

Corso: Fondamenti di Informatica 1 (gruppo 8-9)

Corsi di laurea: Area dell'Informazione

Questionario di Autovalutazione

Laboratorio 5

Domanda n. 1

Argomento: Logica booleana

Peso: 1

Pubblica: N

Domanda: Negando la seguente espressione logica in Java mediante la legge di De Morgan si ottiene:

$$(a \mid\mid !b) \ \&\& \ (!a \mid\mid b)$$

Risposte:

1. $(a \parallel !b) \parallel (!a \parallel b)$

Punteggio: 0

2. $(!a \ \&\& b) \ \&\& (a \ \&\& !b)$

Punteggio: 0

3. $(!a \ \&\& b) \parallel (a \ \&\& !b)$

Punteggio: 1

4. $(!a \parallel b) \ \&\& (a \parallel !b)$

Punteggio: 0

5. nessuna delle precedenti risposte e' corretta

Punteggio: 0

Spiegazione:

Applicare le leggi di De Morgan:

Domanda n. 2

Argomento: Iterazioni, array e stringhe

Peso: 1

Pubblica: N

Domanda: Il seguente frammento di codice in Java:

```
int n = 4;
int p = 0;
int k = n;
while (k > 0)
{
    p += k;
    k--;
}
System.out.println("risultato: " + p);
```

Risposte:

1. compila, ma in esecuzione ciclo all'infinito

Punteggio: 0

2. compila e in esecuzione invia a standard output il numero n

Punteggio: 0

3. compila e in esecuzione invia a standard output la somma dei numeri da 1 a n

Punteggio: 1

4. non compila: il ciclo while deve essere sostituito con un ciclo for

Punteggio: 0

5. nessuna delle precedenti affermazioni e' corretta

Punteggio: 0

Spiegazione:

Eseguire il ciclo.

Domanda n. 3

Argomento: Iterazioni, array e stringhe

Peso: 1

Pubblica: N

Domanda: Il seguente frammento di codice in Java:

```
int n = 4;
int p = 1;
int k = n;
while (k < 0)
{
    p *= k;
    k--;
}
System.out.println("risultato: " + p);
```

Risposte:

1. compila, ma in esecuzione ciclo all'infinito

Punteggio: 0

2. compila e in esecuzione invia a standard output il numero n!

Punteggio: 0

3. compila e in esecuzione invia a standard output la somma dei numeri da 1 a n

Punteggio: 0

4. non compila; il ciclo while deve essere sostituito con un ciclo for

Punteggio: 0

5. nessuna delle precedenti affermazioni e' corretta

Punteggio: 1

Spiegazione:

Il frammento di codice invia a standard output il valore p = 1. Il corpo del ciclo non viene mai eseguito.

Domanda n. 4

Argomento: Variabili e tipi

Peso: 1

Pubblica: N

Domanda: Qual e' il tipo della seguente variabile a in linguaggio Java?

```
String[] a = new String[10];
```

Risposte:

1. e' un riferimento a un array di caratteri

Punteggio: 0

2. e' un riferimento a una stringa

Punteggio: 0

3. e' un riferimento a un array di stringhe

Punteggio: 1

4. nessuna delle precedenti

Punteggio: 0

Spiegazione:

Sintassi java: e' un riferimento a un array di stringhe.

Domanda n. 5

Argomento: Variabili e tipi

Peso: 1

Pubblica: N

Domanda: Qual e' il risultato dell'esecuzione della seguente frammento di codice?

```
public static void main(String[] args)
{   int m = f(7);
    System.out.println(m);
}

private static int f(int k) {
    int s = 0;
    int p = 1;
```

```

        while (k > 0) {
            s += k;
            p *= k;
            decremento(k);
        }
        return p;
    }
private static void decremento(int n) {n--;}

```

Risposte:

1. invia a standard output il numero 7!

Punteggio: 0

2. invia a standard output il numero $21 = 7 + (7 - 1) + \dots 2 + 1$

Punteggio: 0

3. invia a standard output il numero 0

Punteggio: 0

4. Il programma non si arresta per la presenza di un ciclo infinito

Punteggio: 1

5. nessuna delle precedenti

Punteggio: 0

Spiegazione:

il metodo decremento() non e' in grado di modificare il parametro effettivo.

