

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 x = 10;  
int* y = &x;  
*y = 20;
```

Si indichi la risposta corretta

- a x contiene il valore 10
- b x contiene il valore 20
- c y contiene il valore 20
- d il frammento di codice ritorna un errore a tempo di compilazione
- e il frammento di codice compila ma ritorna un errore a tempo di esecuzione

2. Si consideri il seguente frammento di codice.

```
int x = 5;  
int y = ++x;
```

Si indichi la risposta corretta.

- a x contiene il valore 5, y contiene il valore 5
- b x contiene il valore 6, y contiene il valore 5
- c x contiene il valore 6, y contiene il valore 6
- d x contiene il valore 5, y contiene il valore 6
- e nessuna delle precedenti

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

- a "" (stringa vuota)
- b "a"
- c "ab"
- d "abc"
- e non è una stringa C-style

4. Cosa stampa il seguente programma?

```
int arr[3] = {1, 2, 3};  
for (int i = 0; i < 3; i++) {  
    arr[i] = arr[i] + 1;  
}  
for (int i = 2; i >= 0; i--) {  
    cout << arr[i] << " ";
```

- a 1 2 3
- b 3 2 1
- c 4 3 2
- d ritorna un errore a tempo di compilazione
- e compila ma ritorna un errore a tempo di esecuzione

5. Si consideri il seguente tipo struct

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

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

- a `p.y`
- b `(*p).y`
- c `p->y`
- d `p[y]`
- 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 indichi l'espressione corretta per deallocare un array di interi `arr` dinamicamente allocato.

- a `delete arr`
- b `arr.delete()`
- c `delete [] arr`
- d `dealloc(arr)`
- e nessuna delle precedenti

8. Cosa stampa il seguente frammento di codice?

```
int x = 4, y = 4;  
int* p1 = &x;  
int* p2 = &y;  
*p1 = *p2 + 1;  
cout << x << " " << y << endl;
```

- a 4 4
- b 5 5
- c 5 4
- d 4 5
- e nessuna delle precedenti.

9. Dato un puntatore `ptr` di tipo `char*`, è possibile eseguire l'operazione `ptr - 2`.

T F

10. In C++, un array deve contenere valori dello stesso tipo.

T F