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] il frammento di codice ritorna un errore a tempo di compilazione
- [d] il frammento di codice compila ma ritorna un errore a tempo di esecuzione
- [e] nessuna delle precedenti

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 indichi l'espressione corretta per controllare se due stringhe C-style `s1` e `s2` sono uguali

- [a] `s1 == s2`
- [b] `s1.equals(s2)`
- [c] `strcmp(s1, s2) == 0`
- [d] `s1.cmp(s2) == 0`
- [e] nessuna delle precedenti

4. Cosa stampa il seguente programma?

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

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

5. Le liste 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] incrementa il valore della variabile puntata da `p`
- [b] incrementa l'indirizzo di memoria contenuto in `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 inidichi l'espressione corretta per deallocare un array `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 = 5, y = 5;
int* p1 = &x;
int* p2 = &y;
*p1 = *p2 +1;
cout << x << "␣" << y << endl;
```

[a] `5 5` [b] `6 6` [c] `6 5` [d] nessuna delle precedenti.

9. Dato un puntatore a interi `ptr`, è possibile eseguire l'operazione `ptr + 2`. [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]