Domanda n. 6

Argomento: Iterazioni, array e stringhe

Peso: 1

Pubblica: N

Domanda: Quale e' il risultato dell'esecuzione del seguente metodo in Java?

```

public static int fattoriale(int n)
{
    int p = 1;
    while (n > 0)
    {
        p = p * n;
        n = decrementa(n);
    }
    return p;
}

```

```
}

private static int decrementa(int k)
{ return k - 1; }
```

Risposte:

1. il fattoriale di n se n > 0, altrimenti 1

Punteggio: 1

2. sempre 1, per qualsiasi valore di n

Punteggio: 0

3. il metodo non termina a causa di un ciclo infinito

Punteggio: 0

Spiegazione:

L'enunciato n = decrementa(n); e' in grado di decrementare la variabile n, quindi il ciclo risulta finito e calcola n! se n > 0, altrimenti viene stampato il valore di p = 1.

Domanda n. 7

Argomento: Iterazioni, array e stringhe

Peso: 1

Pubblica: N

Domanda: Che risultato produce sull'uscita standard l'esecuzione della classe P?

```
class P {
    private static void f(int i, int j) {
        i--;
        if (i * j == 0) return;
        else j--;
    }

    public static void main(String[] args) {
        int a = 3;
        int b = 5;
        while(a > 0) f(a,b);
        System.out.println(a + " " + b);
    }
}
```

Risposte:

1. la stampa dei valori 3 5

Punteggio: 0

2. la stampa dei valori 0 5

Punteggio: 0

3. la stampa dei valori 0 3

Punteggio: 0

4. Il programma non si arresta per la presenza di un ciclo infinito

Punteggio: 1

5. nessuna delle precedenti risposte e' corretta

Punteggio: 0

Spiegazione:

Il metodo f() non può modificare la variabile a, e quindi il ciclo while risulta infinito.

Domanda n. 8

Argomento: Iterazioni, array e stringhe

Peso: 1

Pubblica: N

Domanda: Qual e' il risultato del seguente frammento di codice?

```
String[] s = {"a", "b", "c" };
for (int i = 0; i <= s.length; i++)
    System.out.print(s[i] + " ");
```

Risposte:

1. compila correttamente e, in esecuzione, stampa a standard output la stringa "a b c" senza generare eccezioni

Punteggio: 0

2. compila correttamente e, in esecuzione, stampa a standard output la stringa "c b a" senza generare eccezioni

Punteggio: 0

3. compila correttamente e, in esecuzione, genera l'eccezione

java.lang.ArrayIndexOutOfBoundsException

Punteggio: 1

4. genera un errore in compilazione

Punteggio: 0

5. nessuna delle precedenti

Punteggio: 0

Spiegazione:

La condizione logica del ciclo for genera java.lang.IndexOutOfBoundsException per i = 3.

Domanda n. 9

Argomento: Iterazioni, array e stringhe

Peso: 1

Pubblica: N

Domanda: Qual e' il risultato del seguente frammento di codice?

```
int a[] = {1, 1, 2, 3, 5, 8, 13, 21};  
int i = a.length - 1;  
int target = 8;  
while (i >= 0) {  
    if (a[i] == target) break;  
    i--;  
}  
System.out.println("a[" + i + "] = " + target);
```

Risposte:

1. l'esecuzione del frammento non termina

Punteggio: 0

2. stampa la stringa "a[5] = 8"

Punteggio: 1

3. errore di compilazione

Punteggio: 0

4. stampa la stringa "a[4] = 8"

Punteggio: 0

5. nessuna delle precedenti risposte e' corretta

Punteggio: 0

Spiegazione:

Stampa la stringa "a[5] = 8". Eseguire il ciclo.

Domanda n. 10

Argomento: Iterazioni, array e stringhe

Peso: 1

Pubblica: N

Domanda: Il metodo definito nel seguito

```
public void f(A[] a) { ... }
```

Risposte:

1. accetta come parametro un array di riferimenti di tipo a

Punteggio: 0

2. accetta come parametro un riferimento di tipo A

Punteggio: 0

3. accetta come parametro un riferimento a un array di elementi di tipo A

Punteggio: 1

4. nessuna delle precedenti affermazioni e' corretta

Punteggio: 0

Spiegazione:

Si veda la sintassi di Java.

