

1

(1 punto)

Sia `a` una lista di `n` dizionari ciascuno contenente `c` elementi a valori `None`. Si consideri il seguente codice:

```
d = {}  
  
for i in range(len(a)):  
    d = unione(d, a[i])
```



La funzione `unione` ritorna un dizionario che contiene l'unione delle chiavi dei dizionari in input.

Assumendo che `c` sia una costante, qual è la complessità temporale del precedente codice?

- ☐ $O(n^2 \log n)$
- ☐ $O(n)$
- ☐ $O(2^n)$
- ☐ $O(n \log n)$
- ☐ $O(n^2)$
- ☐ $O(n^3)$

2



forms.office.com



2

(1 punto)

Sia n un intero positivo maggiore di 1876. si consideri il valore di c al termine del seguente frammento di codice, quale tra le affermazioni elencate (espresse in codice C) è vera?

```
int c = n;

for (int i = 0; i < n; i++){
    c -= 1;
    i += 1;
}
```

- ☐ $c == n/2$
- ☐ $c == (n/2 - 1)$
- ☐ $c == (n/2 + 1)$
- ☐ $c == 0$
- ☐ $c == n$

3

(1 punto)

Cosa succede eseguendo il seguente programma?



forms.office.com



3

(1 punto)

Cosa succede eseguendo il seguente programma?

```
#include <stdio.h>
#include <string.h>

struct str {
    char *a;
    int n;
};
typedef struct str str;

str new_str(){
    str s;
    char x[] = "malloc";
    s.a = x;
    s.n = strlen(x);
    return s;
}

int main(){
    str b = new_str();
    printf("%s\n", b.a);
    return 6;
}
```

- ☐ Un ciclo infinito
- ☐ Errore in compilazione
- ☐ Qualcosa di indefinito
- ☐ Stampa la stringa malloc
- ☐ Stampa l'intero 6



4

Si consideri la seguente funzione dove **a** è una stringa. Quali delle seguenti affermazioni è vera? (1 punto)

```
def f(a):  
    b = ''  
    for x in a:  
        b = x+b  
    return b
```

- ☐ **f** calcola **a[:]** con lo stesso costo computazionale
- ☐ **f** calcola **a[::-1]** con lo stesso costo computazionale
- ☐ **f** calcola **a[::-1]** con un costo computazionale maggiore
- ☐ **f** calcola **a[:]** con un costo computazionale maggiore

5



forms.office.com



5

(1 punto)

Cosa succede compilando il seguente codice?

```
#include <stdio.h>

int main(){
    printf("%f\n", f(7));
}

float f(n){
    return 7/2.0;
}
```

- ☐ La compilazione viene interrotta riportando un errore
- ☐ La compilazione termina con successo ma il codice contiene un errore semantico
- ☐ La compilazione termina con successo
- ☐ La compilazione termina con successo riportando un warning
- ☐ La compilazione non termina, deve essere interrotta manualmente



6

Siano **n** e **m** due interi, come si può descrivere in maniera concisa il valore stampato dal seguente frammento di codice? (1 punto)

```
i = 0
while i < n or i < m:
    i+=1
print(i)
```

- ☐ min(n, m)
- ☐ altro
- ☐ 0
- ☐ max(n, m)
- ☐ min(0, n, m)

7



forms.office.com



☐ $\min(0, n, m)$

7

Si consideri una lista a contenente n interi ordinati dal più piccolo al più grande. Si definisca b come $b = [e] + a$. Qual è il costo temporale del bubble-sort sulla lista b ? (1 punto)

☐ $O(n^2)$

☐ $O(1)$

☐ $O(n \log n)$

☐ $O((n+1)^2)$

☐ $O(n)$

8

Sia a una stringa, cosa stampa `print(a < 2*a)`? (1 punto)



forms.office.com



☐ $O(n)$

8

Sia **a** una stringa, cosa stampa
print(a<2*a)? (1 punto)

☐ qualche volta **False**

☐ sempre **True**

☐ 0

☐ sempre **False**

9

(1 punto)

Si consideri il seguente codice

```
int main(){  
    char a[100] = {11};  
    char *p = &(a[50]);  
}
```



Qual è il valore di `p`?



forms.office.com



☐ sempre **False**

9

(1 punto)

Si consideri il seguente codice

```
int main(){  
    char a[100] = {11};  
    char *p = &a[50];  
}
```



Qual è il valore di `p` ?

- ☐ `a+50`
- ☐ indefinito
- ☐ 11
- ☐ 0
- ☐ 50

10

Sia `a` una lista di interi, in quale casi tra quelli elencati la seguente espressione



forms.office.com



☐ 50

10

Sia **a** una lista di interi, in quale casi tra quelli elencati la seguente espressione viene valutata **True** ?

$2 * \text{sorted}(a) \leq a$
(1 punto)

- ☐ Solo se **a** è ordinata
- ☐ Mai
- ☐ Solo se **a** è vuota
- ☐ Se **a** è non ordinata
- ☐ Sempre

Invia

 Microsoft 365

forms.office.com

