

**Test d'ingresso di Programmazione ad Oggetti**  
**24/09/2018**  
**Tempo a disposizione: 1 ora**

**Domanda 1.** Si consideri la seguente funzione.

```
bool f(int v[], int lun, int x)
{
    for(int i = 0; i < lun; i++)
        if(v[i] == x)
            return true;
    else
        return false;
}
```

Indicare le affermazioni vere nel caso in cui la funzione sia invocata con  $v = [11, 21, 13, 4]$  e  $x=21$ .

**Risposta**

- |  |   |
|--|---|
| 1. La funzione restituisce true.           | 2. La funzione restituisce false.             |
| 3. La funzione entra in un ciclo infinito. | 4. La funzione accede a memoria non allocata. |

**Domanda 2.** Indicare l'output del seguente frammento di codice.

```
int a = 1, b = 2, x = 0, count = 0;
do{
    x += a / b;
    count++;
}while(x < 5);
cout << "-" << count << endl;
```

**Risposta**

- |                      |  |  |                 |
|----------------------|--|--|-----------------|
| 1. La stringa -0     | 2. La stringa 0                                      | 3. La stringa -5                                 | 4. La stringa 5 |
| 4. La stringa -10    | 6. La stringa 10                                     | 7. Nulla, in quanto è presente un ciclo infinito |                 |
| 8. La stringa -count | 9. Nulla, in quanto è presente un errore di sintassi |  |                 |

**Domanda 3.** Si consideri il seguente frammento di codice.

```
int* A = new int(5);
int* B = new int[5];
```

Indicare le affermazioni vere.

**Risposta**

- |  |   |
|--|---|
| 1. La variabile A punta a un intero inizializzato a 5                              | 2. La variabile B punta a un intero inizializzato a 5 |
| 3. La variabile A punta a un array di 5 elementi                                   | 4. La variabile B punta a un array di 5 elementi      |
| 5. La prima istruzione è sintatticamente sbagliata                                 | 6. La seconda istruzione è sintatticamente sbagliata  |
| 7. La prima istruzione sarebbe da evitare  | 8. La seconda istruzione sarebbe da evitare           |
| 9. Il C++ non consente di utilizzare entrambe le istruzioni nello stesso programma |   |

**Domanda 4.** Si consideri il seguente frammento di codice.

```
int* A = new int(5);
int* B = new int[5];
```

Indicare le affermazioni vere.

**Risposta**

- |  |
|--|
| 1. La memoria puntata da A viene deallocata automaticamente quando non è più necessaria.                       |
| 2. La memoria puntata da B viene deallocata automaticamente quando non è più necessaria.                       |
| 3. La memoria puntata da A deve essere deallocata con l'istruzione "delete A;" quando non è più necessaria.    |
| 4. La memoria puntata da B deve essere deallocata con l'istruzione "delete B;" quando non è più necessaria.    |
| 5. La memoria puntata da A deve essere deallocata con l'istruzione "delete [] A;" quando non è più necessaria. |
| 6. La memoria puntata da B deve essere deallocata con l'istruzione "delete [] B;" quando non è più necessaria. |
| 7. La memoria puntata da A deve essere deallocata con l'istruzione "delete A[0];" quando non è più necessaria. |
| 8. La memoria puntata da B deve essere deallocata con l'istruzione "delete B[0];" quando non è più necessaria. |
| 9. La prima istruzione contiene un errore.   |
| 10. La seconda istruzione contiene un errore.  |

**Test d'ingresso di Programmazione ad Oggetti**  
**24/09/2018**  
**Tempo a disposizione: 1 ora**

**Domanda 5.** Si consideri la seguente funzione.

```
void f(int x, int& y){
    if(x > 0){
        x -= 1;
        f(x, y);
    }
    else {
        x = y;
        y--;
    }
}
```

Indicare l'output del seguente frammento di codice.

```
int x = 5, y = 6;
f(x, y);
cout << x << "-" << y << endl;
```

**Risposta**

1. La stringa 5-5   2. La stringa 0-6   3. La stringa 5-0   4. E' impossibile determinarlo  
5. La stringa x-y   6. Sempre e comunque un numero negativo, perché  $x < y$   
7. La stringa 5-6   8. Nulla, in quanto la funzione  $f$  è ricorsiva e richiama se stessa all'infinito.

**Domanda 6.** Si consideri la seguente dichiarazione **char\* X[20]**; Si indichino le istruzioni di allocazione necessarie affinché  $X$  sia una matrice dinamica di dimensione  $20 \times 10$ , e le corrispondenti istruzioni di deallocazione.

**Domanda 7.** Si consideri il seguente programma e si indichi cosa viene stampato in output.

```
int main() {
    unsigned s[20], *p=s+20, i=0;
    while(p!=s){
        p--;
        *p=++i;
    }
    for(unsigned i=0; i<20; i++)
        cout<<s[i]<<endl;
    return 0;
}
```

**Domanda 8.** Si considerino le seguenti istruzioni:

- 1) `int * p1 = new int(20);`   2) `short *p2 = new short[20];`   3) `char ** p3 = new char*[20];`  
4) `unsigned * p4[20];`   5) `bool * p5[20] = new bool [20];`   6) `double ** p6 =new double[20][20];`

Per ognuna di esse indicare se è corretta, se necessita di una corrispondente operazione di delete, e in caso affermativo, specificarla.

- 1)  
2)  
3)  
4)  
5)  
6)

**Domanda 9.** Si implementi una classe Corso che consenta di modellare un corso universitario. I dati della classe devono essere almeno il nome del corso, il semestre in cui tiene, l'elenco dei numeri di matricola degli studenti che frequentano il corso. La classe deve contenere oltre a costruttori/metodi/operatori che si ritengono utili:

- metodi per inserire e rimuovere uno studente nell'elenco
- Un metodo per ordinare l'elenco per matricole crescenti
- Operatori `<<`, `>>` e `[]`