Domanda n. 11

Argomento: Iterazioni, array e stringhe

Peso: 1

Pubblica: N

Domanda: Il numero di operazioni (accessi a singoli elementi) necessarie per eliminare un elemento da un array riempito solo in parte in cui l'ordine sequenziale degli elementi non e' importante

Risposte:

1. dipende solo dalla posizione dell'elemento da eliminare

Punteggio: 0

2. non dipende dalla posizione dell'elemento da eliminare

Punteggio: 1

3. dipende dalla dimensione dell'array e dalla posizione dell'elemento da eliminare

Punteggio: 0

4. nessuna delle precedenti affermazioni e' corretta

Punteggio: 0

Spiegazione:

Richiede un accesso in lettura e uno in scrittura per accedere, rispettivamente, all'ultimo elemento valido dell'array e all'elemento da eliminare.

Domanda n. 12

Argomento: Iterazioni, array e stringhe

Peso: 1

Pubblica: N

Domanda: Il numero di operazioni (accessi a singoli elementi) necessarie per eliminare un elemento da un array riempito solo in parte in cui l'ordine sequenziale degli elementi e' importante

Risposte:

1. dipende dalla posizione dell'elemento da eliminare e dal numero di elementi nell'array

Punteggio: 1

2. non dipende dalla posizione dell'elemento da eliminare

Punteggio: 0

3. dipende solo dalla dimensione dell'array

Punteggio: 0

4. nessuna delle precedenti affermazioni e' corretta

Punteggio: 0

Spiegazione:

Richiede (size - 1 - k) spostamenti, dove size e' il numero di elementi validi inseriti nell'array e k e' l'indice dell'elemento da eliminare.

Domanda n. 13

Argomento: Logica booleana

Peso: 1

Pubblica: N

Domanda: I due enunciati seguenti, dove x e y sono variabili intere

```
int x;  
int y;  
...  
if (x / y > 1 && y != 1) ...  
if (y != 1 && x / y > 1) ...
```

Risposte:

1. sono equivalenti

Punteggio: 1

2. hanno un comportamento diverso quando y vale 0

Punteggio: 0

3. hanno un comportamento diverso quando x e' positivo

Punteggio: 0

4. Nessuna delle precedenti risposte e' corretta

Punteggio: 0

Spiegazione:

Sono equivalenti; quando y = 0 entrambi generano ArithmeticException.

Domanda n. 14

Argomento: Iterazioni, array e stringhe

Peso: 1

Pubblica: N

Domanda: Qual e' il risultato del seguente metodo:

```
private static int m(int[] a)
{
    int p = a[0];

    for (int i = 1; i < a.length; i++)
        if (a[i] < p)
            p = a[i];

    return p;
}
```

Risposte:

1. genera errore in compilazione

Punteggio: 0

2. genera l'eccezione `ArrayIndexOutOfBoundsException`

Punteggio: 0

3. restituisce il massimo fra gli elementi dell'array a

Punteggio: 0

4. nessuna delle precedenti

Punteggio: 1

Spiegazione:

Restituisce il minimo fra gli elementi dell'array a.

Domanda n. 15

Argomento: Iterazioni, array e stringhe

Peso: 1

Pubblica: N

Domanda: Il numero di accessi per trovare la prima occorrenza di un elemento in un

array

Risposte:

1. dipende dall'indice dell'elemento da trovare

Punteggio: 1

2. non dipende dall'indice dell'elemento da trovare

Punteggio: 0

3. e' sempre pari al numero di elementi dell'array

Punteggio: 0

4. nessuna delle precedenti risposte e' corretta

Punteggio: 0

Spiegazione:

E' pari a $k + 1$, dove k e' l'indice dell'elemento da ricercare, se questo e' presente nell'array.

