamente tramite puntatori

☐ V ☐ F

5. Si consideri il seguente tipo struct

```
struct MyStruct {  
    int first;  
    int second;  
};
```

e la seguente dichiarazione: `MyStruct s = {1, 2}`. L'espressione corretta per accedere al campo `first` di `s` è

- ☐ *a* `(*s).first`
☐ *b* `s->first`
☐ *c* `s[first]`
☐ *d* `s.first`
☐ *e* nessuna delle precedenti

6. Cosa stampa il seguente frammento di codice?

```
int arr[5] = {5,4,6,3,1};
int i = 1;
cout << arr[i + 2] + 1;
```

- ☐ a 7
☐ b 4
☐ c 2
☐ d nessuna delle precedenti

7. Cosa stampa il seguente frammento di codice?

```
int a1[3] = {2,7,8};
int a2[3] = {3,5,9};
cout << *a1 + *a2;
```

- ☐ a 5
☐ b 12
☐ c 0
☐ d errore a tempo di compilazione
☐ e errore a tempo di esecuzione

8. Si consideri la seguente funzione

```
void f(int& a, int& b) {
    a = b;
    return;
}
```

Si indichi cosa stampa a video l'esecuzione delle seguenti istruzioni

```
int x = 3, y = 1;
f(x, y);
cout << x;
```

- ☐ a 1 ☐ b 3 ☐ c 4 ☐ d nessuna delle precedenti

9. Cosa stampa il seguente frammento di codice?

```
int x = 1, y = 1;
int* p0 = &x;
int* p1 = &y;
*p0 = *p1 + 1;
cout << x << "□" << y << endl;
```

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

10. Si consideri la seguente funzione ricorsiva

```
int dummy(int par) {
    if (par <= 0)
        return 1;
    else
        return par + dummy(par - 1);
}
```

La chiamata a funzione `dummy(1)` termina.

☐ V ☐ F