

✗ **Non corretto** 0/1 Punti

13

Si consideri il seguente programma C,
quale tra le affermazioni elencate è vera?

```
char *f(char *x, int k){  
    return &(x[k]);  
}  
  
int main() {  
    char a[] = "programming";  
    char *x = f(a, 3);  
}
```



- ☒ `x < a`
- ☐ `*a > *x`
- ☐ `a < &x`
- ☐ `&a < x`
- ☐ `&a < &x`



Questo contenuto è creato dal proprietario del modulo. I dati inoltrati verranno inviati al proprietario del modulo. Microsoft non è responsabile per la privacy o le procedure di sicurezza dei propri clienti, incluse quelle del proprietario di questo modulo. Non fornire

☐ indefinito☐ 10☐ 0☐ 9

✗ Non corretto 0/1 Punti

12

Siano i e n interi positivi con $i < n$,
quanto vale $a[i]$ al termine del seguente
frammento di codice?

```
int i, *a;  
  
for (i = 0; i < n; i++){  
    if (i%10 == 0){  
        a = malloc((i+1)*10*sizeof(int));  
    }  
    a[i] = i;  
}
```

☐ altro☐ 0☐ i solo se $n < 10$ ☐ indefinito sempre



☐ $O(\min(n^2, n \log n))$

☐ $O(n)$

✓ **Esatto** 1/1 Punti

11

Qual è il valore della variabile n al termine del seguente frammento di codice C?

```
int n, a = 9;  
int *b = &a;  
a++;  
n = *b+1;;
```



☒ 11



☐ indefinito

☐ 10

☐ 0

☐ 9

✗ **Non corretto** 0/1 Punti

12

Siano i e n interi positivi con $i < n$.

✓ **Esatto** 1/1 Punti

10

Siano `merge_sort()` e `bubble_sort()` due funzioni Python che implementano nel modo più ottimizzato possibile i due algoritmi omonimi. Sia inoltre `a` una lista di `n` numeri, qual è la complessità temporale della seguente funzione?

```
def rsort(a):  
    a = merge_sort(a)  
    a = bubble_sort(a+a)
```



- ☐ $O(n \cdot n \cdot \log n)$
- ☒ $O(n^2)$
- ☐ $O(n \cdot \log n)$
- ☐ $O(n^{1.5})$
- ☐ $O(\min(n^2, n \cdot \log n))$
- ☐ $O(n)$

✓ **Esatto** 1/1 Punti

☐ 4

✗ **Non corretto** 0/1 Punti

9

Sia `a` una lista di interi e siano `x` e `y` due suoi elementi, quale tra le seguenti affermazioni è sempre vera dopo l'esecuzione del seguente codice?

```
b = sorted(a, key=lambda x: ( x*x, 0 if x < 0 else 1 ) )
```



- ☒ `x < 0` e `y > 0` implica `x` precede `y` in `b`
- ☐ `y <= x` implica `x` precede `y` in `b`
- ☐ `x < 0` e `y > 0` implica `y` precede `x` in `b`
- ☐ nessuna
- ☐ `x <= y` implica `x` precede `y` in `b`



✓ **Esatto** 1/1 Punti

10

Siano `merge_sort()` e `bubble_sort()` due funzioni Python che implementano nel modo più ottimizzato possibile i due



✗ Non corretto 0/1 Punti

8

Qual è il valore di `b` alla fine del seguente frammento di codice?

```
a = [4, 3, 2, 1]

for x in a:
    try:
        b += x
    except NameError:
        b = 1
```



☐ NameError

☐ 7



☐ 1

☐ 6

☒ 10

☐ 4

✗ Non corretto 0/1 Punti

9



peggiore

✓ **Esatto** 1/1 Punti

7

Assumendo che un puntatore occupi 8 byte ed un int 4 byte, quale delle seguenti affermazioni è vera dopo l'esecuzione del seguente frammento di codice?

```
int *a = malloc(2*sizeof(a));  
int b[8] = {0};  
  
int size_a = sizeof(a);  
int size_b = sizeof(b);
```



- ☐ `size_b == 2*size_a`
- ☒ `size_b == 4*size_a`
- ☐ `size_b == size_a`
- ☐ `size_b == 4`



✗ **Non corretto** 0/1 Punti

8

Qual è il valore di `b` alla fine del seguente frammento di codice?



✓ **Esatto** 1/1 Punti

6

Sia n un intero positivo maggiore di 1024, qual è la complessità temporale dell'ultima riga del seguente frammento di codice C?

```
int i;  
float *a = malloc(n*sizeof(float));  
for (i = 0; i < n; i++)  
    a[i] = i;  
a = realloc(a, n*n);
```



- ☒ lineare nel caso peggiore ✓
- ☐ costante nel caso medio
- ☐ costante nel caso peggiore
- ☐ quadratico nel caso peggiore
- ☐ lineare nel caso medio, quadratico nel caso peggiore

✓ **Esatto** 1/1 Punti

7

Assumendo che un puntatore occupi 8 byte ed un int 4 byte, quale delle seguenti

✓ **Esatto** 1/1 Punti

5

Cosa ritorna `enigma('01234567')`?

```
def enigma(a):  
    if len(a) <= 2:  
        return a  
    else:  
        return enigma(a[len(a)//2:]) +  
               enigma(a[:len(a)//2])
```

☐ 76540123☐ 01234567☐ 76543210☒ 67452301☐ 10325476✓ **Esatto** 1/1 Punti

6

Sia n un intero positivo maggiore di 1024, qual è la complessità temporale dell'ultima riga del seguente frammento di codice C?



✗ **Non corretto** 0/1 Punti

4

Si consideri la seguente operazione di assegnazione

$x_0, x_1, \dots = y_0, y_1, \dots$

dove a sinistra dell'uguale compaiono k valori e a destra h variabili. In quale caso questo tipo di operazione è legale?

- ☒ se k è uguale a h oppure a 1
- ☐ se k è minore o uguale a h
- ☐ Mai
- ☐ solo se k è uguale a 1
- ☐ solo se k è uguale a h
- ☐ se k è maggiore o uguale a h



✓ **Esatto** 1/1 Punti

5