Domanda n. 16

Argomento: Variabili e tipi

Peso: 1

Pubblica: N

Domanda: Utilizzando una variabile riferimento che contiene il valore null per invocare un metodo

Risposte:

1. si commette un errore di sintassi

Punteggio: 0

2. si commette un errore logico, segnalato durante l'esecuzione del programma con il lancio di un'eccezione

Punteggio: 1

3. non si commette alcun errore

Punteggio: 0

Spiegazione:

si commette un errore logico che causa in esecuzione la generazione dell'eccezione NullPointerException.

Domanda n. 17

Argomento: Variabili e tipi

Peso: 1

Pubblica: N

Domanda: Assegnando un riferimento null a una variabile di tipo "int"

Risposte:

1. si commette un errore di sintassi

Punteggio: 1

2. si commette un errore logico, segnalato durante l'esecuzione del programma

Punteggio: 0

3. non si commette alcun errore

Punteggio: 0

4. nessuna delle precedenti affermazioni e' corretta

Punteggio: 0

Spiegazione:

si commette un errore di sintassi.

Domanda n. 18

Argomento: Logica booleana

Peso: 1

Pubblica: N

Domanda: Quale delle espressioni logiche in Java sotto riportate e' equivalente alla seguente?

`! (a <= 0 || a >= 10)`

Risposte:

1. $a > 0 \parallel a < 10$

Punteggio: 0

2. $a \geq 0 \&\& a \leq 10$

Punteggio: 0

3. $a \geq 0 \parallel a \leq 10$

Punteggio: 0

4. $a < 0 \&\& a > 10$

Punteggio: 0

5. nessuna delle precedenti risposte e' corretta

Punteggio: 1

Spiegazione:

Si applichi la legge di De Morgan: la risposta corretta e' ($a > 0 \&\& a < 10$).

Domanda n. 19

Argomento: Variabili e tipi

Peso: 1

Pubblica: N

Domanda: Il seguente frammento di codice in Java

```
int k = 'A';
char c = (char) (k + 1);
System.out.println(c);
```

Risposte:

1. genera un errore in compilazione

Punteggio: 0

2. non genera errori in compilazione ma lancia un'eccezione in esecuzione

Punteggio: 0

3. non genera errori in compilazione e non lancia eccezioni in esecuzione

Punteggio: 1

Spiegazione:

esegue stampando il carattere successivo ad 'A' nella tabella Unicode, ovvero 'B'.

Domanda n. 20

Argomento: Iterazioni, array e stringhe

Peso: 1

Pubblica: N

Domanda: Per modificare la dimensione di un array dichiarato con l'istruzione int[] v = new int[10]

Risposte:

1. si assegna un valore alla sua variabile pubblica di esemplare v.length

Punteggio: 0

2. si invoca il suo metodo pubblico v.length(int x), fornendo la nuova dimensione come parametro esplicito

Punteggio: 0

3. non e' possibile cambiare le dimensioni di un array dopo la sua creazione

Punteggio: 1

4. nessuna delle precedenti affermazioni e' corretta

Punteggio: 0

Spiegazione:

non e' possibile modificare la dimensione di un array.

Domanda n. 21

Argomento: Iterazioni, array e stringhe

Peso: 1

Pubblica: N

Domanda: Il seguente frammento di codice

```
int[] a = new int[10];
int[] b = a;
```

Risposte:

1. costruisce due array di int indicati con i nomi "a" e "b"

Punteggio: 0

2. contiene un errore di sintassi

Punteggio: 0

3. contiene un errore logico che verra' segnalato durante l'esecuzione con il lancio di un'eccezione

Punteggio: 0

4. nessuna delle precedenti affermazioni e' corretta

Punteggio: 1

Spiegazione:

costruisce un array il cui riferimento e' contenuto nelle due variabili a e b.

Domanda n. 22

Argomento: Classi

Peso: 1

Pubblica: N

Domanda: In Java, che cosa indica la parola chiave "this" all'interno di un metodo?

Risposte:

1. l'utente che sta eseguendo il programma

Punteggio: 0

2. la classe di cui viene eseguito il metodo

Punteggio: 0

3. il parametro隐式的 del metodo

Punteggio: 1

4. il risultato calcolato dal metodo

Punteggio: 0

Spiegazione:

il parametro隐式.