

(1 punto)

Si consideri la seguente funzione C:

```
void f(int *x, int p){  
    *(a+p) = 2*p;  
}
```

Sia  $a$  un array di  $n > 0$  interi tale che  $a[i] = i$  per ogni indice  $i$  e sia  $0 \leq k < n$ . Dopo l'invocazione di  $f(a, k)$  qual è il valore di  $a[k]$ ?

☐ indefinito

☐  $2k$

☐  $k$

(1 punto)

Qual è il costo computazionale della seguente funzione C?

```
void f(int n){  
    int i, j;  
    for(i = 0; i < n; i++){  
        j = i;  
        while( j < n ){  
            j++;  
        }  
        while( j > 0 ){  
            j--;  
        }  
    }  
}
```

☐  $n^2$ ☐  $n$ ☐  $n^3$

(1 punto)

Sia  $a$  una lista concatenata contenente  $n$  interi ordinati in modo crescente, qual è il costo computazionale della ricerca di una chiave  $k$  in  $a$ ?

- ☐ lineare
- ☐ costante
- ☐ ordine di  $\log(n)$

(1 punto)

`BubbleSort()` è una funzione Python che implementa la versione più ottimizzata dell'algoritmo *bubble-sort*: prende in input una lista di interi e la modifica ordinando i suoi elementi in modo crescente. Sia  $a$  una lista di  $n$  interi, qual è il costo computazionale (caso peggiore) della seguente funzione?

```
def MultSort(a, n):  
    for i in range(n):  
        BubbleSort(a)
```

- ☐ Quadratico
- ☐ Lineare
- ☐ Cubico

(1 punto)

Cosa viene stampato dal seguente codice Python?

```
a = ['zero', 'one', 'two', 'three', 'four', 'five',\
     'six', 'seven', 'eight', 'nine', 'ten' ]
d = {}
for i in range(len(a)):
    d[a[i]] = i

print( a[d['one']] + 3 + d[a[4]] )
```

☐ Un errore

☐ eight

☐ 8

(1 punto)

Sia  $n$  un intero positivo e  $0 \leq k < n$ , qual è il valore di  $\text{len}(a[k])$ ?

```
a = ['x']  
for i in range(n):  
    a.append('x'+a[-1])
```

☐  $k-1$

☐  $k+1$

☐  $k$

(1 punto)

Sia  $n$  un intero positivo ed  $f$  definita come segue, qual è il risultato di  $f(n)$ ?

```
def f(n):  
    a = list(range(n))  
    a.append(list(range(n, 2*n)))  
    a += list(range(2*n, 3*n))  
    return a[n]*a[n+1]
```

- ☐ una lista
- ☐ un intero
- ☐ un errore

(1 punto)

Sia  $a$  una stringa qual è il risultato di `enigma(a)`?

```
def enigma(x):  
    if x == '':  
        return 1  
    else:  
        return 1+enigma(x[1:])
```

- ☐ `len(a)+1`
- ☐ una stringa formata da `len(a)` '1'
- ☐ `len(a)`



(1 punto)

Sia  $a$  una stringa di lunghezza  $n > 1$ , cosa viene stampato dal seguente frammento di codice C?

```
int n = strlen(a);  
char *b = a+n/2;  
*b = '\\0';  
printf("%ld\\n", strlen(a)-strlen(b));
```

- ☐ un numero  $> 0$
- ☐ un numero  $< 0$
- ☐ 0

(1 punto)

$a$  e  $b$  sono due liste concatenate contenenti interi e  $d$  un dizionario, inizialmente vuoto, implementato con liste di trabocco. Gli elementi di  $d$  sono coppie  $(k, v)$  dove la chiave  $k$  è di tipo intero e la chiave  $v$  è di tipo puntatore. Vengono eseguite le seguenti operazioni:

- per ogni elemento  $x$  di  $a$ , la coppia  $(x, \text{NULL})$  viene inserita in  $d$ ;
- per ogni elemento  $x$  di  $b$ , la coppia  $(x, \text{NULL})$  viene inserita in  $d$ .

Se  $a$  contiene  $n$  elementi e  $b$  ne contiene  $m$ , quanti elementi contiene  $d$ ?

- ☐ almeno  $n+m$
- ☐  $\max(n,m)$
- ☐ meno di  $n+m+1$