

Programmazione I

Prova scritta - 22 giugno 2018 - 1h20min

NOTA: Nei programmi si trascuri ogni problema legato al tipo ed al valore di ritorno della funzione **main**, inoltre si sottintenda la presenza delle direttive

```
#include <iostream> / #include <fstream> / using namespace std ;
```

e non si prenda come un buon esempio la formattazione utilizzata (spesso compressa per motivi di spazio). Si interpreti “terminazione forzata”, come l'abbreviazione di “terminazione forzata del programma da parte del sistema operativo”. Infine, laddove si trovi l'affermazione che un programma o frammento di codice produce un certo risultato, è da intendersi che, in accordo alle regole del linguaggio, tale programma o frammento di codice produce quel risultato per qualsiasi esecuzione su qualsiasi macchina.

PARTE 1 – RISPOSTA SINGOLA - Ogni domanda ha una sola risposta VERA.

- Una risposta esatta fa acquisire il punteggio positivo riportato a fianco della domanda
- Una risposta errata fa perdere il punteggio negativo riportato a fianco della domanda
- Una risposta lasciata in bianco viene valutata 0

1. (3, -.5) Il seguente programma:

```
struct pers {char nome[25]; int anno;} ;
pers fun(pers p, const char *s)
{ strcpy(p.nome, s) ; p.anno = 2000; return p ; }
main()
{
  pers z = {"luigi", 1985}; pers y = fun(z, "marco") ;
  cout<<z.nome<<" "<<y.nome ;
  strcpy(z.nome, "anna") ; cout<<" "<<y.anno<<" "<<y.nome ;
}
```

- a) Stampa **marco marco 2000 anna**
- b) Stampa **luigi marco 1985 marco**
- c) Stampa **marco marco 2000 marco**
- d) Stampa **luigi marco 2000 marco**

2. (2, -.5) Data la seguente definizione:

```
char c = static_cast<char>(static_cast<int>('z')) ;
```

- a) si ha perdita di informazione in almeno una delle conversioni
- b) subito prima dell'assegnamento viene effettuata una conversione implicita
- c) se si tentasse di stampare il valore di **c** mediante **cout<<c**; si stamperebbe il codice ASCII del carattere 'z'
- d) nessuna delle altre risposte è corretta

3. (2, -.5) Dato il seguente frammento di codice:

```
int a[3] ; delete [] a; a[0] = 3;
```

- a) il frammento non contiene errori di gestione della memoria;
- b) l'assegnamento **a[0] = 3**; comporta un errore di accesso alla memoria perché la memoria dinamica occupata da **a** è stata deallocata;
- c) l'oggetto **a** non appartiene alla classe di memorizzazione a cui può essere applicato l'operatore **delete**
- d) nessuna delle altre affermazioni è vera.

4. (3, -.5) Data la seguente stringa rappresentata mediante un array

```
char s[] = "abcdef" ;
```

le istruzioni:

```
s[2] = s[5] = '\0' ;
```

- contengono uno o più errori di accesso alla memoria
- trasformano la stringa originaria in due stringhe di lunghezza 2
- non modificano la lunghezza della stringa
- nessuna delle altre risposte è vera

PARTE 2 – (POSSIBILI) RISPOSTE MULTIPLE - Ogni domanda può avere da una a quattro risposte CORRETTE.

- Ogni risposta esatta viene calcolata: +1
- Ogni risposta errata viene calcolata: -.5
- Una risposta lasciata in bianco viene calcolata: 0

5. Nel seguente programma:

```
1:int a = 1;
2:
3:fun()
4:{
5:    int i ;
6:    for (i = 0 ; i < 5 ; i++) {
7:        double a = i / 2.0 ;
8:        cout<<a * i ;
9:    }
10:    cout<<a + i ;
11:}
12:
13:main()
14:{
15:    double b = 1.4 ;
16:    fun() ;
17:    cout<<a + b<<endl ;
18:}
```

