

**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. Si consideri il seguente frammento di codice.

```
float var1 = 5;
float& var2 = var1;
```

Si indichi la risposta corretta

- a var2 contiene l'indirizzo di memoria della variabile var1
- b il frammento di codice ritorna un errore a tempo di compilazione
- c var2 è un riferimento alla variabile var1
- 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[] = {'h', 'e', 'l', 'l', 'o'};`. La variabile s corrisponde alla stringa C-style "hello".

T	F
---	---

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

6. Cosa stampa il seguente programma?

```
int V[] = {5, 11, 20, 17, 8};  
int* p = &(V[1]);  
cout << *(p + 2);
```

- a 11  b 20  c 10  d 17  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` di `m` è

- a `(*m + 1)[c2]`  
 b `m->c2`  
 c `m[c2]`  
 d `m.c2`  
 e nessuna delle precedenti

8. Cosa stampa il seguente frammento di codice?

```
int a[] = {3, 7, 12, 5, -3};  
a = a + 3;  
cout << *a << endl;
```

- a 12  b 5  c -3  d nessuna delle precedenti

9. Si consideri la seguente funzione

```
int fact(int x) {  
    if (x == 0)  
        return 0;  
    else if (x == 1)  
        return 1;  
    else  
        return x * fact(x - 1);  
}
```

La chiamata a funzione `fact(-2)` ritorna 0

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