- la variabile **a** definita alla riga 1 ha tempo di vita pari all'intero programma;
 - la variabile **b** definita alla riga 15 non ha visibilità pari all'intero programma;
 - la variabile **a** definita alla riga 1 non è visibile alla riga 8;
 - la variabile **b** definita alla riga 15 ha tempo di vita pari all'intero programma.
6. Data l'istruzione **cin>>i** ; ove **i** è una variabile di tipo **int** correttamente definita:
- Se, all'esecuzione di tale istruzione, lo *stdin* già contiene una sequenza di caratteri che rappresenta un numero intero, il programma non si blocca in attesa di input, ma prova immediatamente a memorizzare tale numero nella variabile **i**
 - All'esecuzione di tale istruzione il programma si blocca sempre in attesa che l'utente invii dei caratteri sullo *stdin*
 - Se, all'esecuzione di tale istruzione, lo *stdin* contiene una sequenza di caratteri che rappresenta un numero intero, seguita da uno spazio ed infine da una sequenza di caratteri che non rappresenta alcun numero, il **cin** non va in stato di errore
 - Nel caso in cui tale istruzione sia eseguita con successo, ossia venga memorizzato correttamente un numero intero letto da *stdin* nella variabile **i**, lo *stdin* è svuotato di tutto il suo contenuto

7. Dato il seguente programma:

```
void scrivi(ofstream &f) { f<<"prova" ; f.close() ; }
main()
{
    ofstream of("dati.txt") ; scrivi(of) ;
    ifstream is("dati.txt") ; char buf[256] ; is>>buf ;
    cout<<buf ;
}
```

- a) il programma stampa **prova** su *stdout*
- b) quando si riapre il file *dati.txt* in lettura all'interno della funzione *main* non è garantito che il suo contenuto sia stato aggiornato
- c) quando si riapre il file *dati.txt* in lettura, quest'ultimo contiene una sequenza di byte, ciascuno dei quali contiene a sua volta il codice di uno dei caratteri precedentemente scritti nel file stesso mediante la funzione *scrivi*
- d) anche se il parametro formale *f* della funzione *scrivi* muore in uscita dalla funzione stessa, l'invocazione della funzione *f.close()* nella funzione *scrivi* è comunque necessaria per provocare l'aggiornamento del file *dati.txt*

PARTE 3 – DOMANDE APERTE

- **Una risposta esatta fa acquisire il punteggio positivo riportato a fianco della domanda**
- **Una risposta errata può eventualmente causare una penalità che dipende dalla gravità dell'errore**
- **Una risposta lasciata in bianco viene calcolata: 0**

8. (6 pt) Descrivere sintassi, semantica e vantaggi del tipo *booleano*, in non più di otto righe, più eventuali frammenti di codice o ulteriori righe scritte in una qualsiasi notazione (a supporto della descrizione della sintassi). Non verrà valutata la quantità ma la qualità di quello che si scrive, ed il tentativo di scrivere in modo estremamente fitto per aumentare la quantità, così come il superamento del numero massimo di righe, comporteranno una penalità.

9. (7 pt) Scrivere una funzione che prende in ingresso due stringhe, **a** e **b**, e, se vi è una occorrenza della stringa **b** all'interno della stringa **a**, allora elimina la prima occorrenza della stringa **b** dalla stringa **a**. Ad esempio, se alla funzione vengono passate le stringhe *"Piu' per piu'"* e *"meno"*, allora la funzione non modifica la stringa **a**. Se invece la stringa **b** è la stessa del precedente esempio, ma la stringa **a** è *"Piu' per meno meno"*, allora trasforma la prima stringa in *"Piu' per meno"*.

Programmazione I

Prova scritta - 22 giugno 2018

Nome: _____ Cognome: _____

Matricola: _____

Indicare le risposte corrette apponendo una croce nella casella corrispondente. Per superare la prova bisogna aver raggiunto almeno 9 punti nelle domande a risposta singola/multipla, ed almeno 15 complessivamente. Questa è l'unica pagina che dovete consegnare. Per comodità avete anche una copia di questa pagina per calcolare il voto da sole/soli durante la correzione.

	Risposte				Punti/ Penalità	
	A	B	C	D		
1					3	-0,5
2					2	-0,5
3					2	-0,5
4					3	-0,5
5						
6						
7						

Risposta alla domanda 8 (6 pt):

Risposta alla domanda 9 (7 pt):

Programmazione I

Prova scritta - 22 giugno 2018

Nome: _____ Cognome: _____

Matricola: _____

Indicare le risposte corrette apponendo una croce nella casella corrispondente. Per superare la prova bisogna aver raggiunto almeno 9 punti nelle domande a risposta singola/multipla, ed almeno 15 complessivamente. Usate questa copia per calcolare il voto da sole/soli durante la correzione.

	Risposte				Punti/ Penalità	
	A	B	C	D		
1					3	-0,5
2					2	-0,5
3					2	-0,5
4					3	-0,5
5						
6						
7						

Risposta alla domanda 8 (6 pt):

Risposta alla domanda 9 (7